Chapter 21

Generic Packet Classifier

Introduction

The classifier enables you to create packet matching rules - called *classifiers* - to sort packets into *data flows*. For example, you may want all packets with the same destination TCP/IP port to form a flow (e.g. all telnet or HTTP traffic). This chapter describes how to configure the classifier.

You can then configure the router to process all packets in a data flow in a given manner. You have two choices for acting on classified flows:

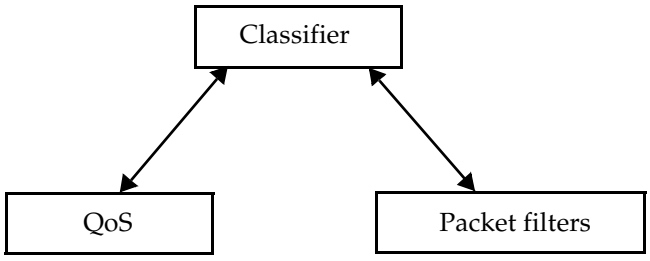
- Quality of Service (QoS).

  QoS prioritises packets and manages bandwidth. QoS is particularly useful for improving VoIP and video links, especially if your network is congested. Theory and configuration of QoS is described in Chapter 22, Quality of Service (QoS).

- Packet filters.

  Filters forward or discard packets. They can also modify the packet's priority settings, send the packet to a mirror port, and do other advanced actions. Classifier-based filters are described in "Classifier-Based Packet Filters" on page 8-34 of Chapter 8, Switching.

Figure 21-1: Interaction between the Classifier, QoS and filters.



Configuration of Classifiers

Configuring the classifier involves creating a set of packet matching rules, called *classifiers*, using the command:

```
create classifier=1..9999 [options]
```

These classifiers can identify any single packet based upon many criteria. Available criteria depend on the type of interface you use the classifier on, and include:

- Port or VLAN

  You can classify packets according to their ingress or egress port, or their destination VLAN.

- Ethernet encapsulation type

  You can classify packets depending on the specific protocol type of each frame. Different values indicate how the packet is formatted. For example, a value of 802.2 indicates the packet is formatted according to IEEE standards 802.2 and 802.3 with a Destination Service Access Point/Source Service Access Point (DSAP/SSAP) value not equal to AAAA in hexadecimal; SAP encapsulation. A value of ETHII indicates the packet is formatted according to RFC 894; Ethernet II encapsulation. For more details on values see the ETHFORMAT parameter in the **create classifier** command on page 21-4.

- Source/Destination MAC address

  You can classify frames from a specific source or destination MAC address. This classification can be used for users on remote networks. You can also specify MAC type to distinguish unicast packets from broadcast or multicast packets.

- Layer 3 protocols

  You can classify frames based on any value for Layer 3 protocols. Layer 3 protocol and Ethernet encapsulation types are interrelated, e.g. IPX Ethernet II encapsulated packets are different to IPX NETWARERAW encapsulated packets.

- DiffServ or IP TOS

  You can classify packets according to the value of the DSCP bits in the DiffServ field of the header, or the TOS precedence bits in the Type of Service (TOS). These fields are alternatives, so are mutually exclusive.

- Source/destination IP address

  You can classify packets based on an exact match of the source or destination IP address information within the IP header.

- IPX settings

  You can classify packets based on their destination IPX address, packet type and source or destination socket.

- Layer 4 protocol (TCP/UDP, ICMP etc.)

  You can classify packets based on specific Layer 4 TCP or UDP destination and source port numbers contained within the IP or IPv6 header.

- Layer 4 source/destination port and TCP flags

  You can classify packets based on a specific port number, and based on TCP flags.

- Up to three 16-bit words inside the first 64 bytes of a packet

  You can specify the bits to match, using the **match** parameter, and their position, using the **mask** and **offset** parameters.

Command Reference

This section describes the commands available to configure and manage the Classifiers.

See "Conventions" on page xxxviii of About this Software Reference in the front of this manual for details of the conventions used to describe command syntax. See Appendix A, Messages for a complete list of messages and their meanings.

create classifier

Syntax

```
CREATE CLASSIFIER=rule-id [VLAN={vlan-name|1..4094|ANY}]
   [EPOrt=portnum] [IPOrt=portnum] [ETHFormat={802.2|
   802.2-Tagged|802.2-Untagged|Ethii|ETHII-Tagged|
   ETHII-Untagged|Netwareraw|Snap|ANY}]
   [IPDaddr={ipaddmask|ANY}] [IPSAddr={ipaddmask|ANY}]
   [IPDScp={0..63|ANY}] [IPPRotocol={TCP|UDP|ICMp|IGMp|
   ipprotocolnum}] [IPTOS={0..7|ANY}] [IPXDAddr={ipxadd|
   ANY}] [IPXDSocket={NCP|SAP|RIP|NNB|DIAg|NLSp|IPXwan|
   ipxsocketnum|ANY}] [IPXSSocket={NCP|SAP|RIP|NNB|DIAg|
   NLSp|IPXwan|ipxsocketnum|ANY}] [IPXPacket={NLSp|RIP|
   SAP|SPX|NCP|NETbios|ipxpacketnum|ANY}]
   [MACDaddr={macadd|ANY}] [MACSaddr={macadd|ANY}][1]
   [MATCH1=hhh] [MASK1=hhh] [OFFSET1=0..62] [MATCH2=hhh]
   [MASK2=hhh] [OFFSET2=0..62] [MATCH3=hhh] [MASK3=hhh]
   [OFFSET3=0..62] [PROTocol={protocoltype|ANY}]
   [TCPDport={portid|ANY}] [TCPSport={portid|ANY}]
   [TCPFlags={{Urg|Ack|Rst|Syn|Fin}[,...]|ANY}]
   [UDPDport={portid|ANY}] [UDPSport={portid|ANY}]
```

where:

- *rule-id* is a decimal number from 1 to 9999.
- *vlan-name* is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits (0-9), the underscore character ("_"), and the hyphen (-). The *vlan-name* cannot be a number, **all** or **any**.
- *portnum* is the number of a physical port.
- *dscplist* is a single number or a group of numbers, either a comma separated list, a range (specified as n-m) or a combination of the two. Numbers start at 0 and end at 63.

