

## X.25 Configuration Commands

---

Use the commands in this chapter to configure Link Access Procedure Balanced (LAPB), X.25, DDN X.25, and Blacker Front-end Encryption (BFE).

LAPB enables you to configure a simple, reliable serial encapsulation method. For historical background and a technical overview of X.25 and LAPB, see the *Internetworking Technology Overview* publication. For X.25 and LAPB configuration information and examples, refer to the “Configuring LAPB and X.25” chapter in the *Access and Communication Servers Configuration Guide*.

## access-class

To configure an incoming access class on virtual terminals, use the **access-class** line configuration command.

**access-class** *access-list-number* **in**

### Syntax Description

*access-list-number*     An integer between 1 and 199 that you select for the access list

**in**                         Restricts incoming connections between a particular Cisco device and the addresses in the access list.

### Default

No incoming access class is defined.

### Command Mode

Line configuration

### Usage Guidelines

The access list number is used for both incoming TCP access and incoming PAD access.

In the case of TCP access, the communication server uses the IP access list defined using the **access-list** command.

For incoming PAD connections, the same numbered X.29 access list is referenced. If you only want to have access restrictions on one of the protocols, you can create an access list that permits all addresses for the other protocol.

### Example

The following example configures an incoming access class on virtual terminal line 4.

```
line vty 4
access-class 4 in
```

### Related Commands

The dagger (†) indicates that the command is documented in another chapter of this manual.

**access-list**  
**line vty** †  
**x29 access-list**

## bfe

Use the **bfe EXEC** command to set the communication server to participate in emergency mode or to end participation in emergency mode when your system is configured for **x25 bfe-emergency decision** and **x25 bfe-decision ask**.

```
bfe {enter | leave} interface type number
```

### Syntax Description

<b>enter</b>	Causes the communication server to send a special address translation packet that includes an enter emergency mode command to the BFE if the emergency mode window is open. If the BFE is already in emergency mode, this command enables the sending of address translation information.
<b>leave</b>	Disables the sending of address translation information from the communication server to the BFE when the BFE is in emergency mode.
<b>interface</b>	Permits you to specify an interface type and number.
<i>type</i>	Interface type.
<i>number</i>	Interface number.

### Command Mode

EXEC

### Example

The following example sets the communication server to participate in BFE mode:

```
bfe enter interface serial 0
```

### Related Commands

**encapsulation bfex25**

**x25 bfe-decision**

**x25 bfe-emergency**

## clear x25-vc

Use the **clear x25-vc** privileged EXEC command to clear virtual circuits.

**clear x25-vc** *type number* [*lcn*]

### Syntax Description

<i>type</i>	Interface type
<i>number</i>	Interface number
<i>lcn</i>	(Optional) Virtual circuit

### Command Mode

Privileged EXEC

### Usage Guidelines

This command without any arguments clears all X.25 virtual circuits at once.

### Example

The following example clears all virtual circuits:

```
clear x25-vc
```

### Related Command

**x25 idle**

## encapsulation lapb

To exchange datagrams over a serial interface using LAPB encapsulation, use the **encapsulation lapb** interface configuration command.

```
encapsulation lapb [dte | dce] [multi | protocol]
```

### Syntax Description

<b>dte</b>	(Optional) Specifies operation as a DTE. This is the default LAPB mode.
<b>dce</b>	(Optional) Specifies operation as a DCE.
<b>multi</b>	(Optional) Specifies the use of multiple local-area network (LAN) protocols to be carried on the LAPB line.
<i>protocol</i>	(Optional) A single protocol to be carried on the LAPB Line. It can be either <b>ip</b> or <b>ipx</b> (Novell IPX). IP is the default protocol.

### Default

The default serial encapsulation is HDLC. You must explicitly configure a LAPB encapsulation method.

DTE operation is the default LAPB mode. IP is the default protocol.

### Command Mode

Interface configuration

### Usage Guidelines

LAPB encapsulations are appropriate only for private connections, where you have complete control over both ends of the link. Connections to X.25 networks should use an **x25 encapsulation** configuration, which operates the X.25 Layer 3 protocol above a LAPB Layer 2.

One end of the link must be a logical DCE and the other end a logical DTE. (This assignment is independent of the interface's hardware DTE/DCE identity.)

Both ends of the LAPB link must specify the same protocol encapsulation.

A single-protocol LAPB encapsulation exchanges datagrams of the given protocol, each in a separate LAPB information frame. You should configure the interface with the protocol-specific parameters needed (for example, a link that carries IP traffic will have an IP address defined for the interface).

A multiprotocol LAPB encapsulation can exchange any or all of the protocols allowed for a single-protocol interface. It also exchanges datagrams, each in a separate LAPB information frame, although two bytes of protocol identification data precede the protocol data. You should configure the interface with all of the protocol-specific parameters needed for each protocol carried.

### Example

The following example sets the operating mode as DTE and specifies that IPX protocol traffic will be carried on the LAPB line:

```
interface serial 3
encapsulation lapb dte ipx
```

## encapsulation x25

To specify an interface's operation as an X.25 device, use the **encapsulation x25** interface configuration command.

```
encapsulation x25 [dte | dce] [[ddn | bfe] | [ietf]]
```

### Syntax Description

<b>dte</b>	(Optional) Specifies operation as a DTE. This is the default X.25 mode.
<b>dce</b>	(Optional) Specifies operation as a DCE.
<b>ddn</b>	(Optional) Specifies DDN encapsulation on a communication server using DDN X.25 standard Service.
<b>bfe</b>	(Optional) Specifies BFE encapsulation on a communication server attached to a BFE device.
<b>ietf</b>	(Optional) Specifies that the interface's datagram encapsulation should default to use of the IETF standard method, as defined by RFC 1356.

### Default

The default serial encapsulation is HDLC. You must explicitly configure an X.25 encapsulation method.

DTE operation is the default X.25 mode. Cisco's traditional X.25 encapsulation method is the default.

### Command Mode

Interface configuration

### Usage Guidelines

One end of an X.25 link must be a logical DCE and the other end a logical DTE. (This assignment is independent of the interface's hardware DTE/DCE identity.) Typically, when connecting to a public data network (PDN), the customer equipment acts as the DTE and the PDN attachment acts as the DCE.

Cisco has supported the encapsulation of a number of datagram protocols for quite some time, using a standard means when available and proprietary means when necessary. More recently the IETF adopted a standard, RFC 1356, for encapsulating most types of datagram traffic over X.25. X.25 interfaces use Cisco's traditional method unless explicitly configured for IETF operation; if the **ietf** keyword is configured, that standard will be used unless Cisco's traditional method is explicitly configured. For details see the **x25 map** command.

When an X.25 interface is reconfigured, all of the interface's X.25 parameters are initialized except the **x25 map** commands. The **x25 map** statements that are configured for an interface are not deleted when the encapsulation is changed, so they will be retained if the interface is later reconfigured for X.25 operation.

A communication server attaching to the Defense Data Network (DDN) or to a Blacker Front End (BFE) device can be configured to use their respective algorithms to convert between IP and X.121 addresses by using the **ddn** or **bfe** options, respectively. An IP address should be assigned to the interface, from which the algorithm will generate the interface's X.121 address; for proper operation, this X.121 address should not be modified.

A communication server DDN attachment can operate as either a DTE or a DCE device. A BFE attachment can operate only as a DTE device. The **ietf** option is not available if either the **ddn** or **bfe** option is selected.

### Example

The following example configures the interface for connection to a Blacker Front End device.

```
interface serial 0
 encapsulation x25 bfe
```



## lapb hold-queue

Use the **lapb hold-queue** interface configuration command to define the number of packets to be held until they can be sent over the LAPB connection. Use the **no lapb hold-queue** command without an argument to remove this command from the configuration file and return to the default value.

```
lapb hold-queue queue-size  
no lapb hold-queue [queue-size]
```

### Syntax Description

<i>queue-size</i>	Defines the number of packets. A hold queue limit of 0 allows an unlimited number of packets in the hold queue. This argument is optional with the <b>no</b> form of the command.
-------------------	---

### Default

12 packets

### Command Mode

Interface configuration

### Example

The following example sets the X.25 hold queue to hold 25 packets:

```
interface serial 0  
lapb hold-queue 25
```

## lapb interface-outage

To specify a period during which a link will remain connected, even if a brief hardware outage occurs, use the **lapb interface-outage** interface configuration command.

**lapb interface-outage** *milliseconds*

### Syntax Description

*milliseconds*

Number of milliseconds a hardware outage can last without having the protocol disconnect the service. The default is 0 milliseconds.

### Defaults

0 milliseconds, which disables this feature.

### Command Mode

Interface configuration

### Usage Guidelines

If a hardware outage lasts longer than the LAPB hardware outage period you select, normal protocol operations will occur. The link will be declared to be down and, when it is restored, a link set up will be initiated.

### Example

The following example sets the interface outage period to 100 milliseconds. The link will remain connected for outages equal to or shorter than that period.

```
encapsulation lapb dte ip  
lapb interface-outage 100
```

## lapb k

To specify the maximum permissible number of outstanding frames, called the window size, use the **lapb k** interface configuration command.

**lapb k** *window-size*

### Syntax Description

*window-size* Frame count. It can be a value from 1 to the modulo size minus 1. The default is 7 frames.

### Default

7 frames

### Command Mode

Interface configuration

### Usage Guidelines

If the window size is changed while the protocol is up, the new value will take effect only when the protocol is reset. The operator will be informed that the new value will not take effect immediately.

When using the LAPB modulo 128 mode (extended mode), the window parameter k should be increased to make use of the ability to send a larger number of frames before acknowledgment is required. This is the basis for its ability to achieve greater throughput on high-speed links that have a low error rate.

This configured value should match the value configured in the peer X.25 switch. Nonmatching values will cause repeated LAPB REJ frames.

