

## Chapter 11

# Interfaces

Introduction .....	11-3
Naming Interfaces .....	11-4
Simple Interface Names .....	11-4
Fully Qualified Interface Names .....	11-5
Ethernet .....	11-6
Encapsulations .....	11-7
Synchronous Interfaces .....	11-9
Encapsulations .....	11-10
Modem Control Signals .....	11-10
Configuration .....	11-11
DS3 Interfaces .....	11-13
C-bit Parity Mode .....	11-14
Loopbacks .....	11-16
DS3 Module Specific Triggers .....	11-17
Interface Triggers .....	11-18
DS3 Configuration .....	11-19
Asynchronous Interfaces .....	11-21
Encapsulations .....	11-21
Configuration .....	11-22
Session Timeout .....	11-25
Connecting a Modem to the Asynchronous Port .....	11-25
MIB Counters .....	11-26
Autobauding .....	11-27
Making Asynchronous Ports Respond More Quickly .....	11-27
Testing Serial Data Circuits .....	11-28
Displaying Interfaces .....	11-28
Interface Link Traps .....	11-29
Managing Interfaces with SNMP .....	11-29
Command Reference .....	11-31
connect asyn .....	11-31
disable asyn .....	11-32
disable ds3 debug .....	11-32
disable ds3 test .....	11-33
disable ds3 trap .....	11-33
disable interface linktrap .....	11-34
disable syn .....	11-35
disable syn debug .....	11-35
enable asyn .....	11-36
enable ds3 debug .....	11-36
enable ds3 test .....	11-37
enable ds3 trap .....	11-39

enable interface linktrap .....	11-40
enable syn .....	11-41
enable syn debug .....	11-41
purge asyn .....	11-42
reset asyn .....	11-42
reset asyn counter .....	11-43
reset asyn history .....	11-44
reset ds3 .....	11-44
reset ds3 counters .....	11-45
reset interface counters .....	11-46
reset syn .....	11-46
reset syn counters .....	11-47
set asyn .....	11-48
set ds3 .....	11-53
set interface mtu .....	11-55
set interface traplimit .....	11-57
set syn .....	11-58
show asyn .....	11-61
show ds3 cmtce .....	11-69
show ds3 configuration .....	11-71
show ds3 counters .....	11-72
show ds3 debug .....	11-80
show ds3 state .....	11-81
show ds3 test .....	11-85
show ds3 trap .....	11-86
show ds3 utilisation .....	11-86
show interface .....	11-87
show syn .....	11-91
show syn counter .....	11-93

## 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.

Some interface and port types mentioned in this chapter may not be supported on your switch. The interface and port types that are available vary depending on your product model and whether an expansion unit is installed, such as a PIC, NSM, or expansion module. For more information, see the Hardware Reference for the switch.

The interfaces described are:

- DS3
- synchronous
- asynchronous

Other interfaces are described in:

- [Chapter 8, Switching](#) (Ethernet switch ports and VLANs).
- [Chapter 12, Integrated Services Digital Network \(ISDN\)](#) (Basic Rate ISDN (BRI) and Primary Rate ISDN (PRI) ports).

The term *interface* refers to one of the physical ports on the switch or on one of its expansion devices (such as PIC, NSM). The physical ports connect the switch to a network, and all data enters and leaves the switch via the interface.

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 62, Terminal Server](#) for information about using asynchronous ports for terminal serving. See [Chapter 63, Line Printer Daemon \(LPD\)](#) for information about using asynchronous ports for print serving using the Line Printer Daemon (LPD) protocol. See [Chapter 64, Stream Printing](#) 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.

Encapsulations supported for synchronous ports are:

- Frame Relay (see [Chapter 15, Frame Relay](#))
- PPP (see [Chapter 16, Point-to-Point Protocol \(PPP\)](#))
- X.25 (see [Chapter 14, X.25](#))

When one of these Layer 2 modules is *attached* to a synchronous interface, this creates a *logical interface*. The term *interface* also refers to these logical interfaces.

## 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.

Interfaces may be identified by their *simple name*, or for physical interfaces, by their *fully qualified name*.

### Simple Interface Names

Create simple interface names 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
<b>Logical</b>	
FR	Frame Relay interface
VLAN	Virtual LAN interface
LAPB	X.25 LAPB interface
PPP	Point-to-Point Protocol interface
X25C	X.25 DCE interface
X25T	X.25 DTE interface
<b>Physical</b>	
ASYN	Asynchronous interface
BRI	Basic Rate ISDN interface
DS3	Digital Signal 3 interface
PORT	Ethernet switch ports (including uplinks)
PRI	Primary Rate ISDN interface
SYN	Synchronous interface

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.

Permanent interfaces are numbered first, followed by removable interfaces (interfaces on PIC or NSM cards). For example, Rapier switches have a single permanent asynchronous console port named asyn0. If the AT-AR040 4-PIC NSM with the AT-AR025 ASYN4 PIC card is installed in an NSM bay, the

asynchronous interfaces on the PIC card would be asyn1, asyn2, asyn3, and asyn4.

The following table shows examples of valid names for simple interfaces.

Interface name	Description
fr2	Frame Relay instance 2
asyn4	Asynchronous port 4
ds30	DS3 port 0
port3	Switch port 3
ppp1	Point-to-Point Protocol instance 1
vlan1	Virtual LAN 1

## Fully Qualified Interface Names

Physical interfaces can be identified by their fully qualified interface name. A fully qualified interface name is one that includes the *path* to that interface. It is constructed by concatenating the interface name to the names of the bays and slots on the physical path from the base unit to the physical interface. Fields in the name are separated by dots. For simple interface names, the name consists of the physical interface type name ([“Simple Interface Names” on page 11-4](#)) followed by an instance number. The instance number uniquely identifies the interface in the NSM, PIC card or base unit.

If the interface is located directly on the base unit (not on a PIC or NSM card), then its fully qualified name is the same as its simple name—the interface type name and the interface instance number.

If the interface is located on a PIC or NSM card installed in the base unit, then the fully qualified name includes the bay where the PIC or NSM card is installed, the interface type name, and the interface instance number. For example, bay0.bri0 is the first Basic Rate ISDN port on the PIC card in PIC bay 0 on the base unit, and nsm0.bri2 is the second Basic Rate ISDN port on the NSM card in NSM bay 0 on the base unit.

If the interface is located on a PIC card installed in an AT-AR040 4-PIC NSM installed in the base unit, then the fully qualified name includes the bay where the NSM card is installed, the bay where the PIC card is installed, the interface type name, and the interface instance number. For example, nsm0.bay1.asyn2 is the third asynchronous port on the PIC card in PIC bay 1 of the NSM card in NSM bay 0 on the base unit.

The following table lists examples of valid fully qualified interface names and the equivalent simple names.

Interface Location	Interface Simple Name	Interface Fully Qualified Name
switch base	asyn0	asyn0
ASYN4 in NSM4PIC Bay 1	asyn1	nsm0.bay1.asyn0
	asyn2	nsm0.bay1.asyn1
	asyn3	nsm0.bay1.asyn2
	asyn4	nsm0.bay1.asyn3
SYNC1 in NSM4PIC Bay 2	syn0	nsm0.bay2.syn0
BRI1 in NSM4PIC Bay 3	bri0	nsm0.bay3.bri0

Any command that takes a physical interface as a parameter accepts either the simple name or the fully qualified name of the interface:

- Parameters that require a simple name created by concatenating an interface type and an instance number (e.g. OVER=syn0) also accept the fully qualified name (e.g. OVER=nsm0.bay0.syn0).
- Parameters that require a logical instance created by concatenating an interface type, an interface instance number and a logical instance number (e.g. INT=ppp1-1) also accept the fully qualified interface name (e.g. INT=nsm0.bay0.ppp0-1).
- Parameters that identify interfaces by an index number such as *ifIndex* (e.g. INT=1) also accept the fully qualified interface name (e.g. INT=nsm0.bay2.syn0).

For a summary of interfaces, including their fully qualified names, see the output of the [show interface command on page 11-87](#).

The [create config command on page 5-22 of Chapter 5, Managing Configuration Files and Software Versions](#) always generates configuration commands for physical interfaces using fully qualified names.

The interface table in the Enterprise MIB includes the read-only MIB object *arInterfaceFullName* that can be used to access the fully qualified names of all detected interfaces. The existing interface name entry is used to access the simple names of all interfaces.

## 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 8, 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.

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.

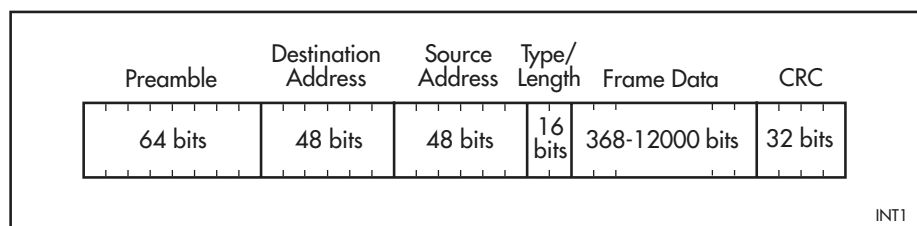
Ethernet was first defined in 1982. The original definition is generally referred to as Type 1 Ethernet and although it differs slightly from the modern standard, it is not very common today. Subsequent standards defined Type 2 Ethernet, which was largely ratified unchanged by the IEEE as IEEE 802.3.

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

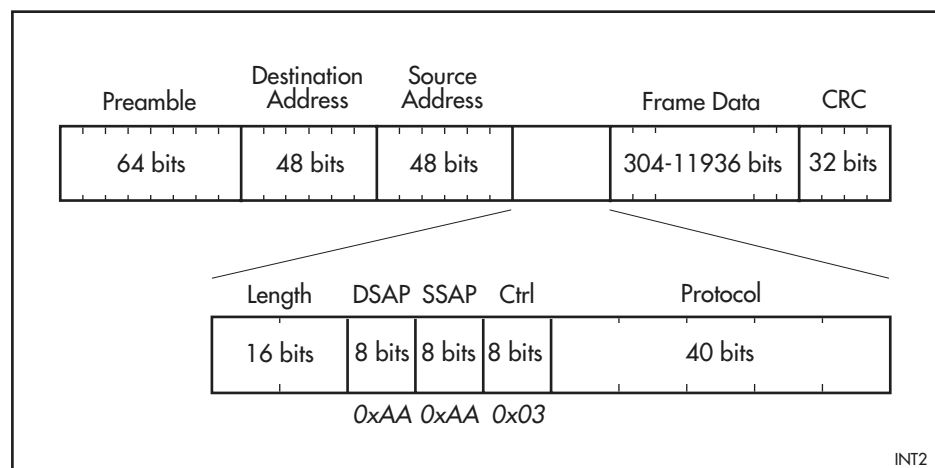
administered by Xerox Corporation and vendors of network equipment may apply to reserve a type field to define vendor-specific protocols.

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

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.

## Synchronous Interfaces

Synchronous or serial interfaces provide an alternative form of communication to Ethernet. Synchronous interfaces allow simultaneous communication in both the incoming and outgoing directions from the switch. Synchronous interfaces typically provide a direct connection between two network devices, although with the use of special procedures, a number of devices can be connected to a single *multidrop* line. The switch does not support multidrop lines.

Synchronous interfaces are normally used to provide wide area networking. In this situation, the synchronous interface on the network equipment, such as the switch, is connected to equipment belonging to the telecommunications provider. This equipment is typically a modem, or for newer installations, a Network Terminating Unit (NTU). The NTU provides access either directly to another site located some distance away, or into a network run by the network provider.

The switch supports synchronous interfaces with speeds of up to 2.048 Mbps, also known as E1. Different numbers of synchronous interfaces are provided on different models of the switch. See the Hardware Reference for details of the number, types and physical characteristics of synchronous interfaces on each model of the switch.

A feature of synchronous interfaces is that besides having circuits for the data being received and transmitted on the line, circuits also exist for clock signals. An item of network equipment must provide the clock on these circuits, and this is one of the main functions of the modem or NTU to which the synchronous interface is connected. However, in a small number of circumstances, it is useful to be able to generate the clock from within the switch. This is especially true for connections between the switch and host machines running X.25, where the switch and host might be in the same room and no modems or NTUs are provided.

The switch automatically generates a clock signal when a DCE transition cable is connected to a synchronous interface (see the Hardware Reference for details of how to construct a cable). The clock speed is set with the command:

```
set syn=n speed=speed
```

## Encapsulations

All encapsulations used by the switch on synchronous interfaces are themselves encapsulated using the HDLC (High-level Data Link Control) protocol. The HDLC protocol has the following features:

- Data comes in frames, delimited by special characters called flags.
- When a frame is not being sent, the sender transmits flags continually. This means that there is constant activity on any synchronous line that is running properly.
- The first bytes of data in a frame are interpreted as an address and a control field. The address may or may not have any special meaning, depending on the encapsulation, but the control field usually has some meaning.
- The last bytes of data in a frame are a CRC (Cyclic Redundancy Check) for detecting errors in the frame.

The data in the frame is interpreted by the device receiving the frame depending on the encapsulation type.

Different encapsulations use different methods to distinguish the different layer 3 protocol types. The switch currently supports three encapsulations for synchronous interfaces: Frame Relay, Point-to-Point (PPP) and Link Access Procedure for B Channels (LAPB).

## Modem Control Signals

Modem control signals can be configured so that output signals follow input signals or are permanently on or off. The modem control signals are grouped in the associated pairs RTS/CTS and DTR/DSR for RS-232/V.35 and C/I for X.21. The transition cable connected to the synchronous interface determines which pair or pairs of signals are being controlled. Changing from DTE mode to DCE mode by changing the transition cable alters which signal of the pair is the input and which is the output. It is possible to independently control the signals for each mode and to control CD in RS-232/V.35 DCE mode.

The actual modem control outputs that are present and which may be configured depends upon the transition cable used. The following table lists the modem control signals available for each transition cable type.

