



# Protocol Translation and Virtual Asynchronous Device Commands

---

Protocol translation provides transparent translation between systems running different protocols. The Cisco IOS software supports two-way virtual terminal protocol translation between nodes running X.25, LAT, and Telnet.

This chapter describes the commands that you use to configure protocol translation.

For protocol translation configuration information and examples, see the chapter “Configuring Protocol Translation and Virtual Asynchronous Devices” in the *Dial Solutions Configuration Guide*.

## service pt-vty-logging

To log the X.121 calling address, Call User Data (CUD), and the IP address assigned to a VTY asynchronous connection, use the **service pt-vty-logging** global configuration command. Use the **no** form of this command to disable this function.

**service pt-vty-logging**  
**no service pt-vty-logging**

### Syntax Description

This command has no arguments or keywords.

### Default

This feature is disabled.

### Command Mode

Global configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

This command permits you to log the X.121 calling address, Call User Data (CUD), and the IP address assigned to a VTY asynchronous connection and direct this information to the console, an internal buffer, or a UNIX syslog server, depending on the logging configuration command you use. This authentication information can be used to associate an incoming PAD VTY-asynchronous connection with an IP address.

---

Note By default, the Cisco IOS software displays all messages to the console terminal.

---

### Example

The following example enables you to log the X.121 calling address, Call User Data (CUD), and the IP address assigned to a VTY asynchronous connection and save this information to a syslog server:

```
service pt-vty-logging
```

The following is sample output resulting from the **service pt-vty-logging** command:

```
01:24:31: PAD18: call from 00011890 on LCI 10 PID 1 0 0 0 CUD "xyz"
```

Table 1 describes the fields shown in the output.

**Table 1 Service PT-VTY-Logging Field Descriptions**

<b>Field</b>	<b>Description</b>
01:24:31:	Time stamp.
PAD18:	Active VTY line number using the PAD connection.
00011890	The source/calling address.
on LCI 10	Incoming call is initiated on Logical Channel 10.
PID 1 0 0 0	The PAD Protocol Identifier is "01000000."
CUD "xyz"	Call User Data "xyz." If no CUD is available, this field will appear as follows: CUD ""

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**logging**

**logging buffered**

## show interfaces virtual-access

Use the **show interfaces virtual-access EXEC** command to display information about virtual access interfaces.

**show interfaces virtual-access *number***

### Syntax Description

*number*                      Number of the virtual terminal (VTY) line on which the virtual access interface has been created.

### Command Mode

EXEC

### Usage Guidelines

This command first appeared in Cisco IOS Release 11.3.

To identify the number of the VTY line on which the virtual access interface was created, issue the **show users EXEC** command included in this feature chapter.

### Sample Display

The following is sample output from the **show interfaces virtual-access** command:

```
router# show interface virtual-access 2

Virtual-Access2 is up, line protocol is up
Hardware is Virtual Access interface
Interface is unnumbered. Using address of Ethernet0 (10.0.21.14)
MTU 1500 bytes, BW 9 Kbit, DLY 100000 usec, rely 255/255, load 1/255
Encapsulation PPP, loopback not set, keepalive not set
DTR is pulsed for 0 seconds on reset
LCP Open
Open: IPCP
Last input 00:00:06, output 00:00:05, output hang never
Last clearing of "show interface" counters 00:14:58
Input queue: 1/75/0 (size/max/drops); Total output drops: 0
Queueing strategy: weighted fair
Output queue: 0/64/0 (size/threshold/drops)
  Conversations 0/1 (active/max active)
  Reserved Conversations 0/0 (allocated/max allocated)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
  4 packets input, 76 bytes, 0 no buffer
  Received 0 broadcasts, 0 runts, 0 giants
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  8 packets output, 330 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets
  0 output buffer failures, 0 output buffers swapped out
  0 carrier transitions
```

Table 2 describes the fields shown in this sample display.