### Example

The following example sets the LAPB window size (the k parameter) to ten frames:

```
interface serial 0
 lapb modulo
 lapb k 10
```

## lapb modulo

To specify the LAPB basic (modulo 8) or extended (modulo 128) protocol mode, use the **lapb modulo** interface configuration command.

**lapb modulo** *modulus*

### Syntax Description

<i>modulus</i>	Either 8 or 128. The value 8 specifies LAPB's basic mode; the value 128 specifies LAPB's extended mode. The default is 8.
----------------	---

### Default

Modulo 8

### Command Mode

Interface configuration

### Usage Guidelines

The modulo parameter determines which of LAPB's two modes is to be used. The modulo values derive from the fact that basic mode numbers information frames between 0 and 7, whereas extended mode numbers them between 0 and 127. Basic mode is widely available and is sufficient for most links. Extended mode is an optional LAPB feature that may achieve greater throughput on high-speed links that have a low error rate.

The LAPB operating mode may be set on X.25 links as well as LAPB Links. The X.25 layer modulo is independent of the LAPB layer modulo. Both ends of a link must use the same LAPB mode.

When using modulo 128 mode, the window parameter *k* should be increased to make use of the ability to send a larger number of frames before acknowledgment is required. This is the basis for its ability to achieve greater throughput on high-speed links that have a low error rate.

If the modulo value is changed while the protocol is up, the new value will take effect only when the protocol is reset. The operator will be informed that the new value will not take effect immediately.

### Example

The following example configures a high-speed X.25 link to use LAPB's extended mode:

```
interface serial 1
encapsulation x25
lapb modulo 128
lapb k 40
clock rate 2000000
```

### Related Command

**lapb k**



Configuring N1 to be less than 2104 will generate a warning message that X.25 might have problems because some nondata packets can use up to 259 bytes.

The N1 parameter cannot be set to a value larger than the default without first increasing the hardware maximum transmission unit (MTU) size.

The X.25 software will accept default packet sizes and CALLs that specify maximum packet sizes greater than what the LAPB layer will support, but will negotiate the CALLs placed on the interface to the largest value that can be supported. For switched CALLs, the packet size negotiation takes place end-to-end through the Cisco communication server so the CALL will not have a maximum packet size that exceeds the capability of either of the two interfaces involved.

### Example

The following example sets the N1 bits to 9600:

```
interface serial 0
lapb n1 9600
```



## lapb protocol

Use the **lapb protocol** interface configuration command to configure a protocol on the LAPB line.

**lapb protocol** *protocol*

### Syntax Description

*protocol*

Protocol choice: **ip** and **ipx** (Novell IPX)

### Default

IP

### Command Mode

Interface configuration

### Example

The following example sets IP as the protocol on the LAPB line:

```
interface serial 1
lapb protocol ip
```

### Related Commands

**encapsulation lapb**

**encapsulation lapb-dce**



## lapb t1

Use the **lapb t1** interface configuration command to set the limit retransmission timer period (the LAPB T1 parameter).

**lapb t1** *milliseconds*

### Syntax Description

*milliseconds*                      Number of milliseconds from 1 through 64000

### Default

3000 milliseconds

### Command Mode

Interface configuration

### Usage Guidelines

The retransmission timer determines how long a transmitted frame can remain unacknowledged before the LAPB software polls for an acknowledgment.

To determine an optimal value for the retransmission timer, use the privileged EXEC command **ping** to measure the round-trip time of a maximum-sized frame on the link. Multiply this time by a safety factor that takes into account the speed of the link, the link quality, and the distance. A typical safety factor is 1.5. Choosing a larger safety factor can result in slower data transfer if the line is noisy. However, this disadvantage is minor compared to the excessive retransmissions and effective bandwidth reduction caused by a timer setting that is too small.

### Example

The following example sets the T1 retransmission timer to 20000 milliseconds:

```
interface serial 0
lapb t1 20000
```

## lapb t4

To set the T4 idle timer, after which the system sends out a Poll packet to determine whether the link has suffered an un signaled failure, use the **lapb t4** interface configuration command.

**lapb t4** *seconds*

### Syntax Description

<i>seconds</i>	Number of seconds between reception of the last frame and the transmission of the outgoing Poll; the default value is 0 seconds.
----------------	--

### Defaults

0 seconds, which disables the T4 timer feature.

### Command Mode

Interface configuration

### Usage Guidelines

Any non-zero T4 duration must be greater than T1, the LAPB retransmission timer period.

### Example

The following example will poll the other end of an active link if it has been 10 seconds since the last frame was received; if the far host has failed, the service will be declared down after N2 tries are timed out.

```
interface serial0
encapsulation x25
lapb t4 10
```

### Related Command

**lapb t1**

## show interfaces serial

Use the **show interfaces serial** EXEC command to display information about a serial interface.

**show interfaces serial** *number*

### Syntax Description

*number*                      Interface port number

### Command Mode

EXEC

### Sample Displays

The following is sample output from the **show interfaces serial** command for a serial interface using LAPB encapsulation:

```
cs# show interfaces serial 1

LAPB state is SABMSENT, T1 3000, N1 12056, N2 20, K7,
VS 0, VR 0, RCNT 0, Remote VR 0, Retransmissions 2
IFRAMEs 0/0 RNRs 0/0 REJs 0/0 SABMs 3/0 FRMRs 0/0 DISCs 0/0
LAPB state is DISCONNECT, T1 3000, N1 12000, N2 20, K7, TH 3000
Window is closed
IFRAMEs 12/28 RNRs 0/1 REJs 13/1 SABMs 1/13 FRMRs 3/0 DISCs 0/11
```

Table 8-2 shows the fields relevant to all LAPB connections.

**Table 8-2 Show Interfaces Serial Field Descriptions When LAPB is Enabled**

Parameter	Description
LAPB state is	State of the LAPB protocol.
T1 3000, N1 12056, ...	Current parameter settings.
VS	Modulo 8 frame number to give to the next outgoing I-frame.
VR	Modulo 8 frame number to give to the next I frame expected to be received.
RCNT	Number of received I-frames that have not yet been acknowledged.
Remote VR	Number of the next I-frame the remote expects to receive.
Retransmissions	Count of I-frames that have been retransmitted.
Window is closed	No more frames can be transmitted until some outstanding frames have been acknowledged.
IFRAMEs	Count of Information frames in the form of sent/received.
RNRs	Count of Receiver Not Ready frames in the form of sent/received.
REJs	Count of Reject frames in the form of sent/received.

Parameter	Description
SABMs	Count of Set Asynchronous Balanced Mode commands in the form of sent/received.
FRMRs	Count of Frame Reject frames in the form of sent/received.
DISCs	Count of Disconnect commands in the form of sent/received.

The following is sample output from the **show interfaces serial** command for a serial X.25 interface:

```
cs# show interfaces serial 1

X25 address 000000010100, state R1, modulo 8, idle 0, timer 0, nvc 1
Window size: input 2, output 2, Packet size: input 128, output 128
Timers: T20 180, T21 200, T22 180, T23 180, TH 0
(configuration on RESTART: modulo 8,
Window size: input 2 output 2, Packet size: input 128, output 128
Channels: Incoming-only none, Two-way 5-1024, Outgoing-only none)
RESTARTs 3/2 CALLs 1000+2/1294+190/0+0/ DIAGs 0/0
```

The stability of the X.25 protocol requires that some parameters not be changed without a RESTART of the protocol. Any change to these parameters will be held until a RESTART is sent or received. If any of these parameters will change, the configuration on RESTART information will be output as well as the values that are currently in effect.

Table 8-3 describes significant fields shown in the display.

**Table 8-3 Show Interfaces Serial Field Descriptions for an X.25 Interface**

Field	Description
X25 address 000000010100	Calling address used in the Call Request packet.
state R1	State of the interface. Possible values include: <ul style="list-style-type: none"> <li>• R1 is the normal ready state</li> <li>• R2 is the DCE not-ready state</li> <li>• R3 is the DTE not-ready state</li> </ul> If the state is R2 or R3, the device is awaiting acknowledgment for a Restart packet.
modulo 8	Modulo value; determines the packet sequence numbering scheme used.
idle 0	Number of minutes the communication server waits before closing idle virtual circuits.
timer 0	Value of the interface timer, which is zero unless the interface state is R2 or R3.
nvc 1	Maximum number of simultaneous virtual circuits permitted to and from a single host for a particular protocol.
Window size: input 2, output 2	Default window size (in packets) for the interface. The <b>x25 facility</b> interface configuration command can be used to override these default values for the switched virtual circuits originated by the communication server.
Packet size: input 128, output 128	Default packet size (in bytes) for the interface. The <b>x25 facility</b> interface configuration command can be used to override these default values for the switched virtual circuits originated by the communication server.

---

Field	Description
Timers: T20 180, T21 200, T22 180, T23 180, TH 0	Values of the Request packet timers: <ul style="list-style-type: none"><li>• T10 through T13 for a DCE device</li><li>• T20 through T23 for a DTE device</li></ul>
Channels: Incoming-only none Two-way 5-1024 Outgoing-only none	Displays the channel sequence range for this interface.
RESTARTs 3/2	Shows RESTART packet statistics for the interface using the format Sent/Received.
CALLs 1000+2/1294+190/0+0	Shows CALL packet statistics for the interface using these formats: <ul style="list-style-type: none"><li>• Successful+Failed /</li><li>• Sent+Failed sent /</li><li>• Received+Failed received</li></ul>
DIAGs 0/0	Shows DIAG packet statistics for the interface using the format Forwarded+Failed forwarded.

---

## show x25 map

To display information about configured address maps, use the **show x25 map** EXEC command.

```
show x25 map
```

### Syntax Description

This command has no arguments or keywords.