1. Due to internal configuration settings, it may not be possible to specify Layer 2 MAC address-based packet matching rules in conjunction with Layer 3 packet matching rules. Given that Layer 3 IP/IPX and other Layer 3 data packet matching rules are more important than MAC address packet matching rules, this parameter may not be available. This applies to both the **macdaddr** and **macsaddr** parameters.

- *ipaddmask* is an IP address in dotted decimal notation with an optional mask. The optional mask is specified by adding "/M" following the IP address, where M is the number of contiguous bits in the mask; M ranges from 0 to 32 inclusive.
- *ipprotocolnum* is either an IP protocol number or a recognised IP protocol name. An IP protocol number is expressed as a 1 byte decimal number.
- *ipxadd* is an IPX network address expressed as a 4 byte hexadecimal number.
- *ipxsocketnum* is either an IPX socket number or a recognised IPX socket type. An IPX socket number is expressed as a 2 byte hexadecimal number.
- *ipxpacketnum* is either an IPX packet number or a recognised IPX packet type. An IPX packet number is expressed as a 2 byte hexadecimal number.
- *macadd* is an Ethernet six-octet MAC address, expressed as six pairs of hexadecimal digits delimited by hyphens.
- *hhh* is a 2 byte hexadecimal number.
- *protocoltype* is either a valid protocol number or a recognised protocol name. A protocol number can be either 1 byte for SAP, 2 bytes for ETHII or 5 bytes for an 802.3/802.2 SNAP type packet, and is specified in hexadecimal.
- *portid* is a TCP/IP or UDP/IP port number.

Description

This command creates a packet matching rule that identifies a particular data flow. This data flow may be specific in nature (e.g. IP packets with a particular TCP destination port), or it may be general (e.g. all IPX packets with a SNAP Ethernet packet encapsulation).

If a packet with an unknown destination port is to be transmitted, the packet is flooded to all ports in the VLAN. For such a packet, no hardware filters or QoS will be applied to the packet. This also applies to any broadcast or multicast IP or IPX packet.

The **classifier** parameter specifies a rule-id for the packet matching rule. This identifier is used to uniquely identify the rule. It does not imply an order between this rule and previously created rules.

The **vlan** parameter specifies the destination VLAN that the packet will be transmitted on. The default is **any**.

The **eport** parameter specifies the egress port on the switch to match for each frame. A classifier which matches traffic on the basis of egress port can only be applied to a port in the same port block as the egress port (ports 1 to 24 or 25 to 48 on a 48-port switch).

The **iport** parameter specifies the ingress port on the switch to match for each frame.

The **ethformat** parameter specifies the Ethernet encapsulation type of the packet. The parameter values and encapsulation types are detailed in Table 21-1 on page 21-6.

Table 21-1: Ethernet encapsulation types

Parameter Value	How the Packet is Formatted	Encapsulation Type
802.2	According to IEEE Standards 802.2 and 802.3 with a DSAP/SSAP value not equal to hexadecimal AAAA.	SAP
ETHII	According to RFC 894, *Standard for the transmission of IP datagrams over Ethernet networks.*	Ethernet II
NETWARERAW	As an IPX packet according to IEEE Standard 802.3. The IPX packet checksum is hexadecimal FFFF.	NetWare Raw or Novell
SNAP	According to IEEE Standards 802.2 and 802.3 and RFC 1042, *Standard for the transmission of IP datagrams over IEEE 802 networks.*	SNAP

If an **ethformat** option is specified (excluding **any**), the **protocol** parameter must also be used. Specifying **ethformat=snap protocol=ip** means using **protocol=0800**. Specifying **ethformat=snap protocol=ipx** means using **protocol=8137**. Specifying **ethformat=802.2 protocol=ipx** means using **protocol=e0**. In addition, the values **802.2-tagged**, **802.2-untagged**, **ethii-tagged** and **ethii-untagged** may be used to classify packets that were either tagged or untagged at ingress.

Table 21-2 shows the available **ethformat** and **protocol** parameter combinations and their implementation in the Classifier.

Table 21-2: Available **ethformat** and **protocol** parameter combinations

ethformat=	protocol=	classifier	basic chip
ETHII	[not specified]	OK	
	ANY	OK	
	NONIPIPX	OK	
	IP	OK (1)	
	IPX	OK (2)	
	protocoltype	OK	
NETWARERAW	[not specified]	OK (3)	
	ANY	OK (3)	
	NONIPIPX	Error	
	IP	Error	
	IPX	OK (3)	
	"IPX 802.3"	OK	
	protocoltype	Error	
SNAP	[not specified]	OK	
	ANY	OK	
	NONIPIPX	OK	
	IP	OK	
	IPX	OK	
	protocoltype	OK	

Table 21-2: Available **ethformat** and **protocol** parameter combinations (Continued)

ethformat=	protocol=	classifier	basic chip
802.2	[not specified]	OK	
	ANY	OK	
	NONIPIPX	OK	
	IP	Error	

Key to table

- [not specified] = the **protocol** parameter is not specified on the command line
- (1) = equivalent to specifying **protocol=0800**
- (2) = equivalent to specifying **protocol=8137**
- (3) = equivalent to specifying **protocol="ipx 802.3"**

The **ipdaddr** parameter specifies the destination IP address (either host or subnet) of an IP packet. The default is **any**.