**Table 2 Show Interfaces Virtual-Access Field Descriptions**

Field	Description
Virtual-Access ... is {up   down   administratively down}	Indicates whether the interface is currently active (whether carrier detect is present), inactive, or has been taken down by an administrator.
line protocol is {up   down   administratively down}	Indicates whether the software processes that handle the line protocol think the line is usable (that is, whether keepalives are successful).
Hardware is Virtual Access interface	Type of interface. In this case, the interface is a dynamically created virtual access interface existing on a VTY line.
Internet address   interface is unnumbered	IP address, or IP unnumbered for the line. If unnumbered, the output lists the interface and IP address to which the line is assigned (Ethernet0 at 10.0.21.14 in this example).
MTU	Maximum transmission unit for packets on the virtual access interface.
BW	Bandwidth of the virtual access interface in kilobits per second.
DLY	Delay of the virtual access interface in microseconds.
rely	Reliability of the virtual access interface as a fraction of 255 (255/255 is 100% reliability), calculated as an exponential average over five minutes.
load	Load on the virtual access interface as a fraction of 255 (255/255 is completely saturated), calculated as an exponential average over five minutes. The calculation uses the value from the <b>bandwidth</b> interface configuration command.
Encapsulation	Encapsulation method assigned to the virtual access interface.
loopback	Test in which signals are sent and then directed back toward the source at some point along the communication path. Used to test network interface usability.
keepalive	Interval set for keepalive packets on the interface. If keepalives have not been enabled, the message is "keepalive not set."
DTR	Data Terminal Ready. An RS232-C circuit that is activated to let the DCE know when the DTE is ready to send and receive data.
LCP open   closed   req sent	Link control protocol (for PPP only; not for SLIP). LCP must come to the open state before any useful traffic can cross the link.
Open IPCP   IPXCP   ATCP	IPCP is IP control protocol for PPP, IPXCP is IPX control protocol for PPP, ATCP is AppleTalk control protocol for PPP. Network control protocols (NCPs) for the PPP suite. The NCP is negotiated after the LCP opens. The NCP must come into the open state before useful traffic can cross the link.
Last input	Number of hours, minutes, and seconds since the last packet was successfully received by a virtual access interface. Useful for knowing when a dead interface failed.
output	Number of hours, minutes, and seconds since the last packet was successfully transmitted by a virtual access interface.

**Table 2 Show Interfaces Virtual-Access Field Descriptions (Continued)**

Field	Description
output hang	Number of hours, minutes, and seconds (or never) since the virtual access interface was last reset because of a transmission that took too long. When the number of hours in any of the “last” fields exceeds 24 hours, the number of days and hours is printed. If that field overflows, asterisks are printed.
Last clearing	Time at which the counters that measure cumulative statistics (such as number of bytes transmitted and received) shown in this report were last reset to zero. Note that variables that might affect routing (for example, load and reliability) are not cleared when the counters are cleared.  *** indicates the elapsed time is too large to be displayed. 0:00:00 indicates the counters were cleared more than 2 <sup>31</sup> ms (and less than 2 <sup>32</sup> ms) ago.
Input queue, drops	Number of packets in input queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped due to a full queue.
Queueing strategy	Type of queueing selected to prioritize network traffic. The options are first-come-first-serve (FCFS) queueing, weighted fair queueing, priority queueing, and custom queueing.
Output queue	Number of packets in output queues. Each number is followed by a slash, the maximum size of the queue, and the number of packets dropped due to a full queue.
Conversations	Number of weighted fair queueing conversations.
Reserved Conversations	Number of reserved weighted fair queueing conversations. The example shows the number of allocated conversations divided by the number of maximum allocated conversations. In this case, there have been 0 reserved conversations.
Five minute input rate, Five minute output rate	Average number of bits and packets transmitted per second in the last five minutes.
packets input	Total number of error-free packets received by the system.
bytes	Total number of bytes, including data and MAC encapsulation, in the error free packets received by the system.
no buffer	Number of received packets discarded because there was no buffer space in the main system. Compare with ignored count. Broadcast storms on Ethernets and bursts of noise on serial lines are often responsible for no input buffer events.
broadcasts	Total number of broadcast or multicast packets received by the virtual access interface.
runts	Number of packets that are discarded because they are smaller than the medium’s minimum packet size.
giants	Number of packets that are discarded because they exceed the medium’s maximum packet size.
input errors	Total number of no buffer, runts, giants, CRCs, frame, overrun, ignored, and abort counts. Other input-related errors can also increment the count, so that this sum might not balance with the other counts.