### Command Mode

EXEC

### Usage Guidelines

The **show x25 map** command shows information about the following:

- Configured maps (defined by the **x25 map** command)
- Maps implicitly defined by encapsulation PVCs (defined by the **x25 pvc** command)
- Dynamic maps (from the X.25 DDN or BFE operations)
- Temporary maps (from unconfigured CMNS endpoints)

### Sample Display

The following is sample output from the **show x25 map** command:

```
cs# show x25 map

Serial0: X.121 1311001 <--> ip 131.108.170.1
      PERMANENT, BROADCAST, 2 VCS: 3 4*
Serial0: X.121 1311005 <--> ipx 128.1
      PERMANENT
```

The display shows that two maps have been configured for the Serial0 interface.

Table 8-4 describes significant fields shown in the output displayed.

**Table 8-4 Show X25 Map Field Description**

Field	Description
Serial0	Interface on which this map is configured.
X.121 1311001	X.121 address of the mapped encapsulation host.
IP 131.108.170.1	Type and address of the higher-level protocol(s) mapped to the remote host. Bridge maps do not have a higher-level address; all bridge datagrams are sent to the mapped X.121 address. CLNS maps refer to a configured neighbor as identified by the X.121 address.

---

Field	Description
PERMANENT	<p>Address-mapping type that has been configured for the interface in this entry. Possible values include:</p> <ul style="list-style-type: none"><li>• <b>CONSTRUCTED</b>—Derived using the DDN or BFE address conversion scheme.</li><li>• <b>PERMANENT</b>—Map was entered using the <b>x25 map</b> interface configuration command.</li><li>• <b>PVC</b>—Map was configured using the <b>x25 pvc</b> interface command.</li><li>• <b>TEMPORARY</b>—A temporary map was created for an incoming unconfigured CMNS connection.</li></ul>
BROADCAST	<p>If any options are configured for an address mapping, they will be listed; the example shows a maps that is configured to forward datagram broadcasts to the mapped host.</p>
2 VCS:	<p>If the map has any active virtual circuits, they are identified.</p>
3 4*	<p>Identifies the circuit number of the active virtual circuits. The asterisk marks the virtual circuit last used to send data.</p> <p>Note that a single protocol virtual circuit might be associated with a multiprotocol map.</p>

---

## show x25 remote-red

Use the **show x25 remote-red** EXEC command to display the one-to-one mapping of the host IP addresses and the remote BFE device's IP addresses.

**show x25 remote-red**

### Syntax Description

This command has no arguments or keywords.

### Command Mode

EXEC

### Sample Display

The following is sample output from the **show x25 remote-red** command:

```
cs# show x25 remote-red
Entry      REMOTE-RED      REMOTE-BLACK    INTERFACE
1          21.0.0.3        21.0.0.7        serial3
2          21.0.0.10       21.0.0.6        serial1
3          21.0.0.24       21.0.0.8        serial3
```

Table 8-5 describes significant fields shown in the display.

**Table 8-5 Show X25 Remote-Red Display Field Description**

Field	Description
Entry	Address mapping entry.
REMOTE-RED	Host IP address.
REMOTE-BLACK	IP address of the remote BFE device.
INTERFACE	Name of interface through which communication with the remote BFE device will take place.



## show x25 vc

To display information about active switched virtual circuits (SVCs) and permanent virtual circuits (PVCs), use the **show x25 vc EXEC** command.

```
show x25 vc [lcn]
```

### Syntax Description

*lcn* (Optional) Logical channel number (LCN)

### Command Mode

EXEC

### Usage Guidelines

To examine a particular virtual circuit, add an LCN argument to the **show x25 vc** command.

This command displays information about virtual circuits that are used for any of the following:

- Encapsulation traffic
- Locally switched traffic
- Remotely switched traffic
- CMNS switched traffic

The connectivity information displayed will vary according to the traffic carried by the virtual circuit. For multiprotocol circuits, the output varies depending on the number and identity of the protocols mapped to the X.121 address and the encapsulation method selected for the circuit.

### Sample Displays

This section provides three sample displays, which correspond to the four types of traffic information this command displays, and tables that describe the fields in each display.

The following sample display shows a virtual circuit that is being used to encapsulate traffic between the communication server and a remote host:

```
cs# show x25 vc 1024

SVC 1024, State: D1, Interface: Serial0
Started 0:00:31, last input 0:00:31, output 0:00:31
Connects 170090 <-->
compressedtcp 131.108.170.90
ip 131.108.170.90
multiprotocol cud pid, standard Tx data PID, Reverse charged
Window size input: 2, output: 2
Packet size input: 128, output: 128
PS: 5 PR: 5 ACK: 4 Remote PR: 5 RCNT: 1 RNR: FALSE
Window is closed
Retransmits: 0 Timer (secs): 0 Reassembly (bytes): 0
Held Fragments/Packets: 0/0
Bytes 505/505 Packets 5/5 Resets 0/0 RNRs 0/0 REJs 0/0 INTs 0/0
```

Table 8-6 describes the general fields shown in the output; Table 8-7 describes the fields specific to encapsulation virtual circuits shown in the output.

**Table 8-6 Show X25 VC Field Descriptions**

Field	Description
SVC 1024	Identifies the type (switched or permanent) and the number of the virtual circuit.
State	State of the virtual circuit (which is independent of the states of other virtual circuits); D1 is the normal ready state. (See the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) <sup>1</sup> X.25 Recommendation for a description of virtual circuit states.)
Interface	Interface or subinterface on which the virtual circuit is established.
Started	Time elapsed since the virtual circuit was created.
last input	Time of last input.
output	Shows time of last output.
Connects...<-->...	Describes the traffic-specific connection information. See Table 8-7, Table 8-8, and Table 8-9 for more information.
Window size	Window sizes for the virtual circuit.
Packet size	Maximum packet sizes for the virtual circuit.
PS	Current send sequence number.
PR	Current receive sequence number.
ACK	Last acknowledged incoming packet.
Remote PR	Last PR number received from the other end of the circuit.
RCNT	Count of unacknowledged input packets.
RNR	State of the Receiver Not Ready flag; this field is true if the network sends a receiver-not-ready packet.
Window is closed	This line appears if the communication server cannot transmit any more packets until the X.25 layer 3 peer has acknowledged some outstanding packets.
Retransmits	Number of times a supervisory packet (RESET or CLEAR) has been retransmitted.
Timer	A nonzero time value indicates that a supervisory packet has not been acknowledged yet or that the virtual circuit is being timed for inactivity.
Reassembly	Number of bytes received and held for reassembly (packets with the More bit set are reassembled into datagrams for encapsulation virtual circuits; switched X.25 traffic does not reassemble).
Held Fragments/Packets	Number of X.25 data fragments to transmit to complete an outgoing datagram, and the number of buffers held for transmission.
Bytes	Total number of bytes sent and received. The Packets, Resets, RNRs, REJs, and INTs fields show the total sent and received packet counts of the indicated types. (RNR is Receiver Not Ready, REJ is Reject, and INT is Interrupt).

1. The ITU-T carries out the functions of the former Consultative Committee for International Telegraph and Telephone (CCITT).

Table 8-7 describes the connection description fields for virtual circuits carrying encapsulation traffic.

Table 8-7 Show X25 VC Encapsulation Traffic Field Descriptions

Field	Description
170090	The X.121 address of the remote host.
ip 131.108.170.90	The higher-level protocol and address values that are mapped to the virtual circuit.
multiprotocol cud pid	Identifies the method used for the protocol identification (PID) in the Call User Data (CUD) field. Because PVCs are not set up using a Call packet, this field is not displayed for encapsulation PVCs. The available methods are as follows: <ul style="list-style-type: none"> <li>cisco—Cisco's traditional method was used to set up a single protocol virtual circuit.</li> <li>ietf—The IETF's standard RFC 1356 method was used to set up a single protocol virtual circuit.</li> <li>snap—The IETF's SNAP method for IP encapsulation was used.</li> <li>multiprotocol—the IETF's multiprotocol encapsulation method was used.</li> </ul>
standard Tx data pid	Identifies the method used for protocol identification (PID) when sending datagrams. The available methods are as follows: <ul style="list-style-type: none"> <li>no—The virtual circuit is a single-protocol virtual circuit; no PID is used.</li> <li>standard—The IETF's standard RFC 1356 method for identifying the protocol is used.</li> <li>snap—The IETF's SNAP method for identifying IP datagrams is used.</li> </ul>
Reverse charged	Some important virtual circuit information might be displayed as needed; a DDN IP precedence encapsulation value, reverse charged virtual circuits, and virtual circuits that allow the D-bit procedure.

The following sample display shows virtual circuits carrying locally switched X.25 traffic.

```

cs# show x25 vc
PVC 1, State: D1, Interface: Serial2
Started 0:01:26, last input never, output never
PVC <--> Serial1 PVC 1 connected
Window size input: 2, output: 2
Packet size input: 128, output: 128
PS: 0 PR: 0 ACK: 0 Remote PR: 0 RCNT: 0 RNR: FALSE
Retransmits: 0 Timer (secs): 0 Reassembly (bytes): 0
Held Fragments/Packets: 0/0
Bytes 0/0 Packets 0/0 Resets 0/0 RNRs 0/0 REJs 0/0 INTs 0/0

SVC 5, State: D1, Interface: Serial2
Started 0:00:16, last input 0:00:15, output 0:00:15
Connects 170093 <--> 170090 from Serial1 VC 5
Window size input: 2, output: 2
Packet size input: 128, output: 128
PS: 5 PR: 5 ACK: 4 Remote PR: 5 RCNT: 1 RNR: FALSE
Retransmits: 0 Timer (secs): 0 Reassembly (bytes): 0
Held Fragments/Packets: 0/0
Bytes 505/505 Packets 5/5 Resets 0/0 RNRs 0/0 REJs 0/0 INTs 0/0

```

Table 8-8 lists the connection description fields for virtual circuits carrying locally switched X.25 traffic.