Transition Cable	Modem Control Outputs	Modem Control Inputs
RS-232 DTE	RTS, DTR	CTS, DSR
V.35 DTE	RTS, DTR	CTS, DSR
X.21 DTE	C	I
RS-232 DCE	CTS, DSR, CD	RTS, DTR
V.35 DCE	CTS, DSR, CD	RTS, DTR
X.21 DCE	I	C

For each transition cable type, one of the modem control inputs is used to determine the operational status of the interface. This status is the `ifOperStatus` object in the interface MIB for the interface and can be displayed using the **show interface** command. For the `ifOperStatus` of a SYN interface to be shown as “Up” the following conditions must apply:

- the SYN interface must be enabled
- a higher layer (e.g. PPP) must be attached to the SYN interface
- the relevant modem control input signal must be asserted.

The following table shows modem control input signals that controls the `ifOperStatus` for each transition cable type.

Cable type	Input signal
RS-232 DTE	CD
V.35 DTE	CD
X.21 DTE	I
RS-232 DCE	DTR
V.35 DCE	DTR
X.21 DCE	C

To avoid spurious `ifOperStatus` transitions due to “glitches” (changes of short duration) on the modem control input signal, the transitions are subjected to some hysteresis. The modem control signal must be asserted continuously for a period of at least one second before the `ifOperStatus` is changed to “Up”, and the modem control signal must be negated continuously for a period of at least two seconds before the `ifOperStatus` is changed to “Down”.

## Configuration

The encapsulation to be used on a synchronous interface is set by using the **create** command for the appropriate module, specifying the desired synchronous interface as the value of the **over** parameter. For example, to use the PPP encapsulation on synchronous interface 1 and to use the name PPP1, use the command:

```
create ppp=1 over=syn1
```

This creates the logical interface `ppp1` attached to synchronous interface 1. This command does not work when there is already an instance of the PPP module numbered 1. The layer 2 module instance number and the synchronous

interface number need not be the same, but it simplifies management when they are. For a complete description of attaching layer 2 module instances to +synchronous interfaces, see [Chapter 16, Point-to-Point Protocol \(PPP\)](#), [Chapter 15, Frame Relay](#) and [Chapter 14, X.25](#). See “Naming Interfaces” on [page 11-4](#) for an explanation of the convention used to name interfaces on the switch.

The configuration of a synchronous interface may be changed with the command:

```
set syn=n [c={on|off|i}] [cd={on|off|dtr}] [cts={on|off|rts}]
[dsr={on|off|dtr}] [dtr={on|off|dsr}] [I={on|off|c}]
[maxoqlenN=max-queue] [mintxint=min-interval] [rts={on|
off|cts}] [speed=speed]
```

The interface number must be specified, but each of the other parameters is optional. The **speed** parameter sets the clock rate of the interface if it is generating clocks and informs the switch of the clock rate if it is receiving clocks. Note that it is not strictly necessary to enter the interface clock speed if it is receiving clocks, but it is advisable as routing modules may need to know this to correctly set routing metrics for the interface. The **maxoqlen** parameter sets the maximum length of the output queue. It is useful to set a maximum output queue length to prevent a heavily loaded interface from causing buffer allocation problems. The **mintxint** parameter sets the minimum time interval between transmitted frames. Setting a minimum interval between transmitted frames may be required when the device at the other end of the synchronous link is unable to receive frames that are transmitted back-to-back. The default values of these parameters set a limit of 100 on the output queue and no minimum delay between transmitted frames. The **c**, **cd**, **cts**, **dsr**, **dtr**, **i** and **rts** parameters are used to set the modem control signals to **on**, to **off**, or to follow the corresponding input signals. The default is **on**.

The configuration of a synchronous interface may be displayed with the command:

```
show syn [=n]
```

where *n* is the number of the synchronous interface. The interface number is optional. If it is omitted the configuration of all synchronous interfaces is displayed.

The switch maintains a number of counters for each synchronous interface. The counters are objects in two standard MIBs and the switch's enterprise MIB. For more information about SNMP and MIBs, see [Chapter 55, Simple Network Management Protocol \(SNMP\)](#).

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

Category	Group	MIB table	RFC
INTERFACE	Interfaces	Interfaces	1213
SYN	Transmission	Synchronous port	1659

MIB counters for a synchronous interface may be displayed by using the command:

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

where *n* is the number of the synchronous interface and *category* is one of the two 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 syn** command.

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

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

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

Using the **reset syn counters** 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 syn counter** command on page 11-93.

Each synchronous interface may be enabled or disabled with the commands:

```
enable syn=n  
disable syn=n
```

where *n* is the number of the synchronous interface. When a synchronous interface is disabled, it does not transmit or receive data. The default state of an interface is enabled. To reset a synchronous interface, use the command:

```
reset syn=n
```

where *n* is the number of the synchronous interface. This is equivalent to an **disable syn** command followed by an **enable syn** command.

Data that is being received or transmitted is lost when the synchronous interface is disabled or reset.

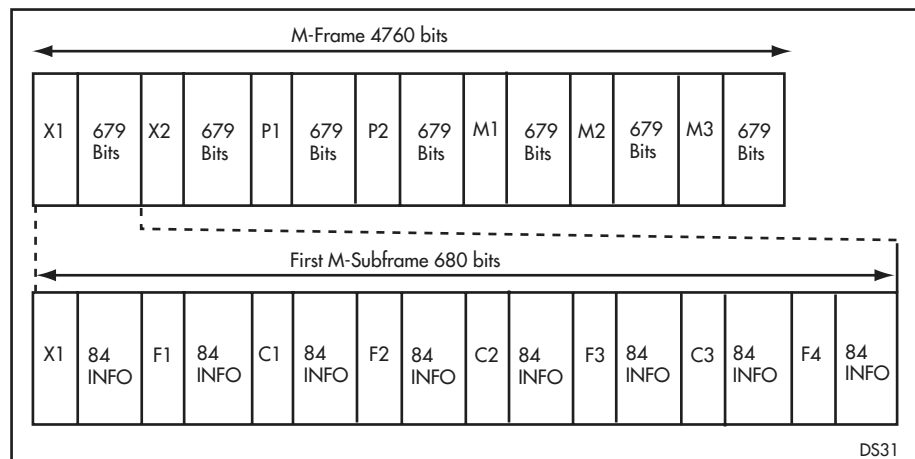
## DS3 Interfaces

---

*Digital Signal 3* (DS3) is a classification of digital signals, and sits at Layer 1 in the OSI model. Layer 1 provides a transmission link between two entities and monitors the quality of links. In DS3 this monitoring is achieved by adding overhead information alongside the data payload.

The DS3 interface rate is 44.736 Mbit/s with a payload rate of 44.210 Mbit/s. The signal is partitioned into *Multi-frames* (M-frames), and the M-frames are partitioned into seven M-subframes. Each M-subframe is further subdivided

into 8 blocks of 85 bits with 84 bits available for payload and one bit used for framing overhead. The following figure shows the framing structure.



The switch with the DS3 interface is called the *near end*. The entity the switch connects to is called the *far end*. X1 and X2 are set to 1 if the near end is receiving an *Alarm Indication Signal* (AIS), a *Loss Of Frame* (LOF), or a *Loss Of Signal* (LOS). This allows the near end to indicate to the far end that it is experiencing a problem and is known as *Far End Receive Failure* (FERF).

Bits P1 and P2 form the *P-bit channel*. They provide parity information for the preceding M-frame.

Bits M1, M2, and M3 form a *frame alignment channel* are used by the hardware to locate all seven M-subframes.

Bits F1, F2, F3, and F4 form an *M-subframe alignment channel* are used by the hardware to identify all frame overhead bit positions.

Bits C1, C2, and C3 form the *C-bit channel*.

## C-bit Parity Mode

The C-Bit maintenance path is used to:

- receive and report alarm status information from the remote end.
- transmit alarm status information to the remote end.
- transmit and receive a number of identification messages in the path overhead of the DS3 signal. These messages are in the format of text characters and lets users at either end of the DS3 path ensure that the correct DS3 signal has reached their equipment.

In C-bit parity mode the first C-bit in M-subframe 1 is set to 1 to identify the format as C-bit parity. If this is zero the format is assumed to be M23. The second C-bit in M-subframe 1 is designated Nr and is set to 1. The third C-bit in M-subframe 1 provides the *Far End Alarm and Control signal* (FEAC) that is used to send alarm or status information from the far end back to the near end, and to initiate DS3 loopbacks.

The three C-bits in M-subframe 3 are designated as CP-bits and are used to implement CP-bit parity. At the near end the CP-bits are set to the same value as the P-bits. The parity of the CP-bits of frame N are compared with the parity

of the CP-bits of frame N+1. A difference in parity between N and N+1 is deemed a CP-bit parity error.

The three C-bits in M-subframe 4 are designated as FEBE bits. The FEBE bits are returned to the far end to indicate the occurrence of a framing error or CP-bit parity error. If none occur, the FEBE bits are set to all ones. One or all of the FEBE bits are set to zero if a CP-bit parity error or an error in the F or M bits is found.

The three C-bits in M-subframe 5 are assigned as a 28.2kbits/s terminal-to-terminal path maintenance data link. This data link can be switched off at the command interface. If it is switched off, the C-bits in M-subframe 5 are set to all ones. If switched on, the maintenance channel can convey the following information:

■ **Path Identification Signal**

A set of ASCII text strings that can be used to uniquely identify this particular DS3 path. This can be useful if the DS3 signal is, at some point in its path, multiplexed into a higher order signal such as DS4 or OC-3. It is common for lower-order signals to be switched within a cross-connect. If this happens it is possible that the wrong DS3 signal is switched through to the switch. When this happens the overhead bits are all correct, so there is no indication that the wrong signal has been applied. The Path Identification Signal can be agreed by the two parties at either end of the network and tested to ensure that it is the right DS3 signal that has been received.

■ **Idle Signal**

A set of ASCII text strings that can be used to provide the location of the source of an idle signal.

■ **Test Signal**

A set of ASCII text strings that can be used to provide the location of the source of a test signal.

These signals are sent once every second.

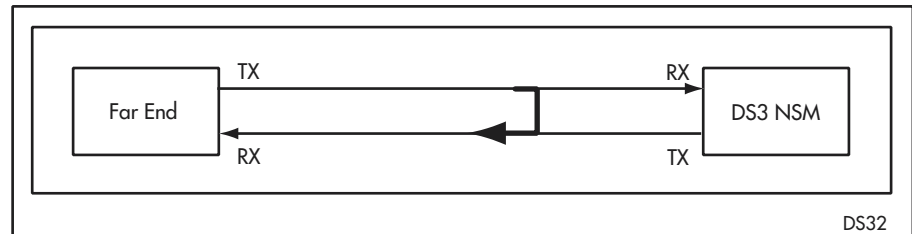
All other C-bits are reserved by the DS3 standards.

## Loopbacks

The DS3 interface provides four types of loopback; line, payload, diagnostic, and remote. These loopbacks are activated by the **enable ds3 test** command.

Line loopback is used to test the line and is shown in the following figure. In this loopback mode the receive signal is looped straight to the transmit signal.

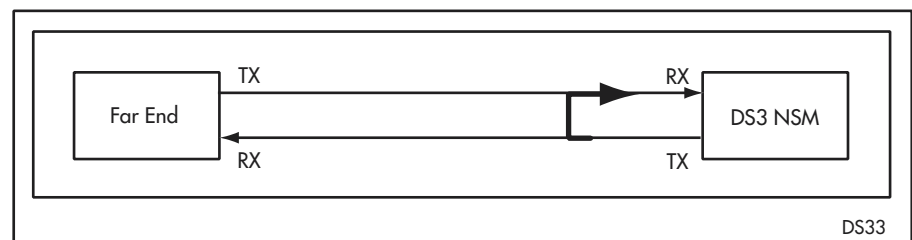
Figure 11-1: Line Loopback.



Payload loopback is similar to line loopback and is also used to test the line. The difference is that payload data is looped back from receive to transmit. The overhead is sourced from the DS3 NSM.

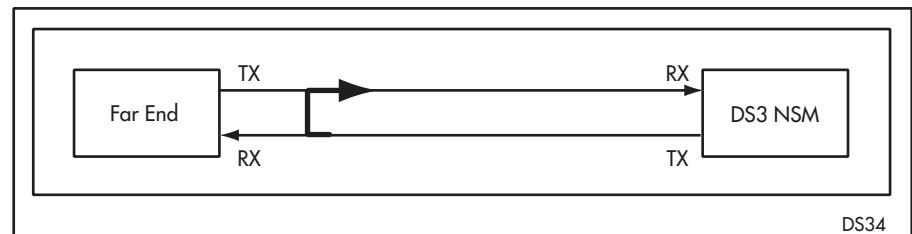
[Figure 11-2](#) shows how diagnostic loopback tests the internal DS3 interface. In this loopback mode the transmit signal is connected straight to the receive signal.

Figure 11-2: Diagnostic Loopback.



In C-bit parity mode it is also possible for the switch to request a loopback at the far end to check the line. This is shown in [Figure 11-3](#). This is achieved using the FEAC channel. The switch can also respond to remote loopback requests via the FEAC channel from the far end. This results in a near end line loopback shown in [Figure 11-2](#) above.

Figure 11-3: Remote Loopback.



Note that it is possible to configure near end local and remote loopbacks at the same time.



## DS3 Module Specific Triggers

The Trigger facility can be used to automatically run specified command scripts when particular events occur. When a trigger is activated by an event, parameters specific to the event are passed to the script that is run. For a description of triggers, see [Chapter 60, Trigger Facility](#) in the Software Reference. The DS3 module specific trigger describes thresholding and alerting for a DS3 interface. DS3 performance primitives and parameters are counted over 15-minute and 24-hour periods. If the threshold exceeds user-defined values set with the **set ds3** command the DS3 trigger is activated.

The following section lists: the events that may be specified for the DS3 module for the **event** parameter; the parameters that may be specified as *module-specific-parameters* for the DS3 module; and the arguments passed to the script activated by the trigger.

**Module** DS3 interface device drivers: MODULE={DS3|109}

**Event** TCA

**Description** This threshold crossing alert (TCA) trigger occurs when a performance monitoring parameter or primitive count exceeds a specified threshold.