The **ipsaddr** parameter specifies the source IP address (either host or subnet) of an IP packet. The default is **any**.

The **ipdscp** parameter specifies the Code Point bits of the DiffServ field of an IP packet (from 0 to 63, or **any**). This parameter cannot be specified in conjunction with the **iptos** parameter. The default is **any**.

The **ipprotocol** parameter specifies a Layer 4 IP protocol of an IP packet. If the command specifies a TCP/IP packet matching rule, (e.g. **tcpdport** is specified), then the default value for this parameter is TCP. If the command specifies a UDP/IP packet matching rule, (e.g. the **udpdport** parameter is specified), then the default value for this parameter is **udp**. The value of **nontcpudp** represents all Layer 4 IP protocols except for TCP and UDP. There can be 29 unique **ipprotocol** values specified across all Classifiers, not including TCP and UDP. If **icmp** is specified, it is considered as one of the 29 unique values. **igmp, tcp, udp, nontcpudp** and **any** are all system default values and are always available.

The **iptos** parameter specifies the value of the precedence field within the TOS byte of an IP packet. This parameter cannot be specified in conjunction with the **ipdscp** parameter. The default is **any**.

The **ipxdaddr** parameter specifies the destination network address of an IPX packet. The default is **any**.

The **ipxpacket** parameter specifies the value of the Packet Type field of an IPX packet. The default is **any**.

The **ipxdsocket** specifies the destination IPX socket number of an IPX packet. There can be 7 unique **ipxdsocket** values specified across all Classifiers. Specifying any of **ncp, sap, rip, nnb, diag, nlsp** or **ipxwan** counts as one of the seven user-specified values. The default is **any**.

The **ipxssocket** specifies the source IPX socket number of an IPX packet. There can be 7 user-specified **ipxssocket** values. Specifying any of **ncp, sap, rip, nnb, diag, nlsp** or **ipxwan** counts as one of the seven user-specified values. The default is **any**.

The **macdaddr** parameter specifies the destination MAC address of the packet. The default is **any**.

The **macsaddr** parameter specifies the source MAC address of the packet. The default is **any**.

The **match*x***, **mask*x*** and **offset*x*** parameters, where *x* is 1, 2 or 3, specify a general 16-bit word to match inside a packet. The three parameters must be specified at the same time on the command line. The **match** parameter specifies the actual data to match. The **mask** parameters specify whether the corresponding bit in the **match** parameter is "on" for a match or "don't care" for a match. If the **mask** bit is set (on), the bit in the **match** parameter must be the same as the corresponding bit in the actual packet (i.e. place "ones" in bit positions you want to match). If the **mask** bit is clear (don't care), the same bit in the **match** parameter will not be checked with the corresponding bit in the actual packet. The **offset*x*** parameters, where x is 1, 2 or 3, must be specified in increasing order.

*It does not matter in which order the parameters are entered, however, all three parameters (e.g. **matchx**, **maskx** and **offsetx**) must be specified otherwise an error will occur.*

The **protocol** parameter specifies the protocol of the packet, as shown in Table 21-3 on page 21-8. The Classifier provides a predefined list of common protocols, as shown in Table 21-4 on page 21-8. Seven user-specified protocols may be entered for any encapsulation type. The **ip, ipx, nonipipx** and **any** protocols are always available. The value of **nonipipx** represents all protocols except for IP and IPX. If the command specifies **protocol=ip**, the IP packets will have either ETHII or SNAP encapsulation (e.g. **ethformat=ethii** or **ethformat=snap**). If the command specifies **protocol=ipx**, the IPX packets may have any of the four types of encapsulation (e.g. **ethformat={802.2 | ethii | netwareraw | snap}**). If a ten digit hexadecimal number is specified, e.g. **protocol=*xxxxxxabcd***, the last four digits (*abcd*) are used for packet classification. If you specify a TCP or UDP packet matching rule or **ipprotocol=icmp**, the default is **ip**. Otherwise, the default is **any**.

Table 21-3: Protocol values of encapsulated packets

Encapsulation Type	Value
SAP	The value of the DSAP field.
ETHII	The value of the ETYPE field.
NETWARERAW	The value of the IPX checksum field (hexadecimal FFFF).
SNAP	The value of the ETYPE field.

Table 21-4: Predefined protocol types for use in the **protocol** parameter

Protocol Name	Protocol Number	Encapsulation	Min. Chars to Enter
SNA Path Control	04	SAP	3
PROWAY-LAN	0E	SAP	7
EIA-RS	4E	SAP	3
PROWAY	8E	SAP	3

Table 21-4: Predefined protocol types for use in the **protocol** parameter (Continued)

Protocol Name	Protocol Number	Encapsulation	Min. Chars to Enter
IPX 802.2	E0	SAP	9
NetBEUI	F0	SAP	3
ISO CLNS IS	FE	SAP	5
IP ETHII	0800	EthII	8
X.75 Internet	0801	EthII	4
NBS Internet	0802	EthII	3
ECMA Internet	0803	EthII	4
Chaosnet	0804	EthII	4
X.25 Level 3	0805	EthII	4
ARP	0806	EthII	3
XNS Compat	0807	EthII	3
Banyan Systems	0BAD	EthII	3
BBN Simnet	5208	EthII	3
DEC MOP Dump/Ld	6001	EthII	9
DEC MOP Rem Cons	6002	EthII	9
DEC DECNET	6003	EthII	7
DEC LAT	6004	EthII	7
DEC Diagnostic	6005	EthII	7
DEC Customer	6006	EthII	7
DEC LAVC	6007	EthII	7
RARP	8035	EthII	4
DEC LANBridge	8038	EthII	7
DEC Encryption	803D	EthII	7
AppleTalk	809B	EthII	3
IBM SNA	80D5	EthII	7
AppleTalk AARP	80F3	EthII	11
IPX EthII	8137	EthII	9
SNMP	814C	EthII	4
IPv6 ETHII	86DD	EthII	10
ETHERTALK 2	080007809B	SNAP	11
ETHERTALK 2 AARP	00000080F3	SNAP	13
IP SNAP	0000000800	SNAP	7
IPX SNAP	0000008137	SNAP	8
IPX 802.3	FFFF	NetWare 802.3 Raw	9
IP	-	EthII, SNAP	2
IPX	-	NetWare 802.3 Raw, EthII, SNAP, SAP	3
IPV6	-	EthII	4
NONIPIPX	-	EthII, SNAP, SAP	4