**Table 8-8 Show X25 VC Local Traffic Field Descriptions**

Field	Description
PVC	Flags PVC information.
Serial1 PVC1	Identifies the other half of a local PVC connection.
connected	Identifies the state of the PVC. If the PVC is not connected, the status of the PVC will also be displayed. See Table 8-10 for PVC status messages.
170093	Identifies the Calling (source) Address of the connection. If a Calling Address Extension was encoded in the call facilities, it will also be displayed. If the source host is a CMNS host, its MAC address will also be displayed.
170090	Identifies the Called (destination) Address of the connection. If a Called Address Extension was encoded in the call facilities, it will also be displayed. If the destination host is a CMNS host, its MAC address will also be displayed.
from Serial1 VC 5	Indicates the direction of the call (“from” or “to”) and the connecting interface and virtual circuit number.

The following sample display shows virtual circuits carrying remotely switched X.25 traffic.

```

cs# show x25 vc
PVC 2, State: D1, Interface: Serial2
Started 0:01:25, last input never, output never
PVC <--> [131.108.165.92] Serial2/0 PVC 1 connected
XOT between 131.108.165.91, 1998 and 131.108.165.92, 27801
Window size input: 2, output: 2
Packet size input: 128, output: 128
PS: 0 PR: 0 ACK: 0 Remote PR: 0 RCNT: 0 RNR: FALSE
Retransmits: 0 Timer (secs): 0 Reassembly (bytes): 0
Held Fragments/Packets: 0/0
Bytes 0/0 Packets 0/0 Resets 0/0 RNRs 0/0 REJs 0/0 INTs 0/0

SVC 6, State: D1, Interface: Serial2
Started 0:00:04, last input 0:00:04, output 0:00:04
Connects 170093 <--> 170090 from
XOT between 131.108.165.91, 1998 and 131.108.165.92, 27896
Window size input: 2, output: 2
Packet size input: 128, output: 128
PS: 5 PR: 5 ACK: 4 Remote PR: 5 RCNT: 1 RNR: FALSE
Retransmits: 0 Timer (secs): 0 Reassembly (bytes): 0
Held Fragments/Packets: 0/0
Bytes 505/505 Packets 5/5 Resets 0/0 RNRs 0/0 REJs 0/0 INTs 0/0
    
```

Table 8-9 lists the connection description fields for virtual circuits carrying remotely switched X.25 traffic.

**Table 8-9 Show X25 VC Remote X.25 Traffic Field Descriptions**

Field	Description
PVC	Flags PVC information.
[131.108.165.92]	Indicates the IP address of the communication server remotely connecting the PVC.
connected	Identifies the state of the PVC. If the PVC is not connected, the status of the PVC will also be displayed. See Table 8-10 for the PVC status messages.
170093	Identifies the Calling (source) Address if the connection. If a Calling Address Extension was encoded in the call facilities, it will also be displayed.
170090	Identifies the Called (destination) Address of the connection. If a Called Address Extension was encoded in the call facilities, it will also be displayed.
from	Indicates the direction of the call (“from” or “to”).
XOT between...	Identifies the IP addresses and port numbers of the XOT connection.

Table 8-10 lists the PVC states that can be reported. These states are also reported by the **debug x25** command in PVC-SETUP packets (for remote PVCs only) as well as in the PVCBAD system error message. Some states apply only to remotely switched PVCs.

**Table 8-10 X.25 PVC States**

Field	Description
waiting to connect	The PVC is waiting to be processed for connecting.
dest. disconnected	The other end disconnected the PVC.
PVC/TCP connection refused	A remote PVC XOT TCP connection was tried and refused.
PVC/TCP routing error	A remote PVC XOT TCP connection routing error was reported.
PVC/TCP connect timed out	A remote PVC SOT TCP connection attempt timed out.
trying to connect via TCP	A remote PVC XOT TCP connection is established and is in the process of connecting.
awaiting PVD-SETUP reply	A remote PVC has initiated an XOT TCP connection and is waiting for a reply to the setup message.
connected	The PVC is up.
no such dest. interface	The remote destination interface was reported to be in error by the remote communication server.
dest interface is not up	The target interface’s X.25 service is down.
non-X.25 dest. interface	The target interface isn’t configured for X.25.
no such dest. PVC	The targeted PVC does not exist.
dest PVC config mismatch	The targeted PVC is already connected.
mismatched flow control values	The configured flow control values do not match.
can’t support flow control values	The window sizes or packet sizes of the PVC cannot be supported by one of its two interfaces.

## x25 accept-reverse

Use the **x25 accept-reverse** interface configuration command to instruct the communication server to accept all reverse charge calls. The **no x25 accept-reverse** command disables this facility.

**x25 accept-reverse**  
**no x25 accept-reverse**

### Syntax Description

This command has no arguments or keywords.

### Default

Disabled

### Command Mode

Interface configuration

### Usage Guidelines

This command causes the interface to accept reverse charge calls by default. This behavior also can be configured on a per-peer basis using the **x25 map** interface configuration command.

### Example

The following example sets acceptance of reverse charge calls:

```
interface serial 0
x25 accept-reverse
```

### Related Command

**x25 map**

## x25 address

Use the **x25 address** interface configuration command to set the X.121 address of a particular network interface.

**x25 address** *x.121-address*

### Syntax Description

*x.121-address* Variable-length X.121 address. The address is assigned by the X.25 network service provider.

### Default

No X.121 address is assigned.

### Command Mode

Interface configuration

### Example

The following example sets the X.121 address for the interface:

```
interface serial 0
x25 address 00000123005
```

The address must match that assigned by the X.25 network service provider.

## x25 bfe-decision

Use the **x25 bfe-decision** interface configuration command to direct how a communication server configured for **x25 bfe-emergency decision** will participate in emergency mode.

```
x25 bfe-decision {no | yes | ask}
```

### Syntax Description

- no** Prevents the communication server from participating in emergency mode and from sending address translation information to the BFE device.
- yes** Allows the communication server to participate in emergency mode and to send address translation information to the BFE when the BFE enters emergency mode. The communication server obtains this information from the table created by the **x25 remote-red** command.
- ask** Configures the communication server to display an onscreen request to enter the **bfe EXEC** command.

### Default

**no**

### Command Mode

Interface configuration

### Example

The following example shows how to configure serial interface 0 to require an EXEC command from the administrator before it participates in emergency mode. The host IP address is 21.0.0.12, and the address of the remote BFE unit is 21.0.0.1. When the BFE enters emergency mode, the communication server will prompt the administrator for EXEC command **bfe enter** to direct the communication server to participate in emergency mode.

```
interface serial 0
x25 bfe-emergency decision
x25 remote-red 21.0.0.12 remote-black 21.0.0.1
x25 bfe-decision ask
```

### Related Commands

**bfe**  
**x25 bfe-emergency**  
**x25 remote-red**



## x25 bfe-emergency

Use the **x25 bfe-emergency** interface configuration command to configure the circumstances under which the communication server participates in emergency mode.

```
x25 bfe-emergency {never | always | decision}
```

### Syntax Description

<b>never</b>	Prevents the communication server from sending address translation information to the BFE. If it does not receive address translation information, the BFE cannot open a new connection for which it does not know the address.
<b>always</b>	Allows the communication server to pass address translations to the BFE when it enters emergency mode and an address translation table has been created.
<b>decision</b>	Directs the communication server to wait until it receives a diagnostic packet from the BFE device indicating that the emergency mode window is open. The window is only open when a condition exists that allows the BFE to enter emergency mode. When the diagnostic packet is received, the communication server's participation in emergency mode depends on how it is configured using the <b>x25 bfe-decision</b> command.

### Default

Never

### Command Mode

Interface configuration

### Example

The following example shows how to configure serial interface 0 to require an EXEC command from the administrator before it participates in emergency mode. The host IP address is 21.0.0.12, and the address of the remote BFE unit is 21.0.0.1. When the BFE enters emergency mode, the communication server will prompt the administrator for EXEC command **bfe enter** to direct the communication server to participate in emergency mode.

```
interface serial 0
x25 bfe-emergency decision
x25 remote-red 21.0.0.12 remote-black 21.0.0.1
x25 bfe-decision ask
```

### Related Commands

**bfe**

**x25 bfe-decision**

## x25 default

Use the **x25 default** interface configuration command to set a default protocol. Use the **no x25 default** command to remove the protocol specified.

```
x25 default protocol  
no x25 default protocol
```

### Syntax Description

*protocol* Specifies the protocol; can only be IP, specified by the **ip** keyword.

### Default

No protocol is set.

### Command Mode

Interface configuration

### Usage Guidelines

This command specifies the protocol assumed by the communication server to interpret incoming calls with unknown Call User Data. If you do not use the **x25 default** interface configuration command, the communication server clears any incoming calls with unknown Call User Data.

### Example

The following example establishes IP as the default protocol for X.25 calls:

```
interface serial 0  
x25 default ip
```

### Related Command

**x25 map**

## x25 facility

Use the **x25 facility** interface configuration command to override the default facility settings on a per-call basis for calls originated by the communication server. Use the **no x25 facility** command to disable the facility.

**x25 facility** *facility-keyword value*  
**no x25 facility** *facility-keyword value*

### Syntax Description

*facility-keyword* User facility; see Table 8-11 for a list of supported facilities and their values.

*value* Facility value; see Table 8-11 for a list of supported facilities and their values.