**Table 2 Show Interfaces Virtual-Access Field Descriptions (Continued)**

<b>Field</b>	<b>Description</b>
CRC	Cyclic redundancy checksum generated by the originating LAN station or far end device does not match the checksum calculated from data received. On a LAN, this often indicates noise or transmission problems on the LAN interface or the LAN bus. A high number of CRCs is usually the result of collisions or a station transmitting bad data. On a serial link, CRCs often indicate noise, gain hits or other transmission problems on the data link.
frame	Number of packets received incorrectly having a CRC error and a noninteger number of octets. On a serial line, this is usually the result of noise or other transmission problems.
overrun	Number of times the serial receiver hardware was unable to hand received data to a hardware buffer because the input rate exceeded the receiver's ability to handle the data.
ignored	Number of received packets ignored by the virtual access interface because the interface hardware ran low on internal buffers. These buffers are different than the system buffers mentioned previously in the buffer description. Broadcast storms and bursts of noise can cause the ignored count to be incremented.
abort	Illegal sequence of one bits on a virtual access interface. This usually indicates a clocking problem between the virtual access interface and the data link equipment.
packets output	Total number of messages transmitted by the system.
bytes	Total number of bytes, including data and MAC encapsulation, transmitted by the system.
underruns	Number of times that the far-end transmitter has been running faster than the near-end communication server's receiver can handle. This might never be reported on some virtual access interfaces.
output errors	Sum of all errors that prevented the final transmission of datagrams out of the virtual access interface being examined. Note that this might not balance with the sum of the enumerated output errors, as some datagrams might have more than one error, and others might have errors that do not fall into any of the tabulated categories.
collisions	Number of packets colliding.
interface resets	Number of times a virtual access interface has been completely reset. This can happen if packets queued for transmission were not sent within several seconds. This can be caused by a malfunctioning modem that is not supplying the transmit clock signal, or by a cable problem. If the system notices that the carrier detect line of a virtual access interface is up, but the line protocol is down, it periodically resets the interface in an effort to restart it. Interface resets can also occur when a virtual access interface is looped back or shut down.
restarts	Number of times the controller was restarted because of errors.
carrier transitions	Number of times the carrier detect (CD) signal of a virtual access interface has changed state. Indicates modem or line problems if the CD line changes state often. If data carrier detect (DCD) goes down and comes up, the carrier transition counter increments two times.
output buffer failures	Number of outgoing packets dropped from the output buffer.
output buffers swapped out	Number of times the output buffer was swapped out.

## show translate

To view translation sessions that have been configured, use the **show translate** global configuration command:

```
show translate
```

### Syntax Description

This command has no arguments or keywords.

### Command Mode

Privileged EXEC

### Usage Guidelines

This command first appeared in Cisco IOS Release 10.3.

The display from this command shows each translation session set up on the router. It shows the incoming device and virtual terminal protocol as well as the outgoing device and protocol.

### Sample Display

The **show translate** output in this sample display is based on the following translation command configured:

```
translate x25 3131415912345 ppp ip-pool scope-name cardinal keepalive 0
```

If the previous **translate** command is enabled, the following output is created by the **show translation** command:

```
router# show translate

Translate From: x25 3131415912345
           To:   PPP ip-pool scope-name cardinal keepalive 0
           1/1 users active, 1 peak, 1 total, 0 failures
```

Table 3 describes fields shown in the display.

**Table 3 Show Translate Field Descriptions**

Field	Description
Translate From: x25 3131415912345	Protocol (X.25) and address (3131415912345) of the incoming device.
To: PPP	The virtual terminal protocol (PPP).
ip-pool	Obtain an IP address from a DHCP proxy client or a local pool.
scope-name cardinal	Specific local scope name (cardinal) from which to obtain an IP address.
keepalive 0	Indicates that keepalive updates have been disabled for the current translation session.
1/1 users active	Number of users active over the total number of users.
1 peak	Maximum number of translate sessions up at any given time.
1 total	Total number of translation sessions.
0 failures	Number of failed translation attempts resulting from this configuration.

The **show translate** output in this sample display is based on the following translation command configured:

```
translate x25 31301234 PPP 192.168.14.23 ipx-client Loopback0
```

If the previous **translate** command is enabled, the following output is created by the **show translation** command:

```
router# show translate

Translate From: x25 31301234
           To:   PPP 192.168.14.23 ipx-client Loopback0
           1/1 users active, 1 peak, 1 total, 0 failures
```

Table 4 describes fields shown in the display.