**Parameters** The following command parameters can be specified in the **create** and **set trigger** commands.

Parameter	Description
Interface	The interface for which the trigger applies must be specified.

**Script arguments** The trigger passes arguments in the following table to the script.

Argument	Description
%1	Has a value between 1 and 10. <a href="#">Table 11-1</a> shows the meaning of the values passed in this argument.
%2	A value of 1 indicates that the trigger was caused by the counters exceeding a 15-minute threshold. A value of 2 indicates that the trigger was caused by the counters exceeding a 24-hour threshold.
%3	A value of 1 indicates that the trigger was caused by a near end counter exceeding a threshold. A value of 2 indicates that the trigger was caused by a far end trigger exceeding a threshold.

Table 11-1: Value definitions for Argument %1

Value	Meaning
1	PESs: The trigger was caused by the P-bit Errored Seconds parameter counter exceeding a specified threshold.
2	PSEs: The trigger was caused by the P-bit Severely Errored Seconds parameter counter exceeding a specified threshold.
3	SEFs: The trigger was caused by the Severely Errored Framing Seconds parameter counter exceeding a specified threshold.
4	UASs: The trigger was caused by the UnAvailable Seconds parameter counter exceeding a specified threshold.
5	LCVs: The trigger was caused by the Line Coding Violations primitive counter exceeding a specified threshold.

Table 11-1: Value definitions for Argument %1 (Continued)

Value	Meaning
6	PCVs: The trigger was caused by the P-bit Coding Violations primitive counter exceeding a specified threshold.
7	LESS: PSEs: The trigger was caused by the Line Error Seconds parameter and the PSEs parameter counters exceeding specified thresholds.
8	CCVs: The trigger was caused by the C-bit Coding Violations primitive counter exceeding a specified threshold.
9	CEs: The trigger was caused by the C-bit Errored Seconds parameter counter exceeding a specified threshold.
10	CSEs: The trigger was caused by the C-bit Severely Errored Seconds parameter counter exceeding a specified threshold.

The thresholds are set using the [set ds3 command on page 11-53](#). The following thresholds are available:

- 24-hour threshold for parameters
- 24-hour threshold for primitives
- 15-minute threshold for parameters
- 15-minute threshold for primitives

When a DS3 trigger is activated, it does not activate again within the same count period. For example, if a trigger is activated due to a 15-minute parameter count, no other 15-minute parameter counts activate a trigger until the 15-minute period has elapsed. This is to stop the trigger system from being overrun by many triggers in a short period of time. The log system catches all counts exceeding thresholds so the information is not lost.

## Interface Triggers

Interface triggers are generated when the interface state goes **up** or **down**.

### Event **up**

**Description** This trigger occurs when the DS3 interface state changes to **up** from any other state. This happens when a DS3 alarm failure is cleared. An alarm is cleared when it remains in a no-defect state for 10 seconds after a previous failure.

**Parameters** There are no command parameters for this event.

**Script arguments** There are no script arguments to pass to the script.

### Event **down**

**Description** This trigger occurs when the DS3 interface state changes from **up** to any other state. This happens when a DS3 alarm failure is set. An alarm is set if it remains in a defect state for 2.5 seconds.

**Parameters** There are no command parameters for this event.

**Script arguments** There are no script arguments to pass to the script.

## DS3 Configuration

The DS3 interface on the switch is automatically configured by the software modules when the switch sets up. Certain aspects of the DS3 signal can be altered to allow the switch to connect to another vendor's equipment via the DS3 interface.

To change the configuration of the DS3 interface, use the command:

```
set ds3=instance [clock={loop|internal}] [cmtce={on|off}]  
[direction={transmit|receive}] [eic=equipment-id]  
[facility=facility-id] [fdet={on|off}] [ric=frame-id]  
[fenno=generator-id] [lic=location-id] [param15=threshold]  
[param24=threshold] [port=port-id] [prim15=threshold]  
[prim24=threshold] [unit=unit-id] [type=[{pid|isid|tsid}]
```

The **clock** parameter specifies the clock source for the DS3 interface. The default is **loop** timing where the clock is derived from the received DS3 signal. If **internal** is selected the DS3 transmit signal is timed using an internal clock.

The **cmtce** parameter specifies whether the terminal-to-terminal path maintenance link is switched on. When **cmtce** is set to **off** the maintenance link is switched off and the C-bits in M-subframe 5 are all set to one. If **cmtce** is set to **on** the maintenance link is switched on. The path maintenance link allows a number of identification messages to be inserted in to the path overhead of the DS3 signal. These messages are in the form of text characters and allow the users at either end of the DS3 path to ensure that the correct DS3 signal has reached their equipment.

The **direction** parameter is used with the **eic**, **fac**, **fic**, **lic** or **unit** parameters to specify whether the text string is the text to transmit, or the text to expect in the received signal. If **transmit** is specified, the user is specifying a text to be transmitted out on the path. If **receive** is specified, the user is specifying the ASCII characters expected on the incoming path. If the **direction** parameter is present the **type** parameter must also be present.

The **eic** parameter specifies the Equipment Identification Code. This is a string 0 to 10 characters long and describes the equipment at the near end, for example, "**rapier sw**". This parameter is meaningful with the **cmtce** parameter set to **on**. With **cmtce** set to **off**, this parameter is ignored. The default is **ignore**.

The **fac** parameter specifies the facility identification code. This is a string 0 to 38 characters long and describes the DS3 path.

**Important** This data element is called FI in the ANSI specification. It is named **fac** in this document to avoid confusion with the **fic** parameter.

The **fac** parameter is valid if the **type** parameter is set to **pid**, and is meaningful with the **cmtce** parameter set to **on**. With **cmtce** set to **off** this parameter is ignored. The default is **ignore**.

The **fdet** parameter specifies whether fast detection of AIS is enabled. If **on** is specified, the AIS detection time is 2.23ms. If **off** is specified, the AIS detection time is 13.5ms. The default is **on**.

The **fic** parameter specifies the Frame Identification Code. This is a string 0 to 10 characters long and describes where the equipment is located within a building, for example, **frame 255**. This parameter is meaningful with the **cmtce** parameter set to **on**. With **cmtce** set to **off**, this parameter is ignored. The default is **ignore**.

The **genno** parameter specifies the test signal identification message. It is a string 0 to 38 characters long and describes the signal generator that initiates a test message. This parameter is valid if the **type** parameter is set to **ISIS**. This parameter is meaningful with the **cmtce** parameter set to **on**. With **cmtce** set to **off**, this parameter is ignored. The default is **ignore**.

The **lic** parameter specifies the Location Identifier Code. This is a string 0 to 11 characters long and describes the specific location of the equipment, for example, **building 1**. This parameter is meaningful with the **cmtce** parameter set to **on**. With **cmtce** set to **off** this parameter is ignored. The default is **ignore**.

The **param15** parameter specifies the 15-minute counter threshold for all performance monitoring parameters (**pes**, **pses**, **sef**, **uas**, **les**, **ces**, **cses**). If a trigger has been created, it asserts when the 24-hour count exceeds the value specified in **param15**. The value must be between 1 and 900. The default is **900**.

The **param24** parameter specifies the 24-hour counter threshold for all performance monitoring parameters (**pes**, **pses**, **sef**, **uas**, **les**, **ces**, **cses**). If a trigger has been created, it asserts when the 24-hour count exceeds the value specified in **param24**. The value must be between 1 and 65535 inclusive. The default is **65535**.

The **port** parameter specifies the port from which test signals are generated, and is a string 0 to 38 characters long. This parameter is valid if the **type** parameter is set to **tsid**. This parameter is meaningful with the **cmtce** parameter set to **on**. With **cmtce** set to **off** this parameter is ignored. The default is **ignore**.

The **prim15** parameter specifies the 15-minute counter threshold for all performance monitoring primitives (LCV, PCV, CCV). If a trigger has been created, it asserts when the 15-minute count exceeds the value specified in **prim15**. The value must be between 1 and 16383 inclusive. The default is **16383**.

The **prim24** parameter specifies the 24-hour counter threshold for all performance monitoring primitives (LCV, PCV, CCV). If a trigger has been created, it asserts when the 24-hour count exceeds the value specified in **prim24**. The value must be between 1 and 1048575 inclusive. The default is **1048575**.

The **type** parameter is used with the **eic**, **fac**, **fic**, **lic**, or **unit** parameters to specify whether the text string describes a path signal, an idle signal, or a test signal.

The **unit** parameter specifies where the equipment is located within a bay, for example, **shelf6**, and is 0 to 6 characters long. This parameter is meaningful with the **cmtce** parameter set to **on**. With **cmtce** set to **off** this parameter is ignored. The default is **ignore**.

To see the current configuration, use the command:

```
show ds3 configuration=n state
```

To display the DS3 counters, use the command:

```
show ds3 counters
```

To reset the counters, use the command:

```
reset ds3 counters
```

## 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 network printer port using the Line Printer Daemon (LPD) protocol. See [Chapter 63, Line Printer Daemon \(LPD\)](#) for more information. To use LPD, IP must be enabled and configured on all switches in the network providing access to the LPD print queues. See [Chapter 23, Internet Protocol \(IP\)](#) for more information.

An asynchronous port can be configured as a stream printer port using the stream printing service. See [Chapter 64, Stream Printing](#) 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 23, 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 asynchronous PPP encapsulation, SLIP (Serial Line Internet Protocol) encapsulation, and CSLIP (Compressed SLIP using Van Jacobson's header compression) encapsulation.

If an asynchronous port is assigned to an Asynchronous Call Control (ACC) call for dial-up connections, the call definition specifies the encapsulation to use or that the user is to be prompted to select an encapsulation during the

login process. See [Chapter 20, Asynchronous Call Control](#) for more information about defining calls and specifying encapsulations.

Asynchronous PPP encapsulation is used when the port is assigned to a PPP interface as a network interface with the command:

```
create ppp=ppp-interface over=acc-callname
```

where *callname* is the name of the asynchronous call. See [Chapter 16, Point-to-Point Protocol \(PPP\)](#) for more details about creating PPP interfaces.

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 23, 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 <b>cdcontrol</b> is set to <b>ignore</b> the switch ignores the state of the DCD input signal. If <b>cdcontrol</b> is set to <b>connect</b> the switch terminates existing connections when the DCD signal is deasserted (i.e. when a modem disconnects). If <b>cdcontrol</b> is set to <b>online</b> 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 <b>dtrcontrol</b> is set to <b>connect</b> the switch asserts DTR for the duration of the connection. If <b>dtrcontrol</b> is set to <b>on</b> or <b>off</b> , 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 <b>flow</b> is set to <b>none</b> the switch ignores all incoming flow control characters and lead transitions. If <b>flow</b> is set to <b>character</b> the switch uses <b>xon/xoff</b> flow control. If <b>flow</b> is set to <b>hardware</b> the switch uses the <b>rts/cts</b> lines for flow control.
Inflow	Sets the flow control mechanism used for the port in the receive direction. If <b>flow</b> is set to <b>none</b> , the switch ignores all incoming flow control characters and lead transitions. If <b>flow</b> is set to <b>character</b> , the switch uses <b>xon/xoff</b> flow control. If <b>flow</b> is set to <b>hardware</b> , the switch uses the <b>rts/cts</b> lines for flow control.

**Asynchronous  
port options**

Option	Description
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 <b>0</b> or <b>off</b> 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. See <a href="#">Chapter 20, Asynchronous Call Control</a> for more information.
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. See <a href="#">Chapter 20, Asynchronous Call Control</a> for more information.
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.
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. See <a href="#">Chapter 20, Asynchronous Call Control</a> for more information.
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 <b>flow</b> is set to <b>none</b> , the switch ignores all incoming flow control characters and lead transitions. If <b>flow</b> is set to <b>character</b> , the switch uses <b>XON/XOFF</b> flow control. If <b>flow</b> is set to <b>hardware</b> , 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 <a href="#">Chapter 43, User Authentication</a> 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 <b>vt100</b> or <b>dumb</b> . A <b>dumb</b> 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
Ipxnetwork	none
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 55, 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 11-31](#). 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, especially if the network configuration involves dynamic interfaces created by ISDN or ACC calls. 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 DS3, synchronous and asynchronous interfaces on the switch.

Some interface and port types mentioned in this chapter may not be supported on your switch. The interface and port types that are available vary depending on your product model and whether an expansion unit (PIC, NSM) is installed. For more information, see the Hardware Reference for the switch.

Some commands require IP and SNMP to be enabled and configured. See [Chapter 23, Internet Protocol \(IP\)](#) for a detailed description of the commands required to enable and configure IP. See [Chapter 55, 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 lxvi 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=asyn-number*

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

**Description**    This command creates a new terminal session that connects a Telnet session or the terminal (for a switch with more than one asynchronous port) 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=*asyn-number*

where *asyn-number* is the number of the port. Ports are numbered sequentially starting with 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 ds3 debug

---

**Syntax**    DISable DS3 [=*instance*] DEBbug [= {LINK | PACKet | ALL}]

where *instance* is the number of the DS3 interface

**Description**    This command disables debug options on the DS3 interface. If an interface is not specified, the debug options are disabled on all DS3 interfaces.

If **link** is specified, significant events in layer 1 of the DS3 interface are displayed.

If **packet** is specified, the contents of packets transmitted and received by the DS3 interface are displayed. If **all** is specified, all debugging options are disabled.

**Examples**    To disable all debugging options on all DS3 interfaces, use the command:

```
dis ds3 deb
```

**See Also**    [enable ds3 debug](#)  
[show ds3 debug](#)



## disable ds3 test

---

**Syntax** `DISable DS3=instance TEST[={test-number|ALL}]`

where:

- *instance* is the number of the DS3 interface.
- *test-number* is the number of the test to be enabled.

**Description** This command disables the specified test on the DS3 interface. If a test is not specified, all tests currently running on the interface are disabled. Only a single test can be disabled on each invocation or all tests disabled at once. Successive commands can be invoked to disable any combination of tests.