Table 8-11 X.25 User Facilities

Option	Description
<b>cug</b> <i>number</i>	Specifies a closed user group (CUG) number; 1 to 99 CUGs are allowed. CUGs can be used by a public data network to create a virtual private network within the larger network and to restrict access.
<b>packetsize</b> <i>in-size out-size</i>	Specifies input packet size ( <i>in-size</i> ) and output packet size ( <i>out-size</i> ) for flow control parameter negotiation. Both values must be one of the following values: 16, 32, 64, 128, 256, 512, 1024, 2048, or 4096.
<b>window</b> <b>size</b> <i>in-size out-size</i>	Specifies the packet count for input windows ( <i>in-size</i> ) and output windows ( <i>out-size</i> ) for flow control parameter negotiation. Both values must be in the range 1 to 127 and must not be greater than the value set for the <b>x25 modulo</b> command.
<b>reverse</b>	Specifies reverse charging on all calls originated by the interface.
<b>throughput</b> <i>in out</i>	Sets the requested throughput class negotiation values for input ( <i>in</i> ) and output ( <i>out</i> ) throughput across the network. Values for <i>in</i> and <i>out</i> are in bits per second (bps) and range from 70 to 48,000 bps.
<b>transit-delay</b> <i>value</i>	Specifies a network transit delay for the duration of outgoing calls for networks that support transit delay. The transit delay value can be between 0 and 65334 milliseconds.
<b>rpoa</b> <i>name</i>	Specifies the name defined by the <b>x25 rpoa</b> command for a list of transit Recognized Private Operation Agencies (RPOAs) to use in outgoing Call Request packets.

### Default

No facility is sent.

### Command Mode

Interface configuration

## Examples

The following example specifies a transit delay value in an X.25 configuration:

```
interface serial 0
x25 facility transit-delay 24000
```

The following example specifies an RPOA name and then send the list via the X.25 user facilities:

```
x25 rpoa green_list 23 35 36
interface serial 0
x25 facility rpoa green_list
x25 map ip 131.108.170.26 10 rpoa green_list
```

## Related Command

**x25 rpoa**

## x25 hic

Use the **x25 hic** interface configuration command to set the highest incoming-only virtual circuit number.

```
x25 hic circuit-number
```

### Syntax Description

*circuit-number* Virtual circuit number from 1 through 4095, or 0 if there is no incoming-only virtual circuit range.

### Default

0

### Command Mode

Interface configuration

### Usage Guidelines

This command is applicable only if you have the X.25 switch configured for incoming only. Incoming is from the perspective of the X.25 DTE. If you do not want any outgoing calls from your DCE, configure the lic and hic values and set the ltc and htc values to 0.

### Example

The following example sets a valid incoming-only virtual circuit range of 1 to 5:

```
interface serial 0
x25 lic 1
x25 hic 5
x25 ltc 6
```

### Related Command

**x25 lic**

## x25 hoc

Use the **x25 hoc** interface configuration command to set the highest outgoing-only virtual circuit number.

**x25 hoc** *circuit-number*

### Syntax Description

*circuit-number* Virtual circuit number from 1 through 4095, or 0 if there is no outgoing-only virtual circuit range.

### Default

0

### Command Mode

Interface configuration

### Usage Guidelines

This command is applicable only if you have the X.25 switch configured for outgoing only. Outgoing is from the perspective of the X.25 DTE. If you do not want any incoming calls from your DTE, configure the loc and hoc values and set the ltc and htc values to 0.

### Example

The following example sets a valid outgoing-only virtual circuit range of 2000 to 2005:

```
interface serial 0
x25 loc 2000
x25 hoc 2005
```

### Related Command

**x25 loc**

## x25 hold-queue

Use the **x25 hold-queue** interface configuration command to modify the maximum number of packets that can be held until a virtual circuit is able to transmit. Use the **no x25 hold-queue** command without an argument to remove this command from the configuration file and restore the default value.

```
x25 hold-queue queue-size  
no x25 hold-queue [queue-size]
```

### Syntax Description

*queue-size* Defines the virtual circuit packet hold queue size. A hold queue value of 0 allows an unlimited number of packets in the hold queue. This argument is optional with the **no** form of this command.

### Default

10 packets

### Command Mode

Interface configuration

### Usage Guidelines

If you set the queue-size to 0 when using the **no x25 hold-queue** command, there will be no hold queue limit. An unlimited hold queue might be appropriate if the interface is configured with a maximum transmission unit (MTU) and IP MTU larger than the X.25 input packet size and X.25 output packet size. If you have configured an MTU and IP MTU larger than these packet sizes, our X.25 software will fragment the encapsulated IP packet into multiple X.25 frames. This fragmentation might cause the X.25 window to close and the hold queue to become full.

### Example

The following example sets the X.25 hold queue to hold 25 packets:

```
interface serial 0  
x25 hold-queue 25
```

### Related Commands

A dagger (†) indicates that the command is documented in another chapter.

```
ip mtu †  
x25 ips  
x25 ops
```

## x25 hold-vc-timer

Use the **x25 hold-vc-timer** interface configuration command to prevent overruns on some X.25 switches caused by Call Request packets. This command uses the Ignore Destination timer to prevent additional calls to a destination for a given period of time. The **no x25 hold-vc-timer** command restores the default value for the timer.

**x25 hold-vc-timer** *minutes*  
**no x25 hold-vc-timer**

### Syntax Description

*minutes*                      Number of minutes to prevent calls from going to a previously failed destination. Incoming calls still will be accepted.

### Default

0

### Command Mode

Interface configuration

### Usage Guidelines

Only Call Requests that the communication server originates will be held down; routed X.25 Call Requests are not affected by this parameter.

Upon receiving a Clear Request for an outstanding Call Request, the X.25 support code immediately tries another Call Request if it has more traffic to send, and this action might cause overrun problems.

### Example

The following example sets the Ignore Destination timer to 3 minutes:

```
interface serial 0
x25 hold-vc-timer 3
```



## x25 host

Use the **x25 host** global configuration command to define a static host name-to-address mapping.  
Use the **no x25 host** command to remove the host name.

```
x25 host name x.121-address [ cud call-user-data]  
no x25 host name
```

### Syntax Description

<i>name</i>	Host name.
<i>x.121-address</i>	The X.121 address.
<b> cud</b> <i>call-user-data</i>	(Optional) Sets the Call User Data (CUD) field in the X.25 Call Request packet.

### Default

No static host name-to-address mapping is defined.

### Command Mode

Global configuration

### Examples

The following example specifies a static address mapping:

```
x25 host Willard 4085551212
```

The following example removes a static address mapping:

```
no x25 host Willard
```

## x25 htc

Use the **x25 htc** interface configuration command to set the highest two-way virtual circuit number.

**x25 htc** *circuit-number*

### Syntax Description

*circuit-number* Virtual circuit number from 1 through 4095, or 0 if there is no two-way virtual circuit range.

### Default

1024 for X.25 network service interfaces

### Command Mode

Interface configuration

### Usage Guidelines

This command is applicable if you have the X.25 switch configured for two-way.

### Example

The following example sets a valid two-way virtual circuit range of 5 to 25:

```
interface serial 0
x25 ltc 5
x25 htc 25
```

### Related Command

**x25 ltc**

## x25 idle

The communication server can clear a switched virtual circuit (SVC) after a period of inactivity. Use the **x25 idle** interface configuration command to set this period.

**x25 idle** *minutes*

### Syntax Description

*minutes*                      Number of minutes in the idle period

### Default

0 (causes the communication server to keep the SVC open indefinitely)

### Command Mode

Interface configuration

### Usage Guidelines

Both calls originated and terminated by the communication server are cleared; switched virtual circuits are not cleared. To clear one or all virtual circuits at once, use the privileged EXEC command **clear x25-vc**.

### Example

The following example sets a 5-minute wait period before an idle circuit is cleared:

```
interface serial 2
x25 idle 5
```

### Related Command

**clear x25-vc**

## x25 ips

Use the **x25 ips** interface configuration command to set the interface default maximum input packet size to match those of the network.

**x25 ips** *bytes*

### Syntax Description

*bytes* Byte count that is one of the following: 16, 32, 64, 128, 256, 512, 1024, 2048, or 4096.

### Default

128 bytes

### Command Mode

Interface configuration

### Usage Guidelines

X.25 network connections have a default maximum input packet size set by the network administrator. Larger packet sizes require less overhead processing. To send a packet larger than the X.25 packet size over an X.25 virtual circuit, a communication server must break the packet into two or more X.25 packets with the M-bit (“more data” bit) set. The receiving device collects all packets with the M-bit set and reassembles them.

---

**Note** Set the **x25 ips** and **x25 ops** commands to the same value unless your network supports asymmetrical input and output packet sizes.

---

### Example

The following example sets the default maximum packet sizes to 512:

```
interface serial 1
x25 ips 512
x25 ops 512
```

### Related Command

**x25 ops**

## x25 ip-precedence

Use the **x25 ip-precedence** interface configuration command to enable the ability to open a new virtual circuit based on the IP Type of Service (TOS) field. The command **no x25 ip-precedence** causes the TOS field to be ignored when opening virtual circuits.

```
x25 ip-precedence  
no x25 ip-precedence
```

### Syntax Description

This command has no arguments or keywords.

### Default

The communication servers open one virtual circuit for all types of service.

### Command Mode

Interface configuration

### Usage Guidelines

There is a problem associated with this feature in that some hosts send nonstandard data in the TOS field, thus causing multiple, wasteful virtual circuits to be created.

### Example

The following example allows new virtual circuits based on the TOS field:

```
interface serial 3  
x25 ip-precedence
```

## x25 lic

Use the **x25 lic** interface configuration command to set the lowest incoming-only virtual circuit number.

**x25 lic** *circuit-number*

### Syntax Description

*circuit-number* Virtual circuit number from 1 through 4095, or 0 if there is no incoming-only virtual circuit range.