**Table 4 Show Translate Field Descriptions**

Field	Description
Translate From: x25 31301234	Protocol (X.25) and address (31301234) of the incoming device.
To: PPP 192.168.14.23	The virtual terminal protocol (PPP) and IP address of the outgoing device.
ipx-client loopback0	Indicates that loopback interface 0 has been configured in client mode.
1/1 users active	Number of users active over the total number of users.
1 peak	Maximum number of translate sessions up at any given time.
1 total	Total number of translation sessions.
0 failures	Number of failed translation attempts resulting from this configuration.

## show users (virtual access interfaces)

To display information about the active lines on the router, use the **show users** user EXEC command.

**show users** [**all**]

### Syntax Description

**all** (Optional) Specifies that all lines be displayed, regardless of whether anyone is using them.

### Command Mode

User EXEC

### Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

This command displays the line number, connection name, idle time, hosts (including virtual access interfaces) and terminal location.

### Sample Display

The following is sample output from the **show users** command. You can use it to identify an active virtual access interface:

```
router> show users

Line          User          Host(s)          Idle    Location
*  0 con 0          idle            01:58
  10 vty 0          Virtual-Access2 0        1212321
```

The asterisk (\*) indicates the current terminal session.

Table 5 describes significant fields shown in the displays.

**Table 5 Show Users Field Descriptions**

Field	Description
Line	Contains three subfields. <ul style="list-style-type: none"><li>The first subfield (0, 10, and Vi2 in the sample output) is the absolute line number.</li><li>The second subfield (con and vty) indicates the type of line. Possible values follow:<ul style="list-style-type: none"><li>con—Console</li><li>aux—Auxiliary port</li><li>TTY—Asynchronous terminal port</li><li>VTY—Virtual terminal</li></ul></li><li>The third subfield (0 in the sample output) indicates the relative line number within the type.</li></ul>
User	User connected to the line. If no user is listed in this field, no one is using the line.

**Table 5 Show Users Field Descriptions (Continued)**

<b>Field</b>	<b>Description</b>
Host(s)	Host to which the user is connected (outgoing connection). A value of <code>idle</code> means that there is no outgoing connection to a host. The value of <code>Virtual-Access2</code> in the example refers to virtual access interface number 2. The value of <code>Virtual PPP (PT)</code> is the virtual access interface referred to by the previous line.
Idle	Interval (in minutes) since the user has entered something.
Location	Either the hard-wired location for the line or, if there is an incoming connection, the host from which incoming connection originated. In the example, 1212321 refers to the X.121 address of an X.25 host.

## translate lat

When receiving a LAT connection request to a service name, the Cisco router can automatically translate the request to another outgoing protocol connection type. To set this up, use the **translate** global configuration command.

```
translate lat incoming-service-name [in-option] protocol outgoing-address [out-options]
[global-options]
```

### Syntax Description

<i>incoming-service-name</i>	A LAT service name. When used on the incoming portion, <i>service-name</i> is the name of the service that users specify when trying to make a translated connection. This name can match the name of final destination resource, but this is not required. This can be useful when making remote translated connections.
<i>in-option</i>	(Optional) Incoming connection request option: <ul style="list-style-type: none"> <li>• <b>unadvertised</b>—Prevents service advertisements from being broadcast to the network. This can be useful, for example, when you define translations for many printers, and you do not want these services advertised to other LAT terminal servers. (VMS systems will be able to connect to the service even though it is not advertised.)</li> </ul>
<i>protocol outgoing-address</i>	A protocol name followed by an IP address or host name. The host name is translated to an IP address during configuration, unless you use the tcp <b>host-name</b> option, which allows load balancing by dynamically resolving an IP address from a host name. These arguments can have the following values: <ul style="list-style-type: none"> <li>• <b>x25 X.121-address</b>—X.25 and an X.121 address. The X.121 address must conform to specifications provided in the <i>CCITT 1984 Red Book</i>. This number generally consists of a portion that is administered by the PDN and a portion that is locally assigned. You must be sure that the numbers that you assign agree with the addresses assigned to you by the X.25 service provider. The X.121 addresses will generally be subaddresses of the X.121 address for the X.25 network interface. Typically, the interface address will be a 12-digit number. Any additional digits are interpreted as a subaddress. The PDN still routes these calls to the interface, and the Cisco IOS software itself is responsible for dealing with the extra digits appropriately.</li> <li>• <b>tcp ip-address</b>—TCP/IP Telnet and a standard IP address or host name. The argument <i>ip-address</i> is a standard, four-part dotted decimal IP address or the name of an IP host that can be resolved by the Domain Name System (DNS) or explicit specification in an <b>ip host</b> command.</li> <li>• <b>slip ip-address</b>—The argument <i>ip-address</i> is a standard, four-part dotted decimal IP address or the name of an IP host that can be resolved by the Domain Name System (DNS).</li> </ul>