Table 21-4: Predefined protocol types for use in the **protocol** parameter (Continued)

Protocol Name	Protocol Number	Encapsulation	Min. Chars to Enter
ANY	-	NetWare 802.3 Raw, EthII, SNAP, SAP	3

Note: When you enter a protocol name that contains spaces, you must surround the name with double quotation marks. You can use lowercase or uppercase letters. For example, to specify ETHERTALK 2 AARP, enter **protocol="ethertalk 2 aarp"** or **protocol="ethertalk 2 a"**.

The **tcpdport** parameter specifies the TCP destination port of a TCP/IP packet. The default is **any**.

The **tcpsport** parameter specifies the TCP source port of a TCP/IP packet. The default is **any**.

The **tcpflags** parameter specifies the TCP flags of a TCP/IP packet. The default is **any**.

The **udpdport** parameter specifies the UDP destination port of an UDP/IP packet.The default is **any**.

The **udpsport** parameter specifies the UDP source port of an UDP/IP packet.The default is **any**.

Examples

To create a packet matching rule that matches all IP packets from the IP subnet 192.168.100.2 (mask=255.255.255.0) with a destination TCP port of 23, use one of the commands:

```
create classifier=1 ipsaddr=192.168.100.2/24 tcpdport=23

create classifier=1 protocol=ip ipsaddr=192.168.100.2/24
   tcpdport=23

create classifier=1 ethformat=ethii vlan=any protocol=ip
   ipaddr=any ipsaddr=192.168.100.2/24 ipdscp=any
   ipprotocol=tcp tcpdport=23 tcpsport=any tcpflags=any
```

To create a packet matching rule that matches all other IP packets with a destination TCP port of 23, use one of the commands:

```
create classifier=2 tcpdport=23

create classifier=2 ipsaddr=any tcpdport=23

create classifier=2 ipdaddr=any ipsaddr=any tcpdport=23

create classifier=2 ethformat=ethii vlan=any protocol=ip
   ipaddr=any ipsaddr=192.168.100.2/24 ipdscp=any
   ipprotocol=tcp tcpdport=23 tcpsport=any tcpflags=any
```

These two classifiers can be used in combination in hardware filters or QoS flow groups to separate Telnet traffic from this subnet from other Telnet traffic.

Related Commands

destroy classifier
set classifier
show classifier

destroy classifier

Syntax `DESTroy CLASSifier={`*`rule-list`*`|ALL}`

where *rule-list* is a single rule-id or a group of rule-ids, either a comma separated list, a range (specified as n-m) or a combination of the two. Rule-ids start at 1.

Description This command destroys one or more packet matching rules.

The **classifier** parameter specifies the rule-id of an existing packet matching rule. This rule cannot be associated with any action in any other software module. If **all** is specified, then all packet matching rules will be destroyed.

Examples To destroy the packet matching rules with rule-ids 3, 5 and 9 to 12, use the command:

```
dest class=3,5,9-12
```

To destroy all packet matching rules, use the command:

```
dest class=all
```

Related Commands **create classifier**
show classifier

set classifier

Syntax

```
SET CLASSIFIER=rule-id [VLAN={vlan-name|1..4094|ANY}]
   [EPOrt=portnum] [IPOrt=portnum] [ETHFormat={802.2|
   802.2-Tagged|802.2-Untagged|ETHII|ETHII-Tagged|ETHII-
   Untagged|Netwareraw|Snap|ANY}] [IPDaddr={ipaddmask|
   ANY}] [IPSaddr={ipaddmask|ANY}] [IPDScp={0..63|ANY]
   [IPPRotocol={tcp|udp|icmp|igmp|ipprotocolnum}]
   [IPTOs={0..7|ANY}] [IPXDAddr={ipxadd|ANY}]
   [IPXDSocket={NCP|SAP|RIP|NNB|DIAg|NLSp|IPXwan|
   ipxsocketnum|ANY}] [IPXSSocket={NCP|SAP|RIP|NNB|DIAg|
   NLSp|IPXwan|ipxsocketnum|ANY}] [IPXPacket={NLSP|RIP|
   SAP|SPX|NCP|NETBIOS|ipxpacketnum|ANY}]
   [MACDaddr={macadd|ANY}] [MACSaddr={macadd|ANY}][1]
   [MATCH1=hhh] [MASK1=hhh] [OFFSET1=0..62] [MATCH2=hhh]
   [MASK2=hhh] [OFFSET2=0..62] [MATCH3=hhh] [MASK3=hhh]
   [OFFSET3=0..62] [PROTocol={protocoltype|ANY}]
   [TCPDport={portid|ANY}] [TCPSport={portid|ANY}]
   [TCPFlags={{Urg|Ack|Rst|Syn|Fin}[,...]|ANY}]
   [UDPDport={portid|ANY}] [UDPSport={portid|ANY}]
```

where:

- *rule-id* is a decimal number from 1 to 9999.
- *vlan-name* is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits (0-9), the underscore character ("_"), and the hyphen (-). The *vlan-name* cannot be a number, **all** or **any**.
- *portnum* is the number of a physical port.
- *dscplist* is a single number or a group of numbers, either a comma separated list, a range (specified as n-m) or a combination of the two. Numbers start at 0 and end at 63.
- *ipaddmask* is an IP address in dotted decimal notation with an optional mask. The optional mask is specified by adding "/M" following the IP address, where M is the number of contiguous bits in the mask; M ranges from 0 to 32 inclusive.
- *ipprotocolnum* is either an IP protocol number or a recognised IP protocol name. An IP protocol number is expressed as a 1 byte decimal number.
- *ipxadd* is an IPX network address expressed as a 4 byte hexadecimal number.
- *ipxsocketnum* is either an IPX socket number or a recognised IPX socket type. An IPX socket number is expressed as a 2 byte hexadecimal number.
- *ipxpacketnum* is either an IPX packet number or a recognised IPX packet type. An IPX packet number is expressed as a 2 byte hexadecimal number.
- *macadd* is an Ethernet six-octet MAC address, expressed as six pairs of hexadecimal digits delimited by hyphens.

1. Due to internal configuration settings, it may not be possible to specify Layer 2 MAC address-based packet matching rules in conjunction with Layer 3 packet matching rules. Given that Layer 3 IP/IPX and other Layer 3 data packet matching rules are more important than MAC address packet matching rules, this parameter may not be available. This applies to both the MACDADDR and MACSADDR parameters.

- *hhh* is a 2 byte hexadecimal number.
- *protocoltype* is either a valid protocol number or a recognised protocol name. A protocol number can be either 1 byte for SAP, 2 bytes for ETHII or 5 bytes for an 802.3/802.2 SNAP type packet, and is specified in hexadecimal.
- *portid* is a TCP/IP or UDP/IP port number.

Description This command creates a packet matching rule that identifies a particular data flow. This data flow may be specific in nature (e.g. IP packets with a particular TCP destination port), or it may be general (e.g. all IPX packets with a SNAP Ethernet packet encapsulation).

If a packet with an unknown destination port is to be transmitted, the packet is flooded to all ports in the VLAN. For such a packet, no hardware filters or QoS will be applied to the packet. This also applies to any broadcast or multicast IP or IPX packet.

The **classifier** parameter specifies a rule-id for the packet matching rule. This identifier is used to uniquely identify the rule. It does not imply an order between this rule and previously created rules.

The **vlan** parameter specifies the destination VLAN that the packet will be transmitted on. The default is **any**.

The **eport** parameter specifies the egress port on the switch to match for each frame. A classifier which matches traffic on the basis of egress port can only be applied to a port in the same port block as the egress port (ports 1 to 24 or 25 to 48 on a 48-port switch).

The **iport** parameter specifies the ingress port on the switch to match for each frame.

The **ethformat** parameter specifies the Ethernet encapsulation type of the packet. The parameter values and encapsulation types are detailed in Table 21-1 on page 21-6.

If an **ethformat** option is specified (excluding **any**), the **protocol** parameter must also be used. Specifying **ethformat=snap protocol=ip** means using **protocol=0800**. Specifying **ethformat=snap protocol=ipx** means using **protocol=8137**. Specifying **ethformat=802.2 protocol=ipx** means using **protocol=e0**. In addition, the values **802.2-tagged, 802.2-untagged, ethii-tagged** and **ethii-untagged** may be used to classify packets that were either tagged or untagged at ingress. Table 21-2 on page 21-6 shows the available **ethformat** and **protocol** parameter combinations and their implementation in the Classifier.

The **ipdaddr** parameter specifies the destination IP address (either host or subnet) of an IP packet. The default is **any**.

The **ipsaddr** parameter specifies the source IP address (either host or subnet) of an IP packet. The default is **any**.

The **ipdscp** parameter specifies the Code Point bits of the DiffServ field of an IP packet (from 0 to 63, or **any**). This parameter cannot be specified in conjunction with the **iptos** parameter. The default is **any**.

The **ipprotocol** parameter specifies a Layer 4 IP protocol of an IP packet. If the command specifies a TCP/IP packet matching rule, (e.g. **tcpdport** is specified), then the default value for this parameter is TCP. If the command specifies a UDP/IP packet matching rule, (e.g. **udpdport** is specified), then the default value for this parameter is **udp**. The value of **nontcpudp** represents all Layer 4 IP protocols except for TCP and UDP. There can be 29 unique **ipprotocol** values specified across all Classifiers, not including TCP and UDP. If **icmp** is specified, it is considered as one of the 29 unique values. **igmp, tcp, udp, nontcpudp** and **any** are all system default values and are always available.

The **iptos** parameter specifies the value of the precedence field within the TOS byte of an IP packet. This parameter cannot be specified in conjunction with the **ipdscp** parameter. The default is **any**.

The **ipxdaddr** parameter specifies the destination network address of an IPX packet. The default is **any**.

The **ipxpacket** parameter specifies the value of the Packet Type field of an IPX packet. The default is **any**.

The **ipxdsocket** specifies the destination IPX socket number of an IPX packet. There can be 7 unique **ipxdsocket** values specified across all Classifiers. Specifying any of **ncp, sap, rip, nnb, diag, nlsp** or **ipxwan** counts as one of the seven user-specified values. The default is **any**.

The **ipxssocket** specifies the source IPX socket number of an IPX packet. There can be 7 user-specified **ipxssocket** values. Specifying any of **ncp, sap, rip, nnb, diag, nlsp** or **ipxwan** counts as one of the seven user-specified values. The default is **any**.

The **macdaddr** parameter specifies the destination MAC address of the packet. The default is **any**.

The **macsaddr** parameter specifies the source MAC address of the packet. The default is **any**.