### Default

0

### Command Mode

Interface configuration

### Usage Guidelines

This command is applicable only if you have the X.25 switch configured for incoming only. Incoming is from the perspective of the X.25 DTE. If you do not want any outgoing calls from your DCE, configure the lic and hic values and set the ltc and htc values to 0.

This command is applicable if you have the X.25 switch configured for two way virtual circuits.

### Example

The following example sets a valid incoming-only virtual circuit range of 1 to 5:

```
interface serial 0
x25 lic 1
x25 hic 5
x25 ltc 6
```

### Related Command

**x25 hic**

## x25 linkrestart

Use the **x25 linkrestart** interface configuration command to force a packet-level restart when the link level resets. This command restarts X.25 Level 3 when errors occur in Level 2 (LAPB). The **no x25 linkrestart** command disables this function.

```
x25 linkrestart  
no x25 linkrestart
```

### Syntax Description

This command has no arguments or keywords.

### Default

Forcing packet-level restarts is the default and is necessary for networks that expect this behavior.

### Command Mode

Interface configuration

### Example

The following example disables the link-level restart:

```
interface serial 3  
no x25 linkrestart
```

## x25 loc

Use the **x25 loc** interface configuration command to set the lowest outgoing-only virtual circuit number.

**x25 loc** *circuit-number*

### Syntax Description

*circuit-number* Virtual circuit number from 1 through 4095, or 0 if there is no outgoing-only virtual circuit range.

### Default

0

### Command Mode

Interface configuration

### Usage Guidelines

This command is applicable only if you have the X.25 switch configured for outgoing only. Outgoing is from the perspective of the X.25 DTE. If you do not want any incoming calls from your DTE, configure the loc and hoc values and set the ltc and htc values to 0.

### Example

The following example sets a valid outgoing-only virtual circuit range of 2000 to 2005:

```
interface serial 0
x25 loc 2000
x25 hoc 2005
```

### Related Command

**x25 hoc**



## x25 ltc

Use the **x25 ltc** interface configuration command to set the lowest two-way virtual circuit number.

**x25 ltc** *circuit-number*

### Syntax Description

*circuit-number* Virtual circuit number from 1 through 4095, or 0 if there is no two-way virtual circuit range.

### Default

1

### Command Mode

Interface configuration

### Usage Guidelines

This command is applicable if you have the X.25 switch configured for two-way.

### Example

The following example sets a valid two-way virtual circuit range of 5 to 25:

```
interface serial 0
x25 ltc 5
x25 htc 25
```

### Related Command

**x25 htc**

## x25 map

To set up the LAN protocols-to-remote host mapping, use the **x25 map** interface configuration command. To retract a prior mapping, use the **no** form of the command with the appropriate network protocol(s) and X.121 address argument.

```
x25 map protocol1 address1 [protocol2 address2...protocol9 address9] x.121-address [option]
no x25 map protocol address x.121-address
```

### Syntax Description

<i>protocol</i>	The protocol type. Supported protocols are entered by keyword, as listed in Table 8-12. You can specify as many as nine protocols in one command line.
<i>address</i>	Protocol address.
<i>x.121-address</i>	Specifies the X.121 address of the remote host.
<i>option</i>	(Optional) Provides additional functionality or allows X.25 facilities to be specified for originated calls. Can be any of the options listed in Table 8-13.

Table 8-12 Protocols Supported by X.25

Keyword	Protocol
<b>compressedtcp</b>	TCP header compression
<b>ip</b>	IP
<b>ipx</b>	Novell IPX

Table 8-13 X.25 Map Options

Option	Description
<b>compress</b>	Specifies X.25 compression should be used when mapping the traffic to this host. Each virtual circuit established for compressed traffic uses a significant amount of memory (for a table of learned data patterns) and for computation (for compression and decompression of all data). It is recommended that compression be used with careful consideration to its impact on overall communication server performance.
<b>method</b> [ <b>cisco</b>   <b>ietf</b>   <b>snap</b>   <b>multi</b> ]	Specifies the encapsulation method. The choices are as follows: <ul style="list-style-type: none"> <li>• <b>cisco</b>—Cisco’s proprietary encapsulation; not available if more than one protocol is to be carried.</li> <li>• <b>ietf</b>—Default RFC 1356 operation: protocol identification of single-protocol virtual circuits and protocol identification within multiprotocol virtual circuits uses the standard encoding, which is compatible with RFC 877. Multiprotocol virtual circuits are used only if needed.</li> <li>• <b>snap</b>—RFC 1356 operation where IP is identified using SNAP rather than the standard IETF method (the standard method is compatible with RFC 877).</li> <li>• <b>multi</b>—forces a map that specifies a single protocol to set up a multiprotocol virtual circuit when a call is originated; also forces a single-protocol PVC to use multiprotocol data identification methods for all datagrams sent and received.</li> </ul>
<b>no-incoming</b>	Use the map only to originate calls.
<b>no-outgoing</b>	Do not originate calls when using the map.
<b>idle</b> <i>minutes</i>	Specifies an idle timeout for Calls other than the interface default; 0 minutes disables the idle timeout.
<b>reverse</b>	Specifies reverse charging for outgoing calls.
<b>accept-reverse</b>	Causes the communication server to accept incoming reverse-charged calls. If this option is not present, the system clears reverse charged calls unless the interface accepts all reverse charged calls.
<b>broadcast</b>	Causes the system to direct any broadcasts sent through this interface to the specified X.121 address. This option also simplifies the configuration of OSPF; see “Usage Guidelines” for more detail.
<b>cug</b> <i>number</i>	Specifies a closed user group number (from 1 to 99) for the mapping in an outgoing call.
<b>nvc</b> <i>count</i>	Sets the maximum number of virtual circuits for this map/host. The default <i>count</i> is the <b>x25 nvc</b> setting of the interface. A maximum number of eight virtual circuits can be configured for each map. Compressed TCP may only use 1 virtual circuits.
<b>packetsize</b> <i>in-size out-size</i>	Proposes maximum input packet size ( <i>in-size</i> ) and maximum output packet size ( <i>out-size</i> ) for an outgoing call. Both values are typically the same and must be one of the following values: 16, 32, 64, 128, 256, 512, 1024, 2048, or 4096.
<b>window</b> <i>size in-size out-size</i>	Proposes the packet count for input window ( <i>in-size</i> ) and output window ( <i>out-size</i> ) for an outgoing call. Both values are typically the same, must be in the range 1 to 127, and must be less than the value set by the <b>x25 modulo</b> command.

Option	Description
<b>throughput</b> <i>in out</i>	Sets the requested throughput class values for input ( <i>in</i> ) and output ( <i>out</i> ) throughput across the network for an outgoing call. Values for <i>in</i> and <i>out</i> are in bits per second (bps) and range from 75 to 48000 bps.
<b>transit-delay</b> <i>number</i>	Specifies the transit delay value in milliseconds (0 to 65534) for an outgoing call, for networks that support transit delay.
<b>nuid</b> <i>username password</i>	Specifies that a network ID facility be sent in the outgoing call with the specified TACACS username and password (in a format defined by Cisco). This option should be used only when connecting to another Cisco communication server. The combined length of the username and password should not exceed 127 characters.
<b>nudata</b> <i>string</i>	Specifies the network user identification in a format determined by the network administrator (as allowed by the standards). This option is provided for connecting to non-Cisco equipment that requires an NUID facility. The string should not exceed 130 characters and must be enclosed in quotation marks (“ ”) if there are any spaces present.
<b>rpoa</b> <i>name</i>	Specifies the name defined by the <b>x25 rpoa</b> command for a list of transit RPOAs to use in outgoing Call Request packets.
<b>passive</b>	Specifies that the X.25 interface should send compressed outgoing TCP datagrams only if they were already compressed when they were received. This option is available only for compressed TCP maps.

## Default

No LAN protocol-to-remote host mapping is set up.

## Command Mode

Interface configuration

## Usage Guidelines

Because no defined protocol can dynamically determine LAN protocol-to-remote host mappings, you must enter all of the information for each host with which the communication server may exchange X.25 encapsulated traffic.

Two methods are available to encapsulate traffic, Cisco’s long-available encapsulation method and the IETF’s standard method (defined in RFC 1356); the latter allows hosts to exchange several protocols over a single virtual circuit. Cisco’s encapsulation method is the default (for backward compatibility, unless the interface configuration specified **ietf**).

When you configure multiprotocol maps, you can specify a maximum of nine protocol and address pairs in an **x25 map** command. However, you can specify a protocol once only. For example, you can specify the IP protocol and an IP address, but you cannot specify another IP address. If **compressedtcp** and **ip** are both specified, the same IP address must be used.

Bridging is supported only using Cisco’s traditional encapsulation method. For correct operation, bridging maps must specify the **broadcast** option.

Because most datagram routing protocols rely on broadcasts or multicasts to send routing information to their neighbors, the **broadcast** keyword is needed to run such routing protocols over X.25.

Encapsulation maps might also specify that traffic between the two hosts should be compressed, thus increasing the effective bandwidth between them at the expense of memory and computation time. Each compression virtual circuit requires memory and computation resources, so compression should be used with care and monitored to maintain acceptable resource usage and overall communication server performance.

OSPF treats a nonbroadcast, multiaccess network such as X.25 much the same way it treats a broadcast network in that it requires selection of a designated communication server. In previous releases, this required manual assignment in the OSPF configuration using the **neighbor interface** router configuration command. When the **x25 map** command is included in the configuration with the broadcast, and the **ip ospf network** command (with the **broadcast** keyword) is configured, there is no need to configure any neighbors manually. OSPF will now run over the X.25 network as a broadcast network. (Refer to the **ip ospf network** interface command for more detail.)

---

**Note** The OSPF broadcast mechanism assumes that IP class D addresses are never used for regular traffic over X.25.

---

You can modify the options of an **x25 map** command by restating the complete set of protocols and addresses specified for the map, followed by the desired options. To delete a map command, you must also specify the complete set of protocols and addresses; the options can be omitted when deleting a map.