- **ppp ip-address**—The argument *ip-address* is a standard, four-part dotted decimal IP address or the name of an IP host that can be resolved by the Domain Name System (DNS).
- **autocommand**—Enables you to specify a string for an outgoing connection. The string executes upon connection to a host. If you want to enable ARA on an outgoing connection, you need to specify **autocommand arap**.

The **autocommand** option is necessary for ARA, because ARA does not use addressing, and **autocommand** permits you to invoke the **arap** string.

If the string following **autocommand** has one or more spaces as part of the string, you must place quotation marks (“ ”) around the string. For example, if you specify **autocommand tn3270 abracadabra**, you must enclose **tn3270 abracadabra** in quotes.

The **autocommand** option applies only to outgoing connections.

You can issue any EXEC command and any switch or host name as an argument to the **autocommand** option.

#### *out-options*

(Optional) Incoming and outgoing connection request options. These arguments can have the following values:

X.25 translation options:

- **cud c-u-data**—Sends the specified Call User Data (CUD) text (*c-u-data*) as part of an outgoing call request after the protocol identification bytes.
- **no-reverse**—Specifies that outgoing calls are not to use reverse charging, when the interface default is that all outgoing calls are reverse charged.
- **profile profile**—Sets the X.3 PAD parameters as defined in the profile created by the **x29 profile** command.
- **reverse**—Provides reverse charging for X.25 on a per-call rather than a per-interface basis. Requests reverse charges on a specified X.121 address, even if the serial interface is not configured to request reverse charge calls. This is an outgoing option only.

Telnet TCP translation option:

- **port number**—For incoming connections, number of the port to match. The default is port 23 (any port). For outgoing connections, number of the port to use. The default is port 23 (Telnet).

SLIP and PPP translation options:

- **ip-pool**—Obtain an IP address from a DHCP proxy client or a local pool. If the **scope-name** option is not specified, the address is obtained from a DHCP proxy client. If the **scope-name** option is specified, the IP address is obtained from the specified local pool.
- **scope-name**—Specific local scope name from which to obtain an IP address. Can specify a range of IP addresses.
- **header-compression [passive]**—Implements header compression on IP packets only. The option **passive** for SLIP connections permits compression on outgoing packets only if incoming TCP packets on the same virtual asynchronous interface are compressed. The default (without the **passive** option) permits compression on all traffic.
- **routing**—Permits routing updates between connections. This option is required if the destination device is not on a subnet connected to one of the interfaces on the router.
- **mtu bytes**—Permits you to change the maximum transmission unit (MTU) of packets that the virtual asynchronous interface supports. The default MTU is 1500 bytes on a virtual asynchronous interface. The acceptable range is 64 through 1,000,000 bytes.

More PPP translation options:

- **keepalive number-of-seconds**—Permits you to specify the interval at which keepalive packets are sent on SLIP and PPP virtual asynchronous interfaces. By default, keepalive packets are enabled and are sent every 10 seconds. To shut off keepalive packets, use a value of 0. The active keepalive interval is 1 through 32767 seconds. When you do not change from the default of 10, the keepalive interval does not appear in **show running-config** or **show translate** output.
- **authentication {chap | pap}**—Use CHAP or PAP authentication for PPP on virtual asynchronous interfaces. If you specify both options, order is significant; the system will try to use the first authentication type, then the second.
- **ppp use-tacacs**—Enables TACACS authentication for CHAP or PAP on virtual asynchronous interfaces (for PPP only; TACACS authentication is not supported for SLIP).

- **ipx loopback** *number*—Permits clients running IPX–PPP over X.25 to connect through virtual terminal (VTY) lines on the router. The **loopback** *number* option specifies the loopback interface to be created. A loopback interface must have been created and configured with a Novell IPX network number before IPX–PPP can work on the VTY line. The VTY line is assigned to the loopback interface.