The **match*x***, **mask*x*** and **offset*x*** parameters, where *x* is 1, 2 or 3, specify a general 16-bit word to match inside a packet. The three parameters must be specified at the same time on the command line. The **match** parameter specifies the actual data to match. The **mask** parameters specify whether the corresponding bit in the **match** parameter is "on" for a match or "don't care" for a match. If the **mask** bit is set (on), the bit in the **match** parameter must be the same as the corresponding bit in the actual packet (i.e. place "ones" in bit positions you want to match). If the **mask** bit is clear (don't care), the same bit in the **match** parameter will not be checked with the corresponding bit in the actual packet. The **offset*x*** parameters, where x is 1, 2 or 3, must be specified in increasing order.

*It does not matter in which order the parameters are entered, however, all three parameters (e.g. **matchx**, **maskx** and **offsetx**) must be specified otherwise an error will occur.*

The **protocol** parameter specifies the protocol of the packet, as shown in Table 21-3 on page 21-8. The Classifier provides a predefined list of common protocols, as shown in Table 21-4 on page 21-8. Seven user-specified protocols may be entered for any encapsulation type. The **ip, ipx, nonipipx** and **any** protocols are always available. The value of **nonipipx** represents all protocols

except for IP and IPX. If the command specifies **protocol=ip**, the IP packets will have either ETHII or SNAP encapsulation (e.g. **ethformat=ethii** or **ethformat=snap**). If the command specifies **protocol=ipx**, the IPX packets may have any of the four types of encapsulation (e.g. **ethformat={802.2 | ethii | netwareraw | snap}**). If a ten digit hexadecimal number is specified, e.g. **protocol=*xxxxxxabcd*** the last four digits (*abcd*) are used for packet classification. If you specify a TCP or UDP packet matching rule or **ipprotocol=icmp**, the default is **ip**. Otherwise, the default is **any**.

The **tcpdport** parameter specifies the TCP destination port of a TCP/IP packet. The default is **any**.

The **tcpsport** parameter specifies the TCP source port of a TCP/IP packet. The default is **any**.

The **tcpflags** parameter specifies the TCP flags of a TCP/IP packet. The default is **any**.

The **udpdport** parameter specifies the UDP destination port of an UDP/IP packet.The default is **any**.

The **udpsport** parameter specifies the UDP source port of an UDP/IP packet.The default is **any**.

Examples

To set packet matching rule 1 so that it matches all IP packets from the IP subnet 192.168.100.2 (mask=255.255.255.0), with a destination TCP port of 23, use one of the commands:

```
set classifier=1 ipsaddr=192.168.100.2/24 tcpdport=23

set classifier=1 protocol=ip ipsaddr=202.36.164.2/24
   tcpdport=23
```

Related Commands

create classifier
show classifier

show classifier

Syntax

```
SHow CLASSIFIER=rule-id [VLAN={vlan-name|1..4094|ANY}]
   [EPOrt=portnum] [IPOrt=portnum] [ETHFormat={802.2|
   802.2-Tagged|802.2-Untagged|ETHII|ETHII-Tagged|ETHII-
   Untagged|Netwareraw|Snap|ANY}] [IPDaddr={ipaddmask|
   ANY}] [IPSaddr={ipaddmask|ANY}] [IPDScp={0..63|ANY]
   [IPPRotocol={tcp|udp|icmp|igmp|ipprotocolnum}]
   [IPTOs={0..7|ANY}] [IPXDAddr={ipxadd|ANY}]
   [IPXDSocket={NCP|SAP|RIP|NNB|DIAg|NLSp|IPXwan|
   ipxsocketnum|ANY}] [IPXSSocket={NCP|SAP|RIP|NNB|DIAg|
   NLSp|IPXwan|ipxsocketnum|ANY}] [IPXPacket={NLSP|RIP|
   SAP|SPX|NCP|NETBIOS|ipxpacketnum|ANY}]
   [MACDaddr={macadd|ANY}] [MACSaddr={macadd|ANY}]
   [MATCH1=hhh] [MASK1=hhh] [OFFSET1=0..62] [MATCH2=hhh]
   [MASK2=hhh] [OFFSET2=0..62] [MATCH3=hhh] [MASK3=hhh]
   [OFFSET3=0..62] [PROTocol={protocoltype|ANY}]
   [TCPDport={portid|ANY}] [TCPSport={portid|ANY}]
   [TCPFlags={{Urg|Ack|Rst|Syn|Fin}[,...]|ANY}]
   [UDPDport={portid|ANY}] [UDPSport={portid|ANY}]
```

where:

- *id-list* is a classifier ID number, a range of classifier ID numbers separated by a hyphen, or a comma-separated list of classifier ID numbers and/or ranges (e.g. 1,3,4-9). Classifier ID numbers range from 1 to 9999.
- *vlan-name* is a unique name from 1 to 32 characters. Valid characters are uppercase and lowercase letters, digits (0-9), the underscore character ("_"), and the hyphen (-). The *vlan-name* cannot be a number, **all** or **any**.
- *portnum* is the number of a physical port.
- *ipaddmask* is an IP address in dotted decimal notation with an optional mask. The optional mask is specified by adding "/M" following the IP address, where M is the number of contiguous bits in the mask; M ranges from 0 to 31 inclusive.
- *dscplist* is a single number or a group of numbers, either a comma separated list, a range (specified as n-m) or a combination of the two. Numbers start at 0 and end at 63.
- *ipprotocolnum* is either an IP protocol number or a recognised IP protocol name. An IP protocol number is expressed as a 1 byte hexadecimal number.
- *ipxadd* is an IPX network address expressed as a 4 byte hexadecimal number.
- *ipxsocketnum* is either an IPX socket number or a recognised IPX socket type. An IPX socket number is expressed as a 2 byte hexadecimal number.
- *ipxpacketnum* is either an IPX packet number or a recognised IPX packet type. An IPX packet number is expressed as a 1 byte hexadecimal number.
- *macadd* is an Ethernet six-octet MAC address, expressed as six pairs of hexadecimal digits delimited by hyphens.
- *hhh* is a 2 byte hexadecimal number.