Once defined, a map's protocols and addresses cannot be changed; this is because the communication server cannot determine whether you want to add to, delete from, or modify an existing map's protocol and address specification (or simply mistyped the command). To change a map's protocol and address specification, you must delete it and create a new map.

A given protocol/address pair cannot be used in more than one map on the same interface.

## Examples

The following example maps IP address 131.08.2.5 to X.121 address 000000010300. The **broadcast** keyword directs any broadcasts sent through this interface to the specified X.121 address.

```
interface serial 0
x25 map ip 131.08.2.5 000000010300 broadcast
```

The following example sets an RPOA name to be used when originating connections:

```
x25 rpoa green_list 23 35 36
interface serial 0
x25 facility rpoa green_list
x25 map ip 131.108.170.26 10 rpoa green_list
```

The following example specifies a network user identifier (NUID) facility to send on calls originated for the address map:

```
interface serial 0
x25 map IP 131.108.174.32 2 nudata "Network User ID 35"
```

Strings can be quoted, but quotation marks are not required unless embedded blanks are present.

### Related Commands

**ip ospf network**  
**show x25 map**  
**x25 compress**  
**x25 map bridge**  
**x25 map cmns**  
**x25 map compressedtcp**  
**x25 rpoa**

## x25 map compressedtcp

Use the **x25 map compressedtcp** interface configuration command to map compressed TCP traffic to X.121 addresses. The **no x25 map compressedtcp** command disables TCP header compression for the link.

```
x25 map compressedtcp ip-address x.121-address [options]  
no x25 map compressedtcp ip-address x.121-address
```

### Syntax Description

<i>ip-address</i>	The IP address.
<i>x.121-address</i>	The X.121 address.
<i>options</i>	(Optional) The same options as those for the <b>x25 map</b> command described in Table 8-13.

### Default

No mapping of compressed TCP traffic to X.121 addresses.

### Command Mode

Interface configuration

### Usage Guidelines

The Call User Data of compressed TCP calls is the single byte 0xD8.

TCP header compression is supported over X.25 links. The implementation of compressed TCP over X.25 uses a virtual circuit to pass the compressed packets. The noncompressed packets use another virtual circuit. The NVC map option cannot be used for TCP header compression, as only one virtual circuit can carry compressed TCP header traffic to a given host.

### Example

The following example establishes packet compression on serial interface 4:

```
interface serial 4  
ip tcp header-compression  
x25 map compressedtcp 131.08.2.5 000000010300
```

### Related Command

**x25 map**

## x25 modulo

Use the **x25 modulo** interface configuration command to set the packet numbering modulo.

**x25 modulo** *modulus*

### Syntax Description

*modulus* Either 8 or 128. The value of the modulo parameter must agree with that of the device on the other end of the X.25 link.

### Default

8

### Command Mode

Interface configuration

### Usage Guidelines

X.25 supports flow control with a sliding window sequence count. The window counter restarts at zero upon reaching the upper limit, which is called the *window modulus*.

### Example

The following example sets the window modulus to 128:

```
interface serial 0
x25 modulo 128
```

### Related Commands

**x25 win**

**x25 wout**



## x25 nvc

Use the **x25 nvc** interface configuration command to specify the maximum number of switched virtual circuits (SVCs) that a protocol can have open simultaneously to one host. To increase throughput across networks, you can establish up to eight switched virtual circuits to a host.

**x25 nvc** *count*

### Syntax Description

*count* Circuit count from 1 to 8. A maximum of eight virtual circuits can be configured for each protocol/host pair. Protocols that do not tolerate out-of-order delivery, such as encapsulated TCP header compression, will only use one virtual circuit despite this value.

### Default

1

### Command Mode

Interface configuration

### Usage Guidelines

When the windows and output queues of all existing connections to a host are full, a new virtual circuit will be opened to the designated circuit count. If a new connection cannot be opened, the data is dropped.

---

**Note** The *count* value specified for **x25 nvc** affects the default value for the number of SVCs. It does not affect the NVC value for any **x25 map** commands that already have been configured.

---

### Example

The following example sets the maximum number of SVCs that can be open simultaneously to 4:

```
interface serial 0
x25 nvc 4
```

## x25 ops

Use the **x25 ops** interface configuration command to set the interface default maximum output packet size to match those of the network.

**x25 ops** *bytes*

### Syntax Description

*bytes* Byte count in the range of 16 through 1024

### Default

128 bytes

### Command Mode

Interface configuration

### Usage Guidelines

X.25 networks use maximum input packet sizes set by the network administration. Larger packet sizes are better because smaller packets require more overhead processing. To send a packet larger than the X.25 packet size over an X.25 virtual circuit, a communication server must break the packet into two or more X.25 packets with the M-bit (“more data” bit) set. The receiving device collects all packets with the M-bit set and reassembles them.

---

**Note** Set the **x25 ips** and **x25 ops** commands to the same value unless your network supports asymmetry between input and output packets.

---

### Example

The following example sets the default maximum packet sizes to 512:

```
interface serial 1
x25 ips 512
x25 ops 512
```

### Related Command

**x25 ips**

## x25 pvc (encapsulating)

To establish an encapsulation permanent virtual circuit (PVC), use the encapsulating version of the **x25 pvc** interface configuration command. To delete the PVC, use the **no x25 pvc** command with the appropriate channel number.

```
x25 pvc circuit protocol address [protocol address] x.121-address [option]
no x25 pvc circuit
```

### Syntax Description

<i>circuit</i>	Virtual-circuit channel number, which must be less than the virtual circuits assigned to the switched virtual circuits (SVCs).
<i>protocol</i>	Protocol type, entered by keyword. Supported protocols are listed in Table 8-14. As many as nine protocol and address pairs can be specified in one command line.
<i>address</i>	Protocol address of the host at the other end of the PVC.
<i>option</i>	(Optional) Provides additional functionality or allows X.25 parameters to be specified for the PVC. Can be any of the options listed in Table 8-15.

Table 8-14 Protocols Supported by X.25 PVCs

Keyword	Protocol
<b>ip</b>	IP
<b>novell</b>	Novell IPX
<b>compressedtcp</b>	TCP header compression

Table 8-15 PVC Options

Option	Description
broadcast	Causes the communication server to direct any broadcasts sent through this interface to this PVC. This option also simplifies the configuration of OSPF; see the “Usage Guidelines” section for more information.
<b>method</b> { <b>cisco</b>   <b>ietf</b>   <b>snap</b>   <b>multi</b> }	Specifies the encapsulation method. The choices are as follows: <ul style="list-style-type: none"> <li>• <b>cisco</b>—Single protocol encapsulation; not available if more than one protocol is carried.</li> <li>• <b>ietf</b>—Default RFC 1356 operation; single-protocol encapsulation unless more than one protocol is carried, and protocol identification (when carrying more than one protocol).</li> <li>• <b>snap</b>—RFC 1356 operation where IP is identified (when carrying more than one protocol) using the SNAP encoding.</li> <li>• <b>multi</b>—Multiprotocol encapsulation used on the PVC.</li> </ul>
<b>packetsize</b> <i>in-size out-size</i>	Maximum input packet size ( <i>in-size</i> ) and output packet size ( <i>out-size</i> ) for the PVC. Both values are typically the same and must be one of the following values: 16, 32, 64, 128, 256, 512, 1024, 2048, or 4096.
passive	Specifies that transmitted TCP datagrams will be compressed only if they were received compressed. This option is available only for PVCs carrying compressed TCP header traffic.
<b>window</b> <i>size in-size out-size</i>	Packet count for input window ( <i>in-size</i> ) and output window ( <i>out-size</i> ) for the PVC. Both values are typically the same, must be in the range 1 to 127, and must be less than the value set for the <b>x25 modulo</b> command.

### Default

No encapsulation PVC is established. The PVC window and maximum packet sizes default to the map values or the interface default values.

### Command Mode

Interface configuration

### Usage Guidelines

PVCs are not supported for ISO CMNS.

You no longer need to specify a datagram protocol/address mapping before you can set up a PVC; a map is implied from the PVC configuration. Configurations generated by the communication server will no longer specify a map for encapsulating PVCs.

An X.121 address must be specified for the PVC, much as is done for an **x25 map** command, although the address does not appear in the PVC data exchange. When configuring a PVC to carry CLNS traffic, the X.121 address is used as the SNPA to associate the PVC with a CLNS neighbor configuration.

### Related Command

**x25 map**

## x25 remote-red

Use the **x25 remote-red** interface configuration command to set up the table that lists the BFE nodes (host or gateways) to which the communication server will send packets.

```
x25 remote-red host-ip-address remote-black blacker-ip-address
```

### Syntax Description

<i>host-ip-address</i>	IP address of the host or a communication server that the packets are being sent to.
<b>remote-black</b>	Delimits the addresses for the table being built.
<i>blacker-ip-address</i>	IP address of the remote BFE device in front of the host to which the packet is being sent.

### Default

No BFE node is set up.

### Command Mode

Interface configuration

### Usage Guidelines

The table that results from this command provides the address translation information the communication server sends to the BFE when it is in emergency mode.

### Example

The following example sets up a short table of BFE nodes for serial interface 0:

```
interface serial 0
x25 remote-red 131.108.9.3 remote-black 131.108.9.13
x25 remote-red 192.108.15.1 remote-black 192.108.15.26
```

### Related Command

**x25 bfe-decision**

## x25 rpoa

Use the **x25 rpoa** global configuration command to set the packet network carrier. The **no x25 rpoa** command removes the specified name.

```
x25 rpoa name number1 [number2...number10]  
no x25 rpoa name
```

### Syntax Description

<i>name</i>	Recognized Private Operating Agency (RPOA), which must be unique with respect to all other RPOA names. It is used in the <b>x25 facility</b> and <b>x25 map</b> interface configuration commands.
<i>number</i>	Number that is used to describe an RPOA; up to 10 numbers are accepted.