*global-options*

(Optional) Translation options that can be used by any connection type. It can be one or more of the following:

- **access-class** *number*—Allows the incoming call to be used by source hosts that match the access list parameters. The argument *number* is the number (integer) previously assigned to an access list. The standard access list is 1-99.
- **max-users** *number*—Limits the number of simultaneous users of the translation to *number* (an integer you specify).
- **local**—Allows Telnet protocol negotiations to *not* be translated.
- **rotor**—Provides a basic load sharing of the IP destinations.
- **login**—Requires that the user log in before the outgoing connection is made. This type of login is specified on the VTY lines with the **login** command.
- **quiet**—Suppresses printing of user-information messages.

## Default

No default translation parameters

## Command Mode

Global configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

Table 6 provides a visual aid for the **translate lat** command. You define protocol translation connections by supplying a protocol keyword and the address, hostname, or service name, followed by optional features. Global options apply to most connection types, but there are exceptions.

**Table 6 Translate LAT Command Options**

	Incoming Address	Options	Outgoing Protocol	Options	Global Options
<b>translate lat</b>	<i>incoming-address</i>	[ <i>in-options</i> ]	<i>protocol outgoing-address</i>	[ <i>out-options</i> ]	[ <i>global-options</i> ]
<b>lat</b>	<i>service-name</i>	<b>unadvertised</b>	<b>x25</b> <i>x.121-address</i>	<b>cul</b> <i>c-u-data</i> <b>no-reverse</b>	<b>access-class</b> <i>number</i> <b>max-users</b> <i>number</i>

**Table 6 Translate LAT Command Options (Continued)**

Incoming Address	Options	Outgoing Protocol	Options	Global Options
			<b>profile</b> <i>profile</i>	<b>local</b>
			<b>reverse</b>	<b>login</b>
		<b>tcp</b> <i>ip-address</i>	<b>port</b> <i>number</i>	<b>quiet</b>
			<b>host-name</b> <i>name</i>	
			<b>multibyte-IAC</b>	
		<b>slip</b> <i>ip-address</i>	<b>ip-pool</b> [ <i>scope-name name</i> ]	
			<b>headercompression</b> [ <i>passive</i> ]	
			<b>routing</b>	
			<b>keepalive</b> <i>number-of-seconds</i>	
			<b>mtu</b> <i>bytes</i>	
		<b>ppp</b> <i>ip-address</i>	<b>ip-pool</b> [ <i>scope-name name</i> ]	
			<b>headercompression</b> [ <i>passive</i> ]	
			<b>routing</b>	
			<b>keepalive</b> <i>number-of-seconds</i>	
			<b>mtu</b> <i>bytes</i>	
			<b>authentication</b> { <b>pap</b>   <b>chap</b> }	
			<b>ppp use-tacacs</b>	
			<b>ipx loopback</b> <i>number</i>	
		<b>autocommand</b> [ <b>arap</b>   <i>exec-string</i> ]		

**Examples**

The following example illustrates incoming LAT to outgoing TCP translations. The **unadvertised** keyword prevents broadcast of service advertisements to other servers. Outgoing translated packets are transmitted to IP host rubble, TCP port 4005.

```
translate lat pt-printer1 unadvertised tcp rubble port 4005
           incoming      option      outgoing  option
```

The following example translates LAT on an incoming line to SLIP on an outgoing line. It uses header compression only if incoming TCP packets on the same interface are compressed.

```
translate lat rudolph slip 10.0.0.4 header-compression
           incoming  outgoing  option
```