- *protocoltype* is either a valid protocol number or a recognised protocol name. A protocol number can be either 1 byte for SAP, 2 bytes for ETHII or 5 bytes for an 802.3/802.2 SNAP type packet, and is specified in hexadecimal.
- *portid* is a TCP/IP or UDP/IP port number.

Description

This command displays the output of the **show classifier** command and packet matching rules. For example output and parameter descriptions see:

- Figure 21-2 on page 21-17 and Table 21-5 on page 21-17, an example of output from the **show classifier** command, listing all rules in summary.
- Figure 21-3 on page 21-18 and Table 21-6 on page 21-18, an example of output from the **show classifier=all** command, listing all rules in detail.
- Figure 21-4 on page 21-19 and Table 21-7 on page 21-19, an example of a classifier rule matching TCP/IP flows.
- Figure 21-5 on page 21-19 and Table 21-7 on page 21-19, an example of a classifier rule matching UDP flows.
- Figure 21-6 on page 21-20 and Table 21-8 on page 21-20, an example of a classifier rule matching on MAC address.
- Figure 21-7 on page 21-21 and Table 21-9 on page 21-21, an example of a classifier rule matching IPX traffic.

If the **classifier** parameter specifies no value, then all rules are displayed.

If the **classifier** parameter specifies a value, then a detailed output is displayed. In the detailed output, only parameters that have a non-default value are shown.

Parameters that are not matched (e.g. value=**any**) or that do not apply to a particular rule may not be displayed in the output.

Figure 21-2: Example output from the **show classifier** command

```
 Classifier General Info
-------------------------------------------------------------
 Total number of rules .... 6

 Rule  Type            Related Module(s)
 ---------------------------------------
    1  L2              L3 switch
    2  L2              L3 switch, QOS
  100  L4,L3,L2        QOS
  200  L3,L2           L3 switch, QOS
  700  L3              None
 9999  Match all       None
-------------------------------------------------------------
```

Table 21-5: Parameters in the output of the **show classifier** command

Parameter	Meaning
Rule	The rule identifier for the packet matching rule.
Type	A list of the OSI layers at which specified parameters (those with non-default values) in the rule operate, one or more of: **L5**, **L4**, **L3**, **L2**, **L1**.
Related module(s)	The name of the module(s) that are currently using the rule.

Figure 21-3: Example output from the **show classifier=all** command

```
 Classifier Rules
-------------------------------------------------------------
 Rule ................. 1
  VLAN ................ default (1)

 Rule ................. 2
  VLAN ................ v2 (2)

 Rule ................. 100
  Protocol ............ IP
  D-IP Address ........ 10.0.0.1/32
  IP Protocol ......... TCP
  D-TCP Port .......... 23

 Rule ................. 200
  Protocol ............ IP
  D-IP Address ........ 10.0.0.1/32

 Rule ................. 700
  MATCH1 .............. 1111
  MASK1 ............... 2222
  OFFSET1 ............. 1

 Rule ................. 9999
  Match all frames
-------------------------------------------------------------
```

Table 21-6: Parameters in the output of the **show classifier=all** command

Parameter	Meaning
Rule	The rule identifier for the packet matching rule.
VLAN	The name of a VLAN with only the first 10 characters shown. The VLAN Identifier appears in brackets. If the packet is Layer 3 switched, this VLAN is the destination VLAN. If the packet is Layer 3 routed by the CPU, this VLAN is the source VLAN. If the packet is Layer 2 switched, its source and destination VLAN are the same.
E-Format	The Ethernet encapsulation format for the packet, one of "802.2", "ETHII", "NETWARERAW" or "SNAP". The ethernet encapsulation format may also be suffixed with "-TAGGED" or "-UNTAGGED" if this has been specified in the rule.
Protocol	The hexadecimal value of the protocol. If the protocol is not for the general family of IP and IPX protocols, and the commonly known name for the protocol is known to the Classifier, then this commonly known name will be printed in brackets after the hexadecimal number.
S-IP Address	The source IP address field of a packet.
D-IP Address	The destination IP address field of a packet.
IP Protocol	The Layer 4 IP protocol field of a packet.
S-TCP Port	The source TCP/IP port field of a packet.
D-TCP Port	The destination TCP/IP port field of a packet.
S-UDP Port	The source UDP/IP port field of a packet.
D-UDP Port	The destination UDP/IP port field of a packet.
D-IPX Address	The destination IPX network address field of a packet.

Table 21-6: Parameters in the output of the **show classifier=all** command (Continued)

Parameter	Meaning
D-IPX Socket	The destination IPX socket field of a packet.
S-IPX Socket	The source IPX socket field of a packet.

Figure 21-4: Example output from the **show classifier** command (TCP/IP data flow)

```
 Classifier Rules
-------------------------------------------------------------
 Rule .................. 1
  Ingress Port ......... 1
  Egress Port .......... 24
  D-MAC Address ........ 00-00-cd-00-03-48
  S-MAC Address ........ 00-00-cd-00-01-e4
  VLAN ................. vlan1234 (1234)
  E-Format ............. ETHII
  Protocol ............. 0800 (IP EthII)
  S-IP Address ......... 192.168.123.123/32
  D-IP Address ......... 192.168.123.123/32
  IP Protocol .......... TCP
  S-TCP Port ........... 23
  D-TCP Port ........... 23
-------------------------------------------------------------
```