**Examples** To disable test 3 on DS3 interface 0, use the command:

```
dis ds3=0 test=3
```

To disable all tests on DS3 interface 0, use the command:

```
dis ds3=0 test
```

**Related Commands** [enable ds3 test](#)  
[show ds3 test](#)

## disable ds3 trap

---

**Syntax** `DISable DS3=instance TRap[={LINEstatustrap|TRap|ALL}]`

where *instance* is the number of the DS3 interface

**Description** This command disables the line status trap on the specified DS3 interface. If a trap is not specified, all traps are disabled.

If **linestatustrap** is specified, the line status trap is disabled. Note that this is the only supported trap.

If **tca** is specified, the threshold crossing alert trap is disabled.

If **all** is specified, all DS3 traps are disabled.

**Examples** To disable all traps on DS3 interface 0, use one of the following commands:

```
dis ds3=0 tr
```

```
dis ds3=0 tr=all
```

**Related Commands** [enable ds3 trap](#)  
[show ds3 trap](#)

## 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:

- FR (such as fr0)
- 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 23, Internet Protocol \(IP\)](#) for a detailed description of the commands required to enable and configure IP. See [Chapter 55, 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 ppp0, use the command:

```
dis int=ppp0 li
```

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

## disable syn

---

**Syntax** `DISable SYN=n`

where *n* is the number of the synchronous interface.

**Description** This command puts the synchronous interface into a state where it does not transmit or receive any frames. The interface number must be specified.

Data being received or transmitted when the synchronous interface is disabled or reset is lost.

**Examples** To disable synchronous interface 2, use the command:

```
dis syn=2
```

**Related Commands** [enable syn](#)  
[reset syn](#)  
[show syn](#)

## disable syn debug

---

**Syntax** `DISable SYN=n DEBug`

where *n* is the number of the synchronous interface.

**Description** This command disables the output of debug messages for a synchronous interface. The output of debug messages is disabled by default.

**Examples** To disable debugging messages on synchronous interface 1, use the command:

```
dis syn=1 debug
```

**Related Commands** [enable syn debug](#)  
[show syn](#)

## enable asyn

---

**Syntax**    `ENABle ASYn=asyn-number`

where *asyn-number* is the number of the port. Ports are numbered sequentially starting with 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 ds3 debug

---

**Syntax**    `ENABle DS3 [=instance] DEBug [{LINK|PACKet|ALL}]`  
              `[NUMPkts={CONTinuous|1..4000000000}]`

where *instance* is the number of the DS3 interface

**Description**    This command enables debug options on the DS3 interface. If an interface is not specified, the debug options are enabled on all DS3 interfaces.

If **link** is specified, significant events in layer 1 of the DS3 interface are displayed.

If **packet** is specified, the contents of packets transmitted and received by the DS3 interface are displayed. If **all** is specified, all debugging options enabled.

The **numpkts** parameter specifies, for packet debugging, the number of packets to be displayed before packet debugging ceases. This is useful when attempting to debug a busy link, since the amount of output generated by packet debugging can easily cause the switch to lock up the device to which the debugging output is being set. The default is **contiguous**.

**Example**    To enable debugging on DS3 interface 0, use the commands:

```
ena ds3=0 deb
```

**Related Commands**    [disable ds3 debug](#)  
[show ds3 debug](#)

## enable ds3 test

**Syntax** `ENABle DS3=instance TEST=test-number { [BERCount=bercount] | [Rate=ber-rate] }`

where:

- *instance* is the number of the DS3 interface.
- *test-number* is the number of the test to be enabled.
- *bercount* is the number of bits errors from 1 to 100 to be injected.
- *ber-rate* is the bit error rate from 0 to 7 to be injected.
  - 0 = No bit errors - switch off Bit error injection.
  - 1=10<sup>1</sup> errors i.e. 1 in 10 bits in error.
  - 1=10<sup>2</sup> errors i.e. 1 in 100 bits in error.
  - 1=10<sup>3</sup> errors i.e. 1 in 1000 bits in error.
  - 1=10<sup>4</sup> errors i.e. 1 in 10000 bits in error
  - 1=10<sup>5</sup> errors i.e. 1 in 100000 bits in error.
  - 1=10<sup>6</sup> errors i.e. 1 in 1000,000 bits in error.
  - 1=10<sup>7</sup> errors i.e. 1 in 10,000,000 bits in error

**Description** This command enables specific tests on the DS3 interface. Only a single test can be enabled on each invocation. Successive commands can be used to enable any combination of tests.

Bit error testing (test 11) can be enabled only if the PRGD test (test 5) is already enabled. This command is required for testing only, and should not be used for normal operation of the DS3 interface.

The following table lists the DS3 interface test modes for an interface with TECT3 framer hardware.

Test	Function	Description
1	Line Loopback	A loopback of the received 44.736 Mbit/s signal back out to the transmit interface.
2	Payload Loopback	A loopback of the received payload back to the transmit interface. The overhead is regenerated.
3	Diagnostic Loopback	A loopback of the transmitted 44.736 Mbit/s signal back to the receive interface.
4	Remote Loopback	A request to activate a loopback at the Far End - Supported in C-bit parity mode.
5	PRGD	A pseudo-random generated pattern, 220-1 as per ITU 0.151, is inserted in the DS3 payload.
6	TSIG	A 100 bit pattern is sent on the transmit interface.
7	FERF	The Far End Receive Failure maintenance signal is sent on the transmit signal (X1 and X2 bits are set to 0).
8	IDL	Enables transmission of the DS3 Idle signal.
9	AIS	The AIS signal is sent on the transmit interface.
10	LOS	The LOS signal is sent on the transmit interface (i.e. the signal is forced to continuous zeros).

Test	Function	Description
11	BER	Injects a specified Number of bit errors or a specified error rate. The number of bit errors is specified by the parameter BERCOUNT=x. Value x must be less than 101. If x is not specified then a single bit error is injected. The rate is specified by the parameter RATE=y. Value y must be between 0 and 7 with 0 switching off the BER rate test. BER injection is valid only when the PRGD test (test 5) is already enabled.
12	Inject LCV	This test ends automatically if the BERCOUNT parameter is specified. This test must be disabled manually if the RATE parameter is specified. The BERCOUNT and RATE parameters can not be specified at the same time.
13	LCV	Inserts a single Line Code Violation into the transmit signal. This test is never disabled as it ends automatically after the LCV has been injected.
14	DFERR	Injects F-bit Framing Errors. This test is never disabled as it ends automatically after the DFERRs have been injected.
15	DMERR	Injects M-bit Framing Errors. This test is never disabled as it ends automatically after the DMERRs have been injected.
16	DCPERR	Injects CP-bit Parity Errors. This test is never disabled as it ends automatically after the DCPERRs have been injected.
17	DPERR	Inject P-bit Parity Errors. This test is never disabled as it ends automatically after the DPERRs have been injected.
18	DFEBE	Injects FEBE Errors. This test is never disabled as it ends automatically after the DFEBEs have been injected.

**Examples** To set an internal loopback and inject a Line Code Violation on DS3 interface 0, use the commands:

```
ena ds3=0 test
ena ds3=0 test=3
ena ds3=0 test=9
```

**Related Commands** [disable ds3 test](#)  
[show ds3 test](#)

---

## enable ds3 trap

---

**Syntax** `ENable DS3=instance TRap [= (LINEstatustrap | TCa | ALL)]`

where *instance* is the number of the DS3 interface

**Description** This command enables SNMP traps on the specified DS3 interface. If a trap is not specified, all traps are disabled.

If **linestatustrap** is specified, the line status trap is enabled. Note that this is the only supported trap.

If **tca** is specified, the threshold crossing alert trap is disabled.

If **all** is specified, all DS3 traps are enabled.

**Examples** To enable all traps on DS3 interface 0, use one of the following commands:

```
ena DS3=0 tr
ena DS3=0 tr=all
```

**Related Commands** [enable ds3 trap](#)  
[show ds3 trap](#)

## 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:

- FR (such as fr0)
- 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 23, Internet Protocol \(IP\)](#), and [Chapter 55, 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](#)



## enable syn

---

**Syntax**    `ENABle SYN=n`

where *n* is the number of the synchronous interface

**Description**    This command is used to reinitialise and enable a synchronous interface that has been disabled. The default state of a synchronous interface is enabled. The interface number must be specified.

**Examples**    To enable synchronous interface 2, use the command:

```
ena syn=2
```

**Related Commands**    [disable syn](#)  
[reset syn](#)  
[show syn](#)

## enable syn debug

---

**Syntax**    `ENABle SYN=n DEBbug`

where *n* is the number of the synchronous interface

**Description**    This command enables output of debug messages for a synchronous interface. The output of debug messages is disabled by default.

**Examples**    To enable debugging messages on synchronous interface 1, use the command:

```
ena syn=1 deb
```

**Related Commands**    [disable syn debug](#)  
[show syn](#)

## 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 ds3

---

**Syntax** RESET DS3=*instance*

where *instance* is the number of the DS3 interface

**Description** This command resets the DS3 interface. The interface number must be specified.

**Examples** To reset DS3 interface 0, use the command:

```
reset ds3=0
```

**Related Commands** [reset ds3 counters](#)  
[set ds3](#)  
[show ds3 counters](#)

---

## reset ds3 counters

---

**Syntax** RESET DS3 [= *instance*] COunters [{HDLc | INTerface | LINK |  
DIAGnostic | STAtE | ALL}]

where *instance* is the number of the DS3 interface

**Description** This command resets the CLI counters for a DS3 interface. The interface number is optional. If the interface number is not specified, then the counters for all DS3 interfaces are reset. If a category is specified, the counters in that category are reset. If the category is not specified, the counters for all categories are reset.

If **hdlc** is specified, the counters stored in the enterprise MIB relevant to a DS3 interface are reset to zero.

If **interface** is specified, the counters from the interfaces table of the interfaces MIB relating to the DS3 are reset to zero.

If **link** is specified, the counters stored in the enterprise MIBs related to the performance of the link during the current 15-minute interval are reset to zero.

If **diagnostic** is specified, the diagnostic counters for the DS3 interface hardware and software drivers are reset to zero.

If **state** is specified, the counters associated with failures and defects are reset to zero.

If **all** is specified, all DS3 counters are reset to zero except the performance monitoring history counters.

Note that where there is an equivalent SNMP GET function, the switch will supply the counters' original values, which may be different from those shown for the CLI counters.

**Examples** To reset the link counters for DS3 interface 0, use the command:

```
reset ds3=0 cou=lin
```

To reset all counters in all DS3 interfaces, use the command:

```
reset ds3 cou
```

**Related Commands** [reset ds3](#)  
[show ds3 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.

- FR (such as fr0)
- 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 ppp0 interface MIB counters, use either of the following commands:

```
reset int=ppp0 cou
reset int=1 cou
```

**Related Commands** [show interface](#)

## reset syn

---

**Syntax** RESET SYN=*n*

where *n* is the number of the synchronous interface.

**Description** This command is equivalent to the command sequence:

```
disable syn=n
enable syn=n
```

Ideally, this command should never be required but some circumstances may require it. The interface number must be specified.

Data being received or transmitted when the synchronous interface is disabled or reset is lost.

**Examples** To reset synchronous interface 2, use the command:

```
reset syn=2
```

**Related Commands** [disable syn](#)  
[enable syn](#)  
[reset syn](#)  
[show syn](#)

---

## reset syn counters

---

**Syntax** `RESET SYN[=n] COUnters[={INTERface|SYN}]`

where *n* is the number of the synchronous interface

**Description** This command resets the MIB counters for a synchronous interface. The interface number is optional. If the interface is not specified, then the counters for all interfaces are reset. If a category is specified, the counters in that category are cleared. If the category is not entered, then the counters for all categories are reset. For a description of the categories, see the **show syn counter** command.

Note that this command does not reset the counters themselves. It first initiates a snapshot of the current counter values, then each time the relevant *Show* command is run each of the snapshot values are subtracted from each of these from the new counter values. This gives the same set of values from the various show commands as would display had the counter been reset.

Where there is an equivalent SNMP GET function, the switch supplies the counters' original values, and not the ones displayed by the **Show** commands.

The control signal transition counters displayed by the **show syn counter** command are reset along with the other counters in the SYN category.

**Examples** To reset the **syn** counter for synchronous interface 2, use the command:

```
reset syn=2 cou=syn
```

**Related Commands** [show syn counter](#)

## set asyn

**Syntax** SET ASYn[=*asyn-number*] [Attention={Break|*alphabetical control char*|^|[None]}] [CDcontrol={Connect|Ignore|Online}] [DATAbits={5|6|7|8}] [DEFAultservice={ON|OFF|YES|NO|True|False}] [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}] [SERvice={*service-name*|None}] [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.
- *service-name* is the name of a service 1 to 15 characters long, with no embedded spaces. The first character must be alphabetic (A–Z). The name 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**, **ipxnetwork**, **mtu**, **service**, 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 **defaultservice** parameter is used to configure a port to automatically connect to a service whenever a user types anything at a terminal connected to the port, or (in the case of a modem attached to the port) when the modem asserts DCD. This parameter is valid if a service is associated with the port, using the **service** parameter. A port configured for **defaultservice** can not be used to enter commands to the switch because the port connects to the default service whenever anything is typed at the switch prompt. The **defaultservice** parameter changes the meaning of the **service** parameter and the way the port operates. If **defaultservice** is set to **off** (the default) the port acts as an interactive service port for the service specified by the **service** parameter, and is used to connect the switch to the RS-232 port of a host. If **defaultservice** is set to **on**, the **service** parameter specifies the name of the service (interactive or Telnet) to which an automatic connection is to be made. In outputs of the [show asyn command on page 11-61](#) for a **defaultservice** port, the service name is either prefixed by an asterisk or followed by the string “(default)”.

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 11-32](#). 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 11-25](#) 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 2-16 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 11-44](#). 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. See [Chapter 20, Asynchronous Call Control](#) for more information. The IP address may be cleared by setting **ipaddress** to **none**. The default is **none**.

The **ipxnetwork** parameter specifies the Novell network number assigned to a user accessing a Novell internetwork via the asynchronous port. See [Chapter 20, Asynchronous Call Control](#) for more information. 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. See [Chapter 20, Asynchronous Call Control](#) for more information. 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 11-61](#).