The following example first shows the command to disable keepalive packets on a PPP line, then shows sample output from the **show translate** command when keepalive packets have been turned off on the line.

```
translate lat ramble ppp 172.21.2.2 keepalive 0
.
.
router# show translate

Translate From: LAT ramble
To:    PPP 172.21.2.2 keepalive 0
      0/0 users active, 0 peak, 0 total, 0 failures
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

- show translate**
- translate tcp**
- translate x25**
- x29 access-list**
- x29 profile**

## translate lat (virtual access interfaces)

When receiving a LAT connection request to a service name, the Cisco router can automatically translate the request to another outgoing protocol connection type. To set this up, use the **translate lat** global configuration command.

The command syntax that follows shows how to apply a virtual interface template in place of outgoing **translate** options. If you are using virtual templates for protocol translation, all outgoing options are defined in the virtual interface template. Table 7 lists all outgoing options and their corresponding interface configuration commands.

**translate lat** *incoming-service-name* [**unadvertised**] **virtual-template** *number* [*global-options*]

## Syntax Description

<i>incoming-service-name</i>	A LAT service name. When used on the incoming portion of the <b>translate lat</b> command, <i>service-name</i> is the name of the service that users specify when trying to make a translated connection. This name can match the name of the final destination resource, but this match is not required. Such matches can be useful when making remote translated connections.
<b>unadvertised</b>	(Optional) The only incoming connection request option for LAT—Prevents service advertisements from being broadcast to the network. This can be useful, for example, when you define translations for many printers, and you do not want these services advertised to other LAT terminal servers. (VMS systems will be able to connect to the service even though it is not advertised.)
<b>virtual-template</b> <i>number</i>	Applies the virtual interface template specified by <i>number</i> in place of outgoing options.
<i>global-options</i>	(Optional) Translation options that can be used by any connection type. It can be one or more of the following: <ul style="list-style-type: none"> <li>• <b>access-class</b> <i>number</i>—Allows the incoming call to be used by source hosts that match the access list parameters. The argument <i>number</i> is the number (integer) previously assigned to an access list. The standard access list is 1 to 99.</li> <li>• <b>max-users</b> <i>number</i>—Limits the number of simultaneous users of the translation to <i>number</i> (an integer you specify).</li> <li>• <b>local</b>—Allows Telnet protocol negotiations to <i>not</i> be translated.</li> <li>• <b>rotor</b>—Provides a basic load sharing of the IP destinations.</li> <li>• <b>login</b>—Requires that the user log in before the outgoing connection is made. This type of login is specified on the VTY lines with the <b>login</b> command.</li> <li>• <b>quiet</b>—Suppresses printing of user-information messages.</li> </ul>

## Default

No default translation parameters

## Command Mode

Global configuration

## Usage Guidelines

This command first appeared before Cisco IOS Release 10.0.

You define the protocol translation connections by choosing a protocol keyword and supplying the appropriate address, host name, or service name. The protocol connection information is followed by optional features for that connection, as appropriate. For example, the **binary** option is only appropriate with TCP/IP connections. The global options, in general, apply to all the connection types, but there are exceptions.

Rather than specifying outgoing translation options in the **translate** command, configure these options as interface configuration commands under the virtual interface template, then apply the virtual interface template to the **translate** command. Table 10 maps outgoing **translate** command options to interface commands you can configure in the virtual interface template.

**Table 7 Mapping Outgoing Translate Command Options to Interface Commands**

Translate Command Options	Corresponding Interface Configuration Command
<b>ip-pool</b>	<b>peer default ip address</b> { ip-address   dhcp   pool [poolname]}
<b>header-compression</b>	<b>ip tcp header compression</b> [on   off   passive]
<b>routing</b>	<b>ip routing</b> or <b>ipx routing</b>
<b>mtu</b>	<b>mtu</b>
<b>keepalive</b>	<b>keepalive</b>
<b>authentication</b> { chap   pap }	<b>ppp authentication</b> { chap   pap }
<b>ppp use-tacacs</b>	<b>ppp use-tacacs</b>
<b>ipx loopback</b>	<b>ipx ppp-client loopback</b> number

### Example

The following example configures PPP tunneling from a PC across a LAT network. The remote PC is given the IP address 10.12.118.12 when it dials in. The **unadvertised** keyword prevents broadcast of service advertisements to other servers.

```
interface Virtual-Template1
 ip unnumbered Ethernet0
 peer default ip address 10.12.118.12
 ppp authentication chap
 !
 translate lat pt-printer1 unadvertised virtual-template 1
           incoming          option          outgoing
```

### Related Commands

You can use the master indexes or search online to find documentation of related commands.

- show translate**
- translate tcp**
- translate x25**
- x29 access-list**
- x29 profile**

## translate tcp

When receiving a TCP connection request to a particular destination address or host name, the Cisco router can automatically translate the request to another outgoing protocol connection type. To set this up, use the **translate** global configuration command.