Figure 21-5: Example output from the **show classifier** command (UDP/IP data flow)

```
 Classifier Rules
-------------------------------------------------------------
 Rule .................. 21
  Ingress Port ......... 1
  Egress Port .......... 24
  D-MAC Address ........ 00-00-cd-00-03-48
  S-MAC Address ........ 00-00-cd-00-01-e4
  VLAN ................. vlan1234 (1234)
  E-Format ............. ETHII
  Protocol ............. 0800 (IP EthII)
  S-IP Address ......... 192.168.123.123/32
  D-IP Address ......... 192.168.123.123/32
  IP Protocol .......... UDP
  S-UDP Port ........... 23
  D-UDP Port ........... 23
-------------------------------------------------------------
```

Table 21-7: Parameters in the output of the **show classifier** command (TCP and UDP/IP data flows)

Parameter	Meaning
Rule	The rule identifier for the packet matching rule/classifier.
Ingress Port	The number of the ingress port associated with the rule.
Egress Port	The number of the egress port associated with the rule.
D-MAC Address	The destination MAC address field of a packet.
S-MAC Address	The source MAC address field of a packet.

Table 21-7: Parameters in the output of the **show classifier** command (TCP and UDP/IP data flows) (Continued)

Parameter	Meaning
VLAN	The name of a VLAN with only the first 10 characters shown. The VLAN Identifier appears in brackets. If the packet is Layer 3 switched, this VLAN is the destination VLAN. If the packet is Layer 3 routed by the CPU, this VLAN is the source VLAN. If the packet is Layer 2 switched, its source and destination VLAN are the same.
E-Format	The Ethernet encapsulation format for the packet, one of "802.2", "ETHII", "NETWARERAW" or "SNAP". The Ethernet format for the packet it may also be suffixed with "-TAGGED" or "-UNTAGGED" if this has been specified in the rule.
Protocol	The hexadecimal value of the protocol. If the protocol is not for the general family of IP and IPX protocols, and the commonly known name for the protocol is known to the Classifier, then this commonly known name will be printed in brackets after the hexadecimal number.
S-IP Address	The source IP address field of a packet.
D-IP Address	The destination IP address field of a packet.
IP Protocol	The Layer 4 IP protocol field of a packet.
TOS/DSCP	The IP TOS or DiffServ Code Point field of a packet.
S-TCP Port	The source TCP/IP port field of a packet.
D-TCP Port	The destination TCP/IP port field of a packet.
S-UDP Port	The source UDP/IP port field of a packet.
D-UDP Port	The destination UDP/IP port field of a packet.
TCP Flags	A series of letters representing the TCP/IP flag field, one of URG, ACK, RST, SYN or FIN.

Figure 21-6: Example output from the **show classifier** command (MAC address)

```
 Classifier Rules
-----------------------------------------------------------
 Rule .................. 2222
  D-MAC Address ........ aa-bb-cc-dd-ee-ff
  S-MAC Address ........ aa-bb-cc-dd-ee-ff
  VLAN ................. vlan1234 (1234)
  E-Format ............. SNAP
  Protocol ............. 1234567890 (-)
-----------------------------------------------------------
```

Table 21-8: Parameters in the output of the **show classifier** command (MAC address)

Parameter	Meaning
Rule	The rule identifier for the packet matching rule/classifier.
D-MAC Address	The destination MAC address field of a packet.
S-MAC Address	The source MAC address field of a packet.

Table 21-8: Parameters in the output of the **show classifier** command (MAC address) (Continued)

Parameter	Meaning
VLAN	The name of a VLAN with only the first 10 characters shown. The VLAN Identifier appears in brackets. If the packet is Layer 3 switched, this VLAN is the destination VLAN. If the packet is Layer 3 routed by the CPU, this VLAN is the source VLAN. If the packet is Layer 2 switched, its source and destination VLAN are the same.
E-Format	The Ethernet encapsulation format for the packet, one of "802.2", "ETHII", "NETWARERAW" or "SNAP". The Ethernet encapsulation format may also be suffixed with "-TAGGED" or "-UNTAGGED" if this has been specified in the rule.
Protocol	The hexadecimal value of the protocol. If the protocol is not for the general family of IP and IPX protocols, and the commonly known name for the protocol is known to the Classifier, then this commonly known name will be printed in brackets after the hexadecimal number.

Figure 21-7: Example output from the **show classifier** command (IPX data flow)

```
 Classifier Rules
-----------------------------------------------------------
 Rule ................. 31
  Protocol ............. IPX
  D-IPX Socket ......... RIP
-----------------------------------------------------------
```

Table 21-9: Parameters in the output of the **show classifier** command (IPX data flow)

Parameter	Meaning
Rule	The rule identifier for the packet matching rule.
D-MAC Address	The destination MAC address field of a packet.
S-MAC Address	The source MAC address field of a packet.
VLAN	The name of a VLAN with only the first 10 characters shown. The VLAN Identifier appears in brackets. If the packet is Layer 3 switched, this VLAN is the destination VLAN. If the packet is Layer 3 routed by the CPU, this VLAN is the source VLAN. If the packet is Layer 2 switched, its source and destination VLAN are the same.
E-Format	The Ethernet encapsulation format for the packet, one of "802.2", "ETHII", "NETWARERAW" or "SNAP". The Ethernet encapsulation format may also be suffixed with "-TAGGED" or "-UNTAGGED" if this has been specified in the rule.
Protocol	The hexadecimal value of the protocol. If the protocol is not for the general family of IP and IPX protocols, and the commonly known name for the protocol is known to the Classifier, then this commonly known name will be printed in brackets after the hexadecimal number.
D-IPX Address	The destination IPX network address field of a packet.
D-IPX Socket	The destination IPX socket field of a packet.
S-IPX Socket	The source IPX socket field of a packet.

Examples To display the number of each of the classifiers and which module is using each classifier, use the command:

```
sh class
```

To display what each classifier matches against, use the command:

```
sh class=all
```

Related Commands

create classifier
destroy classifier
set classifier