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 43, 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 **service** parameter allocates an asynchronous port to be a host port for the named service. This port must be an unallocated terminal port. The service must already have been defined with the [set service command on page 62-21 of Chapter 62, Terminal Server](#) and be of type **interactive**. If **service** is set to **none** the port is deallocated from the service.

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 62-14 of Chapter 62, 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 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 11-27](#)). 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 11-32](#). To re-enable the port, send it a break signal. See [“Session Timeout” on page 11-25](#) 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 service](#)
- [show tty](#)

## set ds3

---

**Syntax** SET DS3=*instance* [CLOCK={LOOP|INTERNAL}] [CMTCE={ON|OFF}]  
 [DIRECTION={TRANSMIT|RECEIVE}] [EIC=*equipment-id*]  
 [FACILITY=*facility-id*] [FDET={ON|OFF}] [FIC=*frame-id*]  
 [GENNO=*generator-id*] [LIC=*location-id*]  
 [PARAM15=*threshold*] [PARAM24=*threshold*] [PORT=*port-id*]  
 [PRIM15=*threshold*] [PRIM24=*threshold*] [UNIT=*unit-id*]  
 [TYPE={ [PID|ISID|TSID] }

where:

- *instance* is the number of the DS3 interface. The first instance is 0.
- *equipment-id* is a character string 0 to 10 characters long. Valid characters are any ASCII printable character. If the string contains spaces, it must be in double quotes.
- *facility-id* is a character string 0 to 38 characters long. Valid characters are any ASCII printable character. If the string contains spaces, it must be in double quotes.
- *frame-id* is a character string 0 to 10 characters long. Valid characters are any ASCII printable character. If the string contains spaces, it must be in double quotes.
- *generator-id* is a character string 0 to 38 characters long. Valid characters are any ASCII printable character. If the string contains spaces, it must be in double quotes.
- *location-id* is a character string 0 to 11 characters long. Valid characters are any ASCII printable character. If the string contains spaces, it must be in double quotes.
- *threshold* is the counter threshold for all performance monitoring parameters.
- *port-id* is a character string 0 to 38 characters long. Valid characters are any ASCII printable character. If the string contains spaces, it must be in double quotes.
- *unit-id* is a character string 0 to 6 characters long. Valid characters are any ASCII printable character. If the string contains spaces, it must be in double quotes.

**Description** This command sets the values of the user configurable DS3 operational parameters.

The **clock** parameter specifies the clock source for the DS3 interface. The default is **loop** timing where the clock is derived from the received DS3 signal. If **internal** is selected the DS3 transmit signal is timed using an internal clock.

The **cmtce** parameter specifies whether the terminal-to-terminal path maintenance link is switched on. When **cmtce** is set to **off** the maintenance link is switched off and the C-bits in M-subframe 5 are all set to one. If **cmtce** is set to **on** the maintenance link is switched on. The path maintenance link allows a number of identification messages to be inserted in to the path overhead of the DS3 signal. These messages are in the form of text characters and allow the users at either end of the DS3 path to ensure that the correct DS3 signal has reached their equipment.

The **direction** parameter is used with the **eic**, **fac**, **fic**, **lic** or **unit** parameters to specify whether the text string is the text to transmit, or the text to expect in the received signal. If **transmit** is specified, the user is specifying a text to be transmitted out on the path. If **receive** is specified, the user is specifying the ASCII characters expected on the incoming path. If the **direction** parameter is present the **type** parameter must also be present.

The **eic** parameter specifies the Equipment Identification Code. This is a string 0 to 10 characters long and describes the equipment at the near end, for example, "**rapier sw**". This parameter is meaningful with the **cmtce** parameter set to **on**. With **cmtce** set to **off**, this parameter is ignored. The default is **ignore**.

The **fac** parameter specifies the facility identification code. This is a string 0 to 38 characters long and describes the DS3 path. This data element is called FI in the ANSI specification. It is named FAC in this document to avoid confusion with the FIC parameter.

The **fac** parameter is valid if the **type** parameter is set to **pid** and is meaningful with the **cmtce** parameter set to **on**. With **cmtce** set to **off**, this parameter is ignored. The default is **ignore**.

The **fdet** parameter specifies whether fast detection of AIS is enabled. If **on** is specified, the AIS detection time is 2.23ms. If **off** is specified, the AIS detection time is 13.5ms. The default is **on**.

The **fic** parameter specifies the Frame Identification Code. This is a string 0 to 10 characters long and describes where the equipment is located within a building, for example, **frame 255**. This parameter is meaningful with the **cmtce** parameter set to **on**. With **cmtce** set to **off**, this parameter is ignored. The default is **ignore**.

The **genno** parameter specifies the test signal identification message. It is a string 0 to 38 characters long and describes the signal generator that initiates a test message. This parameter is valid if the **type** parameter is set to **isis**. This parameter is meaningful with the **cmtce** parameter set to **on**. With **cmtce** set to **off** this parameter is ignored. The default is **ignore**.

The **lic** parameter specifies the Location Identifier Code. This is a string 0 to 11 characters long and describes the specific location of the equipment, for example, **building 1**. This parameter is meaningful with the **cmtce** parameter set to **on**. With **cmtce** set to **off**, this parameter is ignored. The default is **ignore**.

The **param15** parameter specifies the 15-minute counter threshold for all performance monitoring parameters (**pes**, **pses**, **sef**, **uas**, **les**, **ces**, **cses**). If a trigger has been created, it asserts when the 24-hour count exceeds the value specified in **param15**. The value must be between 1 and 900. The default is **900**.

The **param24** parameter specifies the 24-hour counter threshold for all performance monitoring parameters (**pes**, **pses**, **sef**, **uas**, **les**, **ces**, **cses**). If a trigger has been created, it asserts when the 24-hour count exceeds the value specified in **param24**. The value must be between 1 and 65535 inclusive. The default is **65535**.

The **port** parameter specifies the port from which test signals are generated, and is a string 0 to 38 characters long. This parameter is valid if the **type** parameter is set to **tsid**. This parameter is meaningful with the **cmtce** parameter set to **on**. With **cmtce** set to **off** this parameter is ignored. The default is **ignore**.

The **prim15** parameter specifies the 15-minute counter threshold for all performance monitoring primitives (LCV, PCV, CCV). If a trigger has been created, it asserts when the 15-minute count exceeds the value specified in **prim15**. The value must be between 1 and 16383 inclusive. The default is **16383**.

The **prim24** parameter specifies the 24-hour counter threshold for all performance monitoring primitives (LCV, PCV, CCV). If a trigger has been created, it asserts when the 24-hour count exceeds the value specified in **prim24**. The value must be between 1 and 1048575 inclusive. The default is **1048575**.

The **type** parameter is used with the **eic**, **fac**, **fic**, **lic**, or **unit** parameters to specify whether the text string is used to describe a path signal, an idle signal, or a test signal.

The **unit** parameter specifies where the equipment is located within a bay, for example, **shelf6**, and is 0 to 6 characters long. This parameter is meaningful with the **cmtce** parameter set to **on**. With **cmtce** set to **off**, this parameter is ignored. The default is **ignore**.

**Examples** To put the interface in to internal clock mode:

```
set ds3 cl=int
```

**Related Commands** [disable ds3 test](#)  
[enable ds3 test](#)  
[show ds3 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:

- FR (such as fr0)
- PPP (such as ppp0, ppp1-1)
- VLAN (such as vlan1)

To see a list of current interfaces, use the [show interface command on page 11-87](#).

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
FR	256	1600	1600
PPP (not over Eth)	256	1500	1500
PPP over Eth	256	1492 <sup>a</sup>	1492
ETH	256	1500	1500

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

If the interface given by the user is a Frame Relay interface, all logical interfaces in the interface are affected. If a specific interface is given, only that logical interface is affected.

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

```
set int=ppp3 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:

- FR (such as fr0)
- 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 23, Internet Protocol \(IP\)](#) for a detailed description of the commands required to enable and configure IP. See [Chapter 55, 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 ppp2 to 40, use the command:

```
set int=ppp2 tr=40
```

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

## set syn

**Syntax** SET SYN=*n* [C={ON|OFF|I}] [CD={ON|OFF|DTR}] [CTS={ON|OFF|RTS}] [DATAsense={NORMAl|INVerted}] [DSR={ON|OFF|DTR}] [DTR={ON|OFF|DSR}] [I={ON|OFF|C}] [MAXoqlen=*max-queue*] [MINTxint=*min-interval*] [RTS={ON|OFF|CTS}] [SPeed=*speed*] [TXClock={FALling|RISing}]

where:

- *n* is the number of the synchronous interface.
- *speed* is the baud rate for the interface.
- *max-queue* is the maximum permitted output queue length for the interface.
- *min-interval* is the minimum number of microseconds between transmitted frames.

**Description** This command sets the configuration parameters for a synchronous interface. The interface number must be specified.

The **c** parameter specifies the mode of operation for the X.21 modem control signal output called "C". This parameter is valid when an X.21 DTE transition cable is connected to the synchronous interface, and is ignored if any other transition cable is used. When **c** is **on**, the output is always on when the interface is active. When **c** is **off**, the output is always off when the interface is active. When **c** is **i**, then when the interface is active, **c** is on when the modem control input signal **i** is on and **c** is off when **i** is off. An interface is active when there is a layer 2 module attached and the interface is enabled. If the layer 2 module takes control of this output, then the setting of this parameter is ignored. The default value for **c** is **on**.

The **cd** parameter specifies the mode of operation of the RS-232/V.35 modem control signal output called "CD". This parameter is valid when an RS-232 DCE or V.35 DCE transition cable is connected to the synchronous interface, and is ignored if any other transition cable is used. When **cd** is **on**, the output is always on when the interface is active. When **cd** is **off**, the output is always off when the interface is active. When **cd** is **dtr**, then when the interface is active, **cd** is on when the modem control input signal **dtr** is on, and off when **dtr** is off. An interface is active when there is a layer 2 module attached and the interface is enabled. If the layer 2 module takes control of this output, then the setting of this parameter is ignored. The default is **on**.

The **cts** parameter specifies the mode of operation for the RS-232/V.35 modem control signal output called "CTS". This parameter is valid when an RS-232 DCE or V.35 DCE transition cable is connected to the synchronous interface, and is ignored if any other transition cable is used. When **cts** is **on**, the output is always on when the interface is active. When **cts** is **off**, the output is always off when the interface is active. When **cts** is **rts**, then when the interface is active, **cts** is on when the modem control input signal RTS is on, and off when RTS is off. An interface is active when there is a layer 2 module attached and the interface is enabled. If the layer 2 module takes control of this output, then the setting of this parameter is ignored. The default is **on**.

The **datasense** parameter specifies the sense of the transmitted data on the synchronous interface. The default is **normal**. The **inverted** option may be used when the synchronous link is a T1 link and the switch is responsible for limiting the number of consecutive zeroes transmitted, for example when raw

AMI encoding rather than B8ZS is used. Both ends of the T1 link must be set to **inverted** as the receiver also expects to receive inverted data. Note that 68302 and 68562 hardware types do not support data inversion. The hardware type can be read from the output of the [show syn command on page 11-91](#).

The **dsr** parameter specifies the mode of operation for the RS-232/V.35 modem control signal output called “DSR”. This parameter is valid when an RS-232 DCE or V.35 DCE transition cable is connected to the synchronous interface, and is ignored if any other transition cable is used. When **dsr** is **on**, the output is always on when the interface is active. When **dsr** is **off**, the output is always off when the interface is active. When **dsr** is **dtr**, then when the interface is active, DSR is on when the modem control input signal DTR is on, and off when DTR is off. An interface is active when there is a layer 2 module attached and the interface is enabled. When the layer 2 module takes control of this output, then the setting of this parameter is ignored. The default is **on**.

The **dtr** parameter specifies the mode of operation for the RS-232/V.35 modem control signal output called “DTR”. This parameter is valid when an RS-232 DTE or V.35 DTE transition cable is connected to the synchronous interface, and is ignored if any other transition cable is used. When **dtr** is **on**, the output is always on when the interface is active. When **dtr** is **off**, the output is always off when the interface is active. When **dtr** is **dsr**, then when the interface is active, DTR is on when the modem control input signal DSR is on, and off when DSR is off. An interface is active when there is a layer 2 module attached and the interface is enabled. When the layer 2 module takes control of this output, then the setting of this parameter is ignored. The default is **on**.

The **i** parameter specifies the mode of operation for the X.21 modem control signal output called “I”. This parameter is valid when an X.21 DCE transition cable is connected to the synchronous interface, and is ignored if any other transition cable is used. When **i** is **on**, the output is always on when the interface is active. When **i** is **off**, the output is always off when the interface is active. When **i** is **c**, then when the interface is active, I is on when the modem control input signal C is on, and off when C is off. An interface is active when there is a layer 2 module attached and the interface is enabled. When the layer 2 module takes control of this output, then the setting of this parameter is ignored. The default is **on**.

The value of **maxoqlen** is the maximum number of frames that may be queued for transmission over the synchronous interface at any one time. Each frame passed to the SYN module by the layer 2 module is inserted in the output queue according to its priority and the queue length incremented. When the output queue length is then greater than the **maxoqlen** value, then the oldest, lowest priority frame is discarded. The **maxoqlen** parameter is optional. The default is **100**. A value of 0 means there is no limit on the length of the output queue. We recommend setting **maxoqlen** to at least 5. The most common layer 2 protocols (e.g. PPP and Frame Relay) maintain their own packet transmission queues. A small number of packets are queued by the synchronous interface to ensure maximum possible bandwidth utilisation. If the synchronous interface maximum queue length is set too low, packets may be discarded by the synchronous interface to enforce the maximum queue length. This may disrupt correct operation of the layer 2 protocol and any Quality of Service mechanisms that have been configured. The synchronous interface queue is priority aware, so reducing the maximum length will not provide a latency improvement.

The value of **mintxint** sets the minimum delay (in microseconds) between frames transmitted over an interface. This is useful where the switch is communicating with a device that is unable to accept back-to-back frames. The

maximum permitted value is 1000000 (1 second). The **mintxint** parameter is optional. The default value is zero, which means no delay is inserted between transmitted frames.

The **rts** parameter specifies the mode of operation for the RS-232/V.35 modem control signal output called "RTS". This parameter is valid when an RS-232 DTE or V.35 DTE transition cable is connected to the synchronous interface, and is ignored if any other transition cable is used. When **rts** is **on**, the output is always on when the interface is active. When **rts** is **off**, the output is always off when the interface is active. When **rts** is **cts**, then when the interface is active, RTS is on when the modem control input signal CTS is on, and off when CTS is off. An interface is active when there is a layer 2 module attached and the interface is enabled. When the layer 2 module takes control of this output, then the setting of this parameter is ignored. The default is **on**.

The **speed** parameter specifies the baud rate for the interface. The baud rate should be specified, even if the interface is being clocked by an external modem. This enters the speed into the interface MIB so that any module that requires this information (e.g. a routing module) can use it (for example, to calculate metrics for the interface). The baud rate may be set to any value between 0 and 10000000. If the interface is configured to generate clocks then the maximum permitted speed is 64000 for 68302-based synchronous interfaces and 38400 for 68562-based synchronous interfaces (see the *Hardware Reference* for more details). The nearest possible value to the specified baud rate is used. The output from the **show syn** command gives the actual baud rate. If the interface is configured to receive clocks and the speed is set to a value greater than 64000 (or 38400), and then the transition cable is changed to generate clocks, the actual baud rate is reduced to 64000 (or 38400). When an RS-232 DCE transition cable is connected to a synchronous interface configured to generate clocks, the maximum clock speed is 38400 bps. When an X.21 DCE or V.35 DCE transition cable is connected to a synchronous interface configured to generate clocks, the maximum clock speed is 2 Mbps. This can be displayed with the **show syn** command, but the configured value for this interface is **not** changed. The default value for **speed** is 48000.

The **txclock** parameter specifies the edge of the transmit clock used to clock the transmit data out of the interface. The default value is **rising** and is appropriate for the majority of installations. At high baud rates it may be helpful to select **falling** if a high level of errors are occurring on the synchronous communications link. This setting may also be helpful at moderate baud rates if the cable from the synchronous interface to the NTU is long. Note that 68302 and 68562 hardware types support rising edge clocking only. The hardware type can be read from the output of the **show syn** command.

**Examples** To set the baud rate for synchronous interface 2 to 38400, use the command:

```
set syn=2 sp=38400
```

To set the DTR modem control output signal to always be off and the RTS modem control output signal to follow the state of the CTS modem control input signal, on a synchronous interface with a V.35 or RS-232 DTE transition cable, use the command:

```
set syn=0 dtr=off rts=cts
```

**Related Commands** [show syn](#)

## 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 11-4 on page 11-62](#), [Table 11-2 on page 11-63](#)).

The **counter** parameter displays counters from the specified categories ([Figure 11-5 on page 11-66](#), [Table 11-3 on page 11-66](#)). 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 11-6 on page 11-67](#)). The command history can also be displayed with the [show command history command on page 2-16 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 11-7 on page 11-67](#), [Table 11-4 on page 11-67](#)).

Figure 11-4: Example output from the **show asyn** command

```

ASYN 0 : 0000005625 seconds  Last change at: 0000005606 seconds

ASYN information
Name ..... Asyn 0
Status ..... enabled
Mode ..... PPP
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
Attached module ..... ASYN Call Control
Attached module instance .. 2
Type ..... VT100
Service ..... none
Prompt ..... login
Echo ..... yes
Attention ..... break
Manager ..... no
Edit mode ..... insert
History length ..... 20
Page size ..... 22
Idle Timeout (seconds)..... 300

```

Table 11-2: 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 11-2: 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 <b>set asyn</b> 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 11-2: 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 11-5: 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 11-3: 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 11-3: 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 11-6: 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 11-7: Example output from the **show asyn summary** command.

```

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

```

Table 11-4: 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.
Service	The name of the service to which the port is allocated, if any.

**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 service](#)
- [set tty](#)
- [show tty](#)

## show ds3 cmtce

**Syntax** SHow DS3[=*instance*] CMTCe

where *instance* is the number of the DS3 interface

**Description** This command displays the ASCII text data transmitted, received, and expected in the C-bit maintenance path ([Figure 11-8](#), [Table 11-5](#)).

Figure 11-8: Example output from the **show ds3 cmtce** command

```
C-bit maintenance path data for DS3 instance 0:
C-bit maintenance path is switched off
Transmit:
  PID:
    EIC: tx eic pid  LIC: tx lic pid  FIC: tx fic pid  UNIT: txunit
    FAC(FI): This is the FI for the transmit PID
  ISID:
    EIC: txeicisid  LIC: txlicisid  FIC: ignore  UNIT: unit2
    PORT: This is the ISID port
  TSID:
    EIC: txtsideic  LIC: txtsidlic  FIC: txtsidfic  UNIT: unit3
    GEN: This is the transmit TSID GENNo
Receive:
  PID:
    EIC: ignore  LIC: ignore  FIC: ignore  UNIT: ignore
    FAC(FI): ignore
  ISID:
    EIC: ignore  LIC: ignore  FIC: ignore  UNIT: ignore
    PORT: ignore
  TSID:
    EIC: ignore  LIC: ignore  FIC: ignore  UNIT: ignore
    GEN: ignore
Expected:
  PID:
    EIC: RXEICPID  LIC: RXLICPID  FIC: RXFICPID  UNIT: unit4
    FAC(FI): Thhis is the receive PID FI g
  ISID:
    EIC: RXeicISID  LIC: RXLicISID  FIC: RXficISID  UNIT: unit5
    PORT: This is the receive ISIS port
  TSID:
    EIC: ignore  LIC: ignore  FIC: ignore  UNIT: ignore
    GEN: Receive TSID GenNo
```

Table 11-5: Parameters in output of the **show ds3 cmtce** command

Test	Function
Transmit	The ASCII characters displayed in the transmit section are those sent from the switch to the far end over the C-bit maintenance path. These characters are entered by the user using the SET DS3 command. By default, the switch transmits the characters "ignore".
Receive	The ASCII characters displayed in the receive section are those received over the C-bit maintenance path, from the far end. If the C-bit maintenance path is switched off, the Receive section displays "ignore".
Expected	The ASCII characters displayed in the expected section are those the user is expecting to receive from the far end over the C-bit maintenance path. These characters are entered by the user using the <b>set ds3</b> command and can be manually compared by the user with the characters in the receive section (above).
PID	Path IDentification. A message in the C-bit maintenance channel that comprises EIC, LIC, FIC, UNIT and FAC (see below).
ISID	Idle Signal IDentification. A message in the C-bit maintenance channel that comprises EIC, LIC, FIC, UNIT and PORT (see below).
TSID	Test Signal IDentification. A message in the C-bit maintenance channel that comprises EIC, LIC, FIC, UNIT and GEN (see below).
EIC	Equipment Identification Code. This element describes a specific piece of equipment, containing 0 to 10 ASCII characters.
LIC	Location Identification code. This element describes a specific location, containing 0 to 11 ASCII characters.
FIC	Frame Identification Code. This element describes a specific location within a building, containing 0 to 10 ASCII characters.
UNIT	This element describes a specific location within an equipment bay, containing 0 to 6 ASCII characters.
FAC(FI)	FACility identification code. This element should uniquely describe the DS3 signal, containing 0 to 38 characters.
PORT	This element describes the equipment port that originates the IDLE signal, containing 0 to 38 characters.
GEN	This element describes the port that originates the test signal, containing 0 to 38 characters.

## show ds3 configuration

**Syntax** `SHoW DS3 [=instance] CONfiguration`

where *instance* is the number of the DS3 interface

**Description** This command displays the higher layer modules (if any) that have been attached to the DS3 interface (Figure 11-9, Table 11-6). The interface number is optional. If an interface number is not specified, then the attached modules for all DS3 interfaces are displayed.

Figure 11-9: Example output from the **show ds3 configuration** command

```
Configuration for DS3 instance 0:
Module ..... PPP
Module instance identifier .. 0
```

Table 11-6: Parameters in output of the **show ds3 configuration** command

Parameter	Meaning
Module	The higher layer module attached to the DS3 interface.
Module instance identifier	The instance number of the higher layer module instance attached to the DS3 interface.

**Examples** To display the configuration of ds3 interface 0, use the command:

```
sh ds3=0 conf
```

**Related Commands** [show ds3 counters](#)  
[show ds3 state](#)

## show ds3 counters

---

**Syntax** `SHoW DS3 [=instance] COUnTERS [{HDLc | INTeRface | LIInk |  
DIAGnostic}] [HIStory [=interval]] [{NEAR | FAR | BOTH}]`

where:

- *instance* is the number of the DS3 interface
- *interval* is a number from 1 to 96 identifying a 15 minute period from the last 24 hours. If the switch has been rebooted in the last 24 hours, then not all interval numbers are valid.

**Description** This command displays the MIB counters associated with the DS3 interface. If the interface number is not specified, the counters for all DS3 interfaces are displayed. If **hdlc** is specified, the counters stored in the enterprise MIB relevant to a DS3 interface are displayed ([Figure 11-10 on page 11-73](#), [Table 11-7 on page 11-73](#)). If **interface** is specified the counters from the interfaces table of the interfaces MIB relating to the DS3 are displayed ([Figure 11-11 on page 11-74](#), [Table 11-8 on page 11-75](#)). If **link** is specified, the counters stored in the enterprise MIBs related to the performance of the link during the current 15 minute interval and over the past 24 hours are displayed ([Figure 11-12 on page 11-76](#), [Table 11-9 on page 11-77](#)). If **diagnostic** is specified, diagnostic counters for the DS3 interface hardware and driver software are displayed ([Figure 11-13 on page 11-78](#)).

If a counter category is not specified, all counters except the link history and diagnostic counters are displayed.

The **history** parameter specifies within which 15-minute period to view the counters. If a value between 1 and 96 is specified, the 15-minute counters for that interval are displayed. The **history** parameter is valid when **counters=link**. If no interval is specified, all 15-minute counters are displayed.

The **near** parameter specifies that NEAR end counters are displayed. The **near** parameter is valid when **counters=link**.

The **far** parameter specifies that far end counters are displayed. The **far** parameter is valid when **counters=link**.

The **both** parameter specifies that both near and far end counters are displayed. The **both** parameter is valid when **counters=link**.



Figure 11-10: Example output from the **show ds3 counters=hdlc** command

HDLC Counters			
Receive:		Transmit:	
ReadyQueueLength	0	ReadyQueueLength	0
QueueStateErrors	0	QueueStateErrors	0
Frames	0	Frames	0
UnderlengthFrames	0	MalformedFrames	0
OverlengthFrames	0		
NonOctetAligneds	0	Underruns	0
CRCErrors	0	PartialFrames	0
HDLCAborts	0	UnprovisionedChannels	0
		LostInterrupts	0
Overruns	0		
FIFOOverflows	0		
UnprovisionedChannels	0		
		General:	
Misseds	0	BusErrors	0
TooFewBuffers	0	Recovers	0

Table 11-7: Parameters in output of the **show ds3 counters=hdlc** command

Parameter	Meaning
DS3 instance	The instance number of the DS3 interface.
seconds	The current value of sysUpTime.
Last change at	The value of sysUpTime at the time the interface entered its current operational state.
ReadyQueueLength	The number of frames in the hardware receive or transmit queues.
QueueStateErrors	The number of times the receiver or transmitter hardware attempted to write to a full queue or read from an empty queue.
Frames	The number of frames received or transmitted.
UnderlengthFrames	The number of received frames discarded because they were too short.
OverlengthFrames	The number of overlength frames received.
NonOctetAligneds	The number of non-octet aligned frames received, possibly due to noise on the line.
CRCErrors	The number of frames received with a CRC error.
Aborts	The number of received frames terminated with an abort, possibly due to noise on the line.
Overruns	The number of received frames truncated during DMA from the hardware due to a FIFO overflow.
FIFOOverflows	The number of frames lost because the FIFO buffer in the HDLC controller has overflowed.
UnprovisionedChannels	The number of times the reception or transmission of a frame was interrupted by a channel becoming unprovisioned.
ReceiveErrors	The number of receive errors reported from the receive DMA controller. Such errors do not include malformed or corrupted frames, and may indicate a hardware error.

Table 11-7: Parameters in output of the **show ds3 counters=hdlc** command (Continued)

Parameter	Meaning
Misseds	The number of receive frames lost because of a lack of receive buffers, possibly due to HDLC controller or switch overload.
TooFewBuffers	The number of received frames discarded because the number of buffers in the switch had reached a critical level.
MalformedFrames	The number of times the transmitter was given a malformed frame (e.g. bytes in buffer field set to 0) to transmit.
Underruns	The number of times a frame had to be retransmitted due to a transmitter underrun, possibly due to HDLC controller or switch bus overload.
PartialFrames	The number of times the transmitter was given an incomplete frame to transmit.
LostInterrupts	The number of times the transmission or reception of a frame had to be aborted due to no transmit/receive interrupt being received. Usually indicates a hardware malfunction.
BusErrors	The number of times a system or parity error was detected on the PCI bus.
Recovers	The number of times the HDLC controller was reset due to a serious error or a <b>reset ds3</b> command.

Figure 11-11: Example output from the **show ds3 counters=interface** command

DS3 instance 0: 24 Seconds Last change at: seconds			
Interface MIB Counters			
Receive:		Transmit:	
ifInOctets	0	ifOutOctets	0
ifInUcastPkts	0	ifOutUcastPkts	0
ifInNUcastPkts	0	ifOutNUcastPkts	0
ifInDiscards	0	ifOutDiscards	0
ifInErrors	0	ifOutErrors	0
ifInUnknownProtos		ifOutQLen	0

Table 11-8: Parameters in output of the **show ds3 counters=interface** command

Counter	Meaning
DS3 instance	The instance number of the DS3 interface.
Seconds	The current value of sysUpTime.
Last change at	The value of sysUpTime at the time the interface entered its current operational state.
ifInOctets	The number of octets received on this interface.
ifInUcastPkts	The number of unicast frames delivered to a higher layer protocol.
ifInNUcastPkts	The number of non-unicast frames delivered to a higher layer protocol.
ifInDiscards	The number of inbound frames discarded, though no errors had been detected to preventing them from being deliverable to a higher layer protocol. Usually due to an interface reset by command or after a hardware overload or malfunction.
ifInErrors	The number of inbound frames that contained errors (e.g CRC errors) preventing them from being deliverable to a higher-layer protocol.
ifInUnknownProtos	The number of frames discarded because they were for an unconfigured protocol. Neither relevant nor incremented for DS3 interfaces.
ifOutOctets	The number of octets transmitted, including framing.
ifOutUcastPkts	The number of unicast frames transmitted or discarded.
ifOutNUcastPkts	The number of non-unicast frames transmitted or discarded.
ifOutDiscards	The number of frames discarded, though no errors had been detected preventing their being transmitted. Usually due to output queue limiting.
ifOutErrors	The number of frames not transmitted because of errors. Usually due to a channel reset or shutdown, or because of a hardware malfunction.
ifOutQLen	Length of output frame queue.

Figure 11-12: Example output from the **show ds3 counters=link** command