### Default

No packet network carrier

### Command Mode

Global configuration

### Usage Guidelines

This command lists transit RPOAs, referenced by name.

### Example

The following example sets an RPOA name and then sends the list via the X.25 user facilities:

```
x25 rpoa green_list 23 35 36  
interface serial 0  
x25 facility rpoa green_list  
x25 map ip 131.108.170.26 10 rpoa green_list
```

### Related Commands

**x25 facility**

**x25 map**

## x25 suppress-called-address

Use the **x25 suppress-called-address** interface configuration command to omit the called address in outgoing calls. Use the **no x25 suppress-called-address** command to reset this command to the default state.

```
x25 suppress-called-address  
no x25 suppress-called-address
```

### Syntax Description

This command has no arguments or keywords.

### Default

The called address is sent by default.

### Command Mode

Interface configuration

### Usage Guidelines

This command omits the called (destination) X.121 address in Call Request packets and is required for networks that expect only subaddresses in the called address field.

### Example

The following example suppresses or omits the called address in Call Request packets:

```
interface serial 0  
x25 suppress-called-address
```

## x25 suppress-calling-address

Use the **x25 suppress-calling-address** interface configuration command to omit the calling address in outgoing calls. Use the **no x25 suppress-calling-address** command to reset this command to the default state.

**x25 suppress-calling-address**  
**no x25 suppress-calling-address**

### Syntax Description

This command has no arguments or keywords.

### Default

The calling address is sent by default.

### Command Mode

Interface configuration

### Usage Guidelines

This command omits the calling (source) X.121 address in Call Request packets and is required for networks that expect only subaddresses in the calling address field.

### Example

The following example suppresses or omits the calling address in Call Request packets:

```
interface serial 0
x25 suppress-calling-address
```



## x25 t10

Use the **x25 t10** interface configuration command to set the limit for the Restart Request retransmission timer (T10) on DCE devices.

**x25 t10** *seconds*

### Syntax Description

*seconds*                      Amount of time in seconds

### Default

60 seconds

### Command Mode

Interface configuration

### Example

The following example sets the T10 timer to 30 seconds:

```
interface serial 0
x25 t10 30
```

## x25 t11

Use the **x25 t11** interface configuration command to set the limit for the Call Request Completion timer (T11) on DCE devices.

**x25 t11** *seconds*

### Syntax Description

*seconds*                      Amount of time in seconds

### Default

180 seconds

### Command Mode

Interface configuration

### Example

The following example sets the T11 timer to 90 seconds:

```
interface serial 0
x25 t11 90
```

## x25 t12

Use the **x25 t12** interface configuration command to set the limit for the Reset Request retransmission timer (T12) on DCE devices.

**x25 t12** *seconds*

### Syntax Description

*seconds*                      Amount of time in seconds

### Default

60 seconds

### Command Mode

Interface configuration

### Example

The following example sets the T12 timer to 30 seconds:

```
interface serial 0
x25 t12 30
```

## x25 t13

Use the **x25 t13** interface configuration command to set the limit for the Clear Request retransmission timer (T13) on DCE devices.

**x25 t13** *seconds*

### Syntax Description

*seconds*                      Amount of time in seconds

### Default

60 seconds

### Command Mode

Interface configuration

### Example

The following example sets the T13 timer to 30 seconds:

```
interface serial 0
x25 t13 30
```

## x25 t20

Use the **x25 t20** interface configuration command to set the limit for the Restart Request retransmission timer (T20) on DTE devices.

**x25 t20** *seconds*

### Syntax Description

*seconds*                    Amount of time in seconds

### Default

180 seconds

### Command Mode

Interface configuration

### Example

The following example sets the T20 timer to 90 seconds:

```
interface serial 0
x25 t20 90
```

## x25 t21

Use the **x25 t21** interface configuration command to set the limit for the Call Request Completion timer (T21) on DTE devices.

**x25 t21** *seconds*

### Syntax Description

*seconds*                      Amount of time in seconds

### Default

200 seconds

### Command Mode

Interface configuration

### Example

The following example sets the T21 timer to 100 seconds:

```
interface serial 0
x25 t21 100
```

## x25 t22

Use the **x25 t22** interface configuration command to set the limit for the Reset Request retransmission timer (T22) on DTE devices.

**x25 t22** *seconds*

### Syntax Description

*seconds*                      Amount of time in seconds

### Default

180 seconds

### Command Mode

Interface configuration

### Example

The following example sets the T22 timer to 90 seconds:

```
interface serial 0
x25 t22 90
```

## x25 t23

Use the **x25 t23** interface configuration command to set the limit for the Clear Request retransmission timer (T23) on DTE devices.

**x25 t23** *seconds*

### Syntax Description

*seconds*                      Amount of time in seconds

### Default

180 seconds

### Command Mode

Interface configuration

### Example

The following example sets the T23 timer to 90 seconds:

```
interface serial 0
x25 t23 90
```



## x25 th

Use the **x25 th** interface configuration command to instruct the communication server to send an acknowledgment packet when it has received a threshold of data packets it has not acknowledged, instead of waiting until its input window is full. This command improves line responsiveness at the expense of bandwidth.

**x25 th** *delay-count*

### Syntax Description

*delay-count* Value between zero and the input window size. A value of 1 sends one Receiver Ready acknowledgment per packet at all times.

### Default

0 (which disables the delayed acknowledgment strategy)

### Command Mode

Interface configuration

### Usage Guidelines

The communication server sends acknowledgment packets when the number of input packets reaches the count you specify, providing there are no other packets to send. For example, if you specify a count of 1, the communication server can send an acknowledgment per input packet.

### Example

The following example sends five Receiver Ready acknowledgments per packet as an input packet delay:

```
interface serial 1
x25 th 5
```

### Related Commands

**x25 win**  
**x25 wout**

## x25 win

Use the **x25 win** interface configuration command to change the default maximum number of unacknowledged incoming packets for the interface.

**x25 win** *packets*

### Syntax Description

*packets* Packet count that can range from 1 to one less than the window modulus

### Default

2 packets

### Command Mode

Interface configuration

### Usage Guidelines

This command determines how many packets the communication server can receive before sending an X.25 acknowledgment. To maintain high bandwidth utilization, assign this limit the largest number that the network allows.

---

**Note** Set **x25 win** and **x25 wout** to the same value unless your network supports asymmetry between input and output window sizes.

---

### Example

The following example specifies that five packets must be received before sending an X.25 acknowledgment:

```
interface serial 1
x25 win 5
```

### Related Commands

**x25 modulo**

**x25 th**

**x25 wout**

## x25 wout

Use the **x25 wout** interface configuration command to change the default maximum number of unacknowledged packets to allow.

**x25 wout** *packets*

### Syntax Description

*packets* Packet count that can range from 1 to the window modulus

### Default

2 packets

### Command Mode

Interface configuration

### Usage Guidelines

This command determines the default number of packets the communication server can send before waiting for an X.25 acknowledgment. To maintain high bandwidth utilization, assign this limit the largest number that the network allows.

---

**Note** Set **x25 win** and **x25 wout** to the same value unless your network supports asymmetry between input and output window sizes.

---

### Example

The following example specifies an upper limit of five for the number of outstanding unacknowledged packets for the output window:

```
interface serial 1
  x25 wout 5
```

### Related Commands

**x25 modulo**

**x25 th**

**x25 win**

## x29 access-list

To limit access to the communication server from certain X.25 hosts, use the **x29 access-list** global configuration command. To delete an entire access list, use the **no** form of this command.

```
x29 access-list access-list-number { deny | permit } x.121-address  
no x29 access-list access-list-number
```

### Syntax Description

<i>access-list-number</i>	Number of the access list. It can be a value between 1 and 199.
<b>deny</b>	Denies access and clears call requests immediately.
<b>permit</b>	Permits access to the protocol translator.
<i>x.121-address</i>	X.121 address, with or without regular expression pattern-matching characters, with which to compare for access.

### Default

No access lists are defined.

### Command Mode

Global

### Usage Guidelines

An access list can contain any number of access list items. The list are processed in the order in which you entered them, with the first match causing the permit or deny condition. If an X.121 address does not match any of the regular expression in the access list, access will be denied.

Access lists take advantage of the message field defined by Recommendation X.29, which describes procedures for exchanging data between two PADs or a PAD and a DTE device.

The UNIX-style regular expression characters allow for pattern matching of characters and character strings in the address. Various pattern-matching constructions are available that will allow many addresses to be matched by a single regular expressions. Refer to the “Regular Expressions” appendix in this manual for more information.

### Example

The following example permits connections to hosts with addresses beginning with the string 31370:

```
x29 access-list 2 permit ^31370
```

## x29 profile

To create a PAD profile script for use by the **translate** command, use the **x29 profile** global configuration command.

```
x29 profile name parameter:value [parameter:value]
```

### Syntax Description

<i>name</i>	Name of the PAD profile script.
<i>parameter:value</i>	X.3 PAD parameter number and value separated by a colon. You can specify multiple parameter-value pairs.

### Default

No PAD profile script is created.

### Command Mode

Global configuration

### Usage Guidelines

When an X.25 connection is established, the communication server acts as if an X.29 SET PARAMETER packet had been sent containing the parameters and values set by the **x29 profile** command and sets the communication server accordingly.

### Example

The following profile script turns local edit mode on when the connection is made and establishes local echo and line termination upon receipt of a Return. The name “linemode” is used with the **translate** command to effect use of this script.

```
x29 profile linemode 2:1 3:2 15:1
```

### Related Command

**translate**