```
Manager > sho ds3=0 counters=link

Link Counters for DS3 interface.....: 0
Elapsed seconds in current interval...: 74
Number of complete 15 minute intervals: 0

Current Interval count for instance: 0

Near End:
  P-bit Errored Seconds.....: 20
  P-bit Severely Errored Seconds...: 0
  Severely Errored Framing Seconds: 0
  UnAvailable Seconds.....: 0
  Line Coding Violations.....: 47
  P-bit Coding Violations.....: 28
  Line Errored Seconds.....: 33
  C-bit Coding Violations.....: 26
  C-bit Errored Seconds.....: 20
  C-bit Severely Errored Seconds...: 0

Far End:
  Time Elapsed.....: 74
  Valid Intervals.....: 0
  C-bit Errored Seconds.....: 1
  C-bit Severely Errored Seconds...: 0
  C-bit Coding Violations.....: 9
  UnAvailable Seconds.....: 0

Total count for last 24 hour period:

Near End:
  P-bit Errored Seconds.....: 0
  P-bit Severely Errored Seconds...: 0
  Severely Errored Framing Seconds.: 0
  UnAvailable Seconds.....: 0
  Line Coding Violations.....: 0
  P-bit Coding Violations.....: 0
  Line Errored Seconds.....: 0
  C-bit Coding Violations.....: 0
  C-bit Errored Seconds.....: 0
  C-bit Severely Errored Seconds...: 0

Far End:
  C-bit Errored Seconds.....: 0
  C-bit Severely Errored Seconds...: 0
  C-bit Coding Violations.....: 0
  UnAvailable Seconds.....: 0

Manager >
```

Table 11-9: Parameters in output of the **show ds3 counters=link** command

Parameter	Meaning
DS3 Instance	The instance number of the DS3 interface
Elapsed Seconds	The total number of seconds elapsed in the current 15 minute interval.
Number of Complete 15-minute intervals	The number of 15-minute intervals elapsed since the switch was restarted. Once this value gets to 96 it stays at 96 to show that there is a complete 24 hour period of performance data.
P-Bit Errored Seconds	The number of one second intervals during which one or more P-bit parity errors or AIS defects occurred.
P-bit Severely Errored Seconds	The number of one second intervals during which 45 or more P-bit parity errors or AIS defects occurred.
Severely Errored Framing Seconds	A second with one or more Out Of Frame defects or a detected incoming AIS.
Unavailable Seconds	The number of one second intervals during which the DS3 Path was unavailable. The link becomes unavailable at the onset of 10 continuous Severely Errored Path Seconds and becomes available again at the onset of 10 contiguous seconds with no Severely Errored Path Seconds.
Line Coding Violations	The total number of bi-polar violations and excessive zero errors encountered on the receive signal.
P-bit Coding Violations	The number of P-bit Parity errors.
Line Errored Seconds	The number of one-second intervals during which one or more Bi-polar violations (BPVs), or one or more excessive zeroes (EXZs), or one or more LOS defects occur.
C-bit Coding Violations	The number of CP-bit parity errors.
C-Bit Errored Seconds	The number of one second intervals during which one or more CP-bit parity errors or AIS defects occurred. This counter is available when C-bit parity is enabled.
C-Bit Severely Errored Seconds	The number of one second intervals during which 45 or more CP-bit parity errors or AIS defects occurred. This counter is available when C-bit parity is enabled.

Figure 11-13: Example output from the **show ds3 counters=diagnostic** command

DS3 instance 0: 30 Seconds		Last change at: seconds	
Device Dependent Diagnostic Counters (FREEDM-32 hardware)			
Hardware queue values:			
numRPDs	126	numTDs	126
rxFreeQueue length	126	txFreeQueue length	126
rxReadyQueue length	0	Confirmed length	126
		Unconfirmed length	0
		txReadyQueue length	0
Interrupt counters:			
TFUDRI: txUFlowCnt	0	RPQLFI: rxPdlfqcrCnt	0
IOCI: txIocCnt	0	RPQSFI: rxPdsfqcrCnt	0
TDFQEI: txDfqeCnt	0	RFOVRI: rxOFlowCnt	0
TDQRDYI: txDfqcrCnt	1	RPFEI: rxPktFormErrCnt	0
TDQFI: txDFqwCnt	1	RABRTI: rxAbortCnt	0
RPDRQEI: rxPdrqeCnt	0	RFCSEI: rxFcseCnt	0
RPDFQEI: rxPdfqeCnt	0	PERRI: parityErrCnt	0
RPQRDYI: rxPDrqwCnt	0	SERRI: sysErrCnt	0

1

Table 11-10: Parameters in output of the **show ds3 counters=diagnostic** command

Parameter	Meaning
DS3 instance	The instance number of the DS3 interface.
Seconds	The current value of sysUp Time.
Last change at	The value of sysUp Time at the time the interface entered its current operational state.
numRPDs	The number of receive buffer descriptors used by the DS3 receiver hardware.
rxFreeQueue length	The number of receive buffers that are empty and available for use.
rxReadyQueue length	The number of receive buffers containing unprocessed received frames.
numTDs	The number of transmit buffer descriptors used by the DS3 transmitter hardware.
txFreeQueue length	The number of transmit that are not awaiting transmission.
Confirmed length	The number of transmit buffers that are empty and available for use.
Unconfirmed length	The number of transmit buffers that have been transmitted but not yet marked as empty and available for use.
txReadyQueue length	The number of transmit buffers containing frames ready for transmission.
TFUDRI: txUFlowCnt	The number of transmit FIFO underrun interrupts that have occurred.
IOCI: txIocCnt	The number of transmission complete interrupts that have occurred.
TDFQEI: txDfqcCnt	The number of transmit free queue overflow interrupts that have occurred.

Table 11-10: Parameters in output of the **show ds3 counters=diagnostic** command

Parameter	Meaning
TDQRDYI: txDfgcrCnt	The number of transmit ready queue cache read interrupts that have occurred.
RPDRQEI: rxPdrqeCnt	The number of receive ready queue overflow interrupts that have occurred.
RPDFQEI: rxPdfqeCnt	The number of receive free queue underflow interrupts that have occurred.
RPQRDYI: rxPDrqwCnt	The number of receive ready queue write interrupts that have occurred.
RPQLFI: rxPdlfqcrCnt	The number of receive large-buffer free queue cache read interrupts that have occurred.
RPQSFI: rxPdsfqcrCnt	The number of receive small-buffer free queue cache read interrupts that have occurred.
RFOVRI: rxOFlowCnt	The number of receive FIFO overrun interrupts that have occurred.
RPFEI: rxPktFormErrCnt	The number of receive packet format error interrupts that have occurred.
RABRTI: rxAbortCnt	The number of receive abort interrupts that have occurred.
RFCSEI: rxFcseCnt	The number of receive frame CRC error interrupts that have occurred.
PERRI: parityErrCnt	The number of PCI-bus parity error interrupts that have occurred. These interrupts indicate a hardware malfunction.
SERRI: sysErrCnt	The number of PCI-bus system error interrupts that occurred. These interrupts indicate a hardware malfunction.

**Examples** To display the counters for the DS3 interface:

```
sh ds3=0 cou
```

**Related Commands**

- [set ds3](#)
- [show ds3 debug](#)
- [show ds3 state](#)
- [show ds3 utilisation](#)

## show ds3 debug

**Syntax** SHow DS3 [=instance] DEBug

where *instance* is the number of the DS3 interface

**Description** This command displays the setting of the debug options for the DS3 interface. If an interface is not specified, the settings for all DS3 interfaces are displayed (Figure 11-14, Table 11-11).

Figure 11-14: Example output from the **show ds3 debug** command

```
Manager > sho ds3 debug

Debug switches for DS3 instance 0:

Link ..... no
Packet ..... no
```

Table 11-11: Parameters in output of the **show ds3 debug** command

Parameter	Meaning
Link	The significant layer 1 events.
Packet	The contents of the data packets transmitted or received by the DS3 interface.

**Examples** To display the state of debugging options for DS3 interface 0, use the command:

```
sh ds3=0 deb
```

**Related Commands** [disable ds3 debug](#)  
[enable ds3 debug](#)



## show ds3 state

**Syntax** SHow DS3 [=instance] STAtE

where *instance* is the number of the DS3 interface

**Description** This command displays information about the current state of the DS3 interface and of any potential alarms on the interface. If an interface is not specified, the state of all DS3 interfaces is displayed ([Figure 11-15](#) [Table 11-12](#) on page 11-82).

Figure 11-15: Example output from the **show ds3 state** command

```

State for DS3 instance 0:

Line Type ..... C-Bit Parity
Line Status..... Alarm
  RX RAI ..... no
  TX RAI ..... yes
  RX AIS ..... no
  TX AIS ..... no
  RX LOF ..... no
  RX LOS ..... yes
  RX Loopback ..... no
  Near end Unavailable ..... no
  Equipment Out of Service ..... no

Status Last Changed..... 250
Clock Source ..... Loop Timed
Loopback Status ..... No Loopbacks
FDET ..... on

C-Bit Mtce. Channel Enabled .... on

dsx3LineStatusTrap ..... enabled
Threshold Crossing Alert ..... enabled

Receiving IDLE Code ..... no

Threshold Crossing Alert Thresholds      15 Min      24 hour
PES PSES SEFS UAS LES CES CSES           10        65535
LCV PCV CCV                             16383      1048575

Defect:      Defect Count      Failures      This Second      Last Second
LOS :        1                1                0                0
AIS :        0                0                0                0
OOF :        0                0                0                0
FERF :       0                -                0                0
RAI :        -                0                -                -

FEAC:
Transmit: Remote LOS
Receive : No Activity

```

Table 11-12: Parameters in output of the **show ds3 state** command

Parameter	Meaning
Line Type	The type of DS3 interface. "C-bit parity" is the only type supported.
Line Status	Shows the current status of the interface. When no alarms or loopbacks are active, this shows "No Alarm". When alarms or loopbacks are present, then "Alarm" is displayed and the next 9 lines show the status of each individual status field.
RX RAI	Displayed when the Line Status has a valid alarm. Shows 'yes' when the interface is currently receiving a Remote Alarm Indication. This happens when the far end is in a LOS, LOF or AIS condition.
TX RAI	Displayed when the Line Status has a valid alarm. Shows "yes" when the interface is currently transmitting a Remote Alarm Indication. This happens when the interface is in an alarm failure condition (either LOS, LOF or AIS).
RX AIS	Displayed when the Line Status has a valid alarm. Shows "yes" when the interface is currently receiving an Alarm Indication Signal. This normally shows that the interface at the far end is not configured.
TX AIS	Displayed when the Line Status has a valid alarm. Shows "yes" when the interface is currently transmitting an Alarm Indication Signal). In the switch, this happens when a user switches AIS on with the <a href="#">enable ds3 test command on page 11-37</a> .
RX LOF	Displayed when the Line Status has a valid alarm. Shows "yes" when the interface is currently in a Loss Of Frame condition. This happens when there is excessive noise on the line such that the framing bit patterns in the DS3 signals overhead are corrupted.
RX LOS	Displayed when the Line Status has a valid alarm. Shows "yes" when the interface is currently in a Loss Of Signal condition. This happens when the connection to the DS3 equipment has been broken or the far end equipment switched off.
Rx Loopback	Displayed when the Line Status has a valid alarm. Shows "yes" when the interface is currently in a loopback condition. See <a href="#">"Loopbacks" on page 11-16</a> for a description of the loopbacks.
Near End Unavailable	Displayed when the Line Status has a valid alarm. Shows "yes" when the interface is not available. This is always "no" in the switch as the DS3 interface is always available when it is present in the switch.
Equipment Out Of Service	This is always "no" in the switch. The DS3 interface is always in service when it is present in the switch.
Status Last Changed	Shows the number of seconds since the last time the line Status was updated.
Clock Source	Displays "Internal Timing" when the interface retrieves its clock from an internal oscillator. Displays "Loop Timed" when the interface retrieves its clock from the incoming DS3 signal. This is defined with the <a href="#">set ds3 command on page 11-53</a> .

Table 11-12: Parameters in output of the **show ds3 state** command (Continued)

Parameter	Meaning
Loopback Status	Displays the loopback that is enabled. See <a href="#">“Loopbacks” on page 11-16</a> for a description of the loopbacks.
FDET	Displays when fast detection of AIS is enabled. If yes, then AIS is detected in 2.23ms. If no, then AIS is detected in 13.5ms. This is defined with the <a href="#">set ds3 command on page 11-53</a> .
C-Bit Mtce. Channel Enabled	Displays when the C-bit maintenance channel is enabled. This channel allows various ASCII characters to be passed to the far end. These ASCII characters uniquely identify the interface.
PID	Path Identification.
ISID	Idle Signal Identification.
TSID	Test Signal Identification.
Transmit	This is the C-bit maintenance path data the near end transmits to the far end.
Receive	This is the C-bit maintenance path data the near end is receiving from the far end.
Expected	This is the C-bit maintenance path data the near end is expecting to receive from the far end.
EIC	Equipment Identification Code. This 10 character ASCII code is used to describe a specific piece of equipment. This code is valid if the C-Bit Mtce Channel is enabled.
LIC	Location Identifier. This 11 character ASCII code describes the location of the equipment.
FIC	Frame Identification Code. This 10 character ASCII code identifies where the equipment is located in the building.
UNIT	This 6 character code identifies the equipment location within a bay.
PID	This 38 character string uniquely identifies the DS3 signal.
Defect	Shows counts for the LOS, LOF, AIS and FERF alarms.
Defect Count	Count of the number of times a defect is observed. This counter is incremented even if the defect exists for a short period of time (in other words if the hardware detects the defect this counter is incremented). This counter is reset to zero if the switch is restarted or if the counters are reset using the RESET DS3 command.
Failures	Count of the number of times a failure is observed. A failure is a defect that persists for 2.5 seconds. This counter is reset to zero if the switch is restarted or if the counters are reset using the RESET DS3 command.
This Second	Count of the number of defects in the current one second period. This count is reset to zero every second.
Last Second	Count of the number of defects in the last one second period. This count is set to the This Second count every second.
Transmit	Displays the code that is currently being transmitted on the FEAC channel

Table 11-12: Parameters in output of the **show ds3 state** command (Continued)

Parameter	Meaning
Receive	Displays the code that is currently being received on the FEAC channel.

**Examples** To display the state of DS3 interface 0, use the command:

```
sh ds3=0 sta
```

**Related Commands**

- [set ds3](#)
- [show ds3 configuration](#)
- [show ds3 counters](#)
- [show ds3 debug](#)

## show ds3 test

**Syntax** `SHOW DS3 [=instance] TEST`

where *instance* is the number of the DS3 interface

**Description** This command displays the settings of the test options for the DS3 interface (Figure 11-16, Table 11-13). If an interface is not specified, settings for all DS3 interfaces are displayed.

Figure 11-16: Example output from the **show ds3 test** command

```
Test Switches for DS3 instance 0:
Number Action                               States
-----
 1   Line Loopback enabled                  no
 2   Payload Loopback enabled              no
 3   Diagnostic Loopback enabled           no
 4   Remote Loopback enabled               no
 5   TX PRGD Pattern                       yes
 6   TX TSIG Pattern                      no
 7   TX Far End Receive Failure            no
 8   IDLE Signal                          no
 9   Alarm Indication Signal               no
10   Simulate Loss of Signal               no
11   Bit Error                            yes : Bit error rate = 10E-4
20   HDLC Diagnostic Loopback              no
23   HDLC Line Loopback                   no

PRGD Test:
Sync state: In Sync
BER count: 23692
Number of times sync has changed: 1
```

Table 11-13: Parameters in output of the **show ds3 test** command

Parameter	Meaning
Number	Number entered by the user to invoke the test.
Action	Description of the test.
States	Whether the test is enabled or disabled.

**Examples** To display the tests running on DS3 interface 0, use the command:

```
sh ds3=0 test
```

**Related Commands** [disable ds3 test](#)  
[enable ds3 test](#)  
[show ds3 state](#)

## show ds3 trap

---

**Syntax** SHow DS3 [=instance] TRap

where *instance* is the number of the DS3 interface

**Description** This command displays the status of the DS3 interface traps. If an interface is not specified, the status for all DS3 interfaces are displayed (Figure 11-17).

Figure 11-17: Example output from the **show ds3 trap** command

```
Manager > sho sd3 trap

Trap switches for DS3 instance instance 0:

dsx3LineStatusTrap ..... enabled
Threshold Crossing Alert. enabled
```

**Example** To display the status of traps for DS3 interface 0, use the command:

```
sh ds3=0 tr
```

**Related Commands** [disable ds3 trap](#)  
[enable ds3 trap](#)

## show ds3 utilisation

---

**Syntax** SHow DS3 [=instance] UTILisation

where *instance* is the number of the DS3 interface

**Description** This command displays the link bandwidth of the DS3 interface and the percentage of link bandwidth in use in the receive and transmit directions. If an interface is not specified, utilisation is displayed for all DS3 interfaces (Figure 11-18).

Figure 11-18: Example output from the **show ds3 utilisation** command

```
Utilisation for DS3 instance 0:

Bandwidth (bits/sec) , , , , , , 44209694
Receive utilisation ..... 19 %
Transmit utilisation ..... 92 %
```

**Example** To show how much of the bandwidth of DS3 interface 0 is in use, use the command:

```
sh ds3=0 util
```

**Related Commands** [show ds3 counters](#)

## 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 11-19 on page 11-87, Table 11-14 on page 11-87). If an interface is specified, detailed information is displayed about it including the counters (Figure 11-20 on page 11-88, Table 11-15 on page 11-88).

The **counters** parameter displays interface counters for all interfaces (Figure 11-21 on page 11-89, Table 11-16 on page 11-89). 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 11-19: 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 11-14: 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.

Table 11-14: Parameters in output of the **show interface** command (Continued)

Parameter	Meaning
Number of unencrypted PPP/FR links	Total number of PPP interfaces and Frame Relay circuits that are configured to send plaintext. Does not include disabled PPP interfaces or enabled Frame Relay circuits on disabled Frame Relay interfaces.
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 11-20: 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 11-15: 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.



Table 11-15: Parameters in output of the **show interface** command for a specific interface (Continued)

Parameter	Meaning
Interface Counters	Counters for 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.

Figure 11-21: Example output from the **show interface counter** command

Interface Counters	
Interface: vlan1	
ifInOctets .....	22852
ifInUcastPkts .....	184
ifInNUcastPkts .....	19
ifInDiscards .....	0
ifInErrors .....	0
ifInUnknownProtos ...	30
ifOutOctets .....	15565
ifOutUcastPkts .....	148
ifOutNUcastPkts .....	0
ifOutDiscards .....	0
ifOutErrors .....	0
Interface: ISDN Basic Rate Interface	
ifInOctets .....	0
ifInUcastPkts .....	0
ifInNUcastPkts .....	0
ifInDiscards .....	0
ifInErrors .....	0
ifInUnknownProtos ...	0
ifOutOctets .....	0
ifOutUcastPkts .....	0
ifOutNUcastPkts .....	0
ifOutDiscards .....	0
ifOutErrors .....	0

Table 11-16: 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.

Table 11-16: Parameters in output of the **show interface counter** command (Continued)

Parameter	Meaning
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](#)

## show syn

**Syntax** SHow SYN[=*n*]

where *n* is the number of the synchronous interface

**Description** This command displays the configuration of a synchronous interface and the state of the interface control signals (Figure 11-22, Table 11-17). If an interface is not specified, then information is displayed for all interfaces.

Figure 11-22: Example output from the **show syn** command

SYN instance 1:			
	3088 seconds	Last change at:	0 seconds
Module .....	PPP		
State .....	enabled		
Active .....	yes		
Interface type .....	RS-232 DTE		
Clocks .....	receive		
Actual baud rate .....	determined externally		
Configured baud rate .....	48000		
Max output queue length ...	100		
Min interframe delay .....	no delay		
Date sense .....	normal		
Tx clock edge .....	rising		
Hardware type .....	68302		
Debug .....	on		
Control signals	State	Output mode	Transitions
CTS (in) .....	off		8
DCD (in) .....	off		2
DSR (in) .....	off		6
RTS (out) .....	off	follow CTS	6
LL (out) .....	off	layer 2 control	6
RL (out) .....	off		4

Table 11-17: Parameters in output of the **show syn** command

Parameter	Meaning
Module	Name of the layer 2 module attached to this interface, if any.
State	Whether the interface is enabled.
Active	Status of the interface. Yes if enabled and a layer 2 module attached.
Interface type	Type of the transition cable attached to the interface.
Clocks	Direction of the interface clocks, receive or generate.
Actual baud rate	Actual baud rate of the interface, may be different from configured baud rate for generate clocks.
Configured baud rate	Baud rate configured by the <a href="#">set syn</a> command on page 11-58 (or the default).
Max output queue length	Maximum output queue length configured by the <a href="#">set syn</a> command (or the default).
Min interframe delay	Minimum transmit interframe delay configured by the <a href="#">set syn</a> command (or the default).
Date sense	Whether the sense of the transmitted data is normal or inverted.

Table 11-17: Parameters in output of the **show syn** command (Continued)

Parameter	Meaning
Tx clock edge	Transmit clock edge used to clock out the transmitted data; either rising or falling.
Hardware type	Whether the hardware device for the interface is 68302, 68360, or 68562.
Debug	Whether debug messaging is on.
Control signals	Modem control signals available for the current interface type.
State	Whether the configured state of the modem control signals is on, off, not monitored, or not controlled.
Output mode	Control mode for modem control output signals; either "always off", "always on", "layer 2 control" and "follow XXX" where "XXX" is the name of the paired modem control input signal.
Transitions	Number of transitions the modem control signal has made.

**Related Commands**   [show syn counter](#)  
[set syn](#)

## show syn counter

**Syntax** `SHoW SYN[=n] COUnTer[={INTErface|SYN}]`

where *n* is the number of the synchronous interface

**Description** This command displays the MIB counters for a synchronous interface. The interface number is optional. If an interface is not specified, then information is displayed for all interfaces. If a category is not specified, all counters are displayed.

If **interface** is specified, counters from the interfaces table from the interface MIB are displayed (Figure 11-23, Table 11-18).

If **syn** is specified, the counters from the synchronous interface table of the switch's enterprise MIB and the counters from the synchronous port table of the RS-232-like hardware devices MIB are displayed (Figure 11-24 on page 11-94, Table 11-19 on page 11-95).

Figure 11-23: Example output from the **show syn counter=interface** command

SYN instance 0: 3098 seconds Last change at: 0 seconds			
Interface MIB Counters			
Receive:		Transmit:	
ifInOctets	21349408	ifOutOctets	18683238
ifInUcastPkts	65391	ifOutUcastPkts	34271
ifInNUcastPkts	0	ifOutNUcastPkts	0
ifInDiscards	0	ifOutDiscards	0
ifInErrors	93	ifOutErrors	0
ifInUnknownProtos	0	ifOutQLen	1

Table 11-18: Parameters in output of the **show syn counter=interface** command

Counter	Meaning
ifLastChange	The value of sysUpTime at the time the interface entered its current operational state.
ifInOctets	The number of octets received on this interface.
ifInUcastPkts	The number of unicast packets delivered to a higher-layer protocol.
ifInNUcastPkts	The number of non-unicast packets delivered to a higher-layer protocol.
ifExtnsMulticastsReceivedOKs	The number of frames successfully received for a multicast address other than a broadcast address.
ifExtnsBroadcastsReceivedOKs	The number of frames successfully received for a broadcast address other than a multicast address.
ifInDiscards	The number of inbound packets discarded without errors that prevented them from being deliverable to higher-layer protocol.
ifInErrors	The number of inbound packets with errors that prevented them from being deliverable to a higher-layer protocol.

Table 11-18: Parameters in output of the **show syn counter=interface** command

Counter	Meaning
ifInUnknownProtos	The number of packets discarded because they were for an unconfigured protocol.
ifOutOctets	The number of octets transmitted, including framing.
ifOutUcastPkts	The number of unicast packets transmitted or discarded.
ifOutNUcastPkts	The number of non-unicast packets transmitted or discarded.
ifExtnsMulticastsTransmittedOKs	The number of frames successfully transmitted to a multicast address other than a broadcast address.
ifExtnsBroadcastsTransmittedOKs	The number of frames successfully transmitted to a broadcast address other than a multicast address.
ifOutDiscards	The number of packets discarded though no errors had been detected preventing their being transmitted.
ifOutErrors	The number of packets not transmitted because of errors.
ifOutQLen	The length of the output packet queue.

Figure 11-24: Example output from the **show syn counter=syn** command

SYN instance 0: 1051 seconds Last change at: 0 seconds			
Synchronous Counters			
Receive:		Transmit:	
Frames	0	Frames	0
OverlengthFrames	0	CTSLosses	0
UnderlengthFrames	0	Underruns	0
CRCErrors	0	LostInterrupts	0
Aborts	0	DroppedFrames	0
NonOctetAligneds	0	NoPackets	0
Overruns	0	QueueLength	0
NonmatchAddresses	0	Recovers	0
Misseds	0	SDMABusErrors	0
TooFewBuffers	0	CommandTimeouts	0
QueueLength	0	LastCommand	0

Table 11-19: Parameters in output of the **show syn counter=syn** command for 68302- and 68360-based synchronous interfaces

Counter	Meaning
Frames	The number of frames received or queued for transmission by a higher layer.
OverlengthFrames	The number of frames discarded because they were too long.
UnderlengthFrames	The number of frames discarded because they were too short.
CRCErrors	The number of frames received with a CRC error.
Aborts	The number of received frames terminated by an abort.
NonOctetAligneds	The number of frames discarded because they did not contain an integral number of octets.
Overruns	The number of frames not received because of a receiver overrun.
NonmatchAddresses	The number of frames not received because the address of the HDLC frame was not one that was configured to be received.
Misseds	The number of frames that could not be received due to lack of a receive buffer.
TooFewBuffers	The number of received frames discarded because the number of buffers in the switch reached a critical level.
QueueLength	The number of frames on the receiver or transmitter queue.
CTSLosts	The number of frames during which the CTS input was negated.
Underruns	The number of frames not transmitted because a transmitter underrun occurred during transmission.
LostInterrupts	The number of frames that failed to transmit because no interrupt was received to signal transmission complete.
DroppedFrames	The number of frames discarded because the maximum transmit queue length was exceeded.
NoPackets	The number of times the 68302 or 68360 reported a transmit error, but there was no packet being transmitted or the errored packet could not be identified.
Recovers	The number of times the 68302 or 68360 was reset due to a serious error or a <a href="#">reset syn command on page 11-46</a> .
SDMABusErrors	The number of bus errors experienced by the 68302 or 68360 for this interface.
CommandTimeouts	The number of commands to the 68302 or 68360 that failed to complete.
LastCommand	The code of the command that was to be issued when a command timeout was detected.

**Related Commands** [reset syn counters](#)  
[show syn](#)