```
translate tcp incoming-address [in-options] protocol outgoing-address [out-options]
[global-options]
```

### Syntax Description

<i>incoming-address</i>	TCP/IP Telnet and a standard IP address or host name. The argument <i>ip-address</i> is a standard, four-part dotted decimal IP address or the name of an IP host that can be resolved by the Domain Name System (DNS) or explicit specification in an <b>ip host</b> command.
<i>in-options</i>	<p>(Optional) Incoming connection request options. These arguments can have the following values:</p> <p>Telnet TCP translation options:</p> <ul style="list-style-type: none"> <li>• <b>binary</b>—Negotiates Telnet binary mode on the Telnet connection. (This was the default in previous versions of the protocol translation software and is set automatically when you enter at <b>translate</b> command in the old format.)</li> <li>• <b>port number</b>—For incoming connections, number of the port to match. The default is port 23 (any port). For outgoing connections, number of the port to use. The default is port 23 (Telnet).</li> <li>• <b>printer</b>—Supports LAT and X.25 printing over a TCP network among multiple sites. Causes the protocol translation software to delay the completion of an incoming Telnet connection until after the outgoing protocol connection (to LAT or X.25) has been successfully established. An unsuccessful outgoing connection attempt results in the TCP connection to the router being refused, rather than being accepted and then closed, which is the default behavior. Note that using this option will force the global option <i>quiet</i> to be applied to the translation.</li> <li>• <b>stream</b>—Performs stream processing, which enables a raw TCP stream with no Telnet control sequences. A stream connection does not process or generate any Telnet options, and prevents Telnet processing of the data stream as well. This option might be useful for connections to ports running UUCP or other non-Telnet protocols, or to ports connected to printers. For ports connected to printers using Telnet, the stream option prevents some of usual problems associated with using Telnet for printers, such as strange things happening to bare carriage returns or line feeds and echoing of data back to VMS systems.</li> </ul>

*protocol outgoing-address*

Name of a protocol followed by a service name, IP address, or host name. The host name is translated to an IP address during configuration. These arguments can have the following values:

- **lat** *service-name*—LAT and a LAT service name. You must learn the service name, through LAT service advertisements, before you can use it.
- **x25** *X.121-address*—X.25 and an X.121 address. The X.121 address must conform to specifications provided in the *CCITT 1984 Red Book*. This number generally consists of a portion that is administered by the PDN and a portion that is locally assigned. You must be sure that the numbers that you assign agree with the addresses assigned to you by the X.25 service provider. The X.121 addresses will generally be subaddresses of the X.121 address for the X.25 network interface.
- **slip** *ip-address*—The argument *ip-address* is a standard, four-part dotted decimal IP address or the name of an IP host that can be resolved by the Domain Name System (DNS). The **slip** argument applies only to outgoing connections; SLIP is not supported on incoming protocol translation connections.
- **ppp** *ip-address*—The argument *ip-address* is a standard, four-part dotted decimal IP address or the name of an IP host that can be resolved by the Domain Name System (DNS). The **ppp** argument applies only to outgoing connections; PPP is not supported for incoming protocol translation connections.
- **autocommand**—Enables you to specify a string for an outgoing connection. The string executes upon connection to a host. If you want to enable ARA on an outgoing connection, you need to specify **autocommand arap**.

The **autocommand** option is necessary for ARA, because ARA does not use addressing, and **autocommand** permits you to invoke the **arap** string.

If the string following **autocommand** has one or more spaces as part of the string, you must place quotation marks (“ ”) around the string. For example, if you specify **autocommand tn3270 abracadabra**, you must enclose **tn3270 abracadabra** in quotes.

The **autocommand** option applies only to outgoing connections.

You can issue any EXEC command and any switch or host name as an argument to the **autocommand** option.

*out-options*

(Optional) Outgoing connection request options. These arguments can have the following values:

LAT translation options:

- **node** *node-name*—Connects to the specified node (*node-name*) that offers a service. By default, the connection is made to the highest-rated node that offers the service.
- **port** *port-name*—Destination LAT port name (*port-name*) in the format of the remote system. This parameter is usually ignored in most timesharing systems, but is used by terminal servers that offer reverse-LAT services.

X.25 translation options:

- **cud** *c-u-data*—Sends the specified Call User Data (CUD) text (*c-u-data*) as part of an outgoing call request after the protocol identification bytes.
- **no-reverse**—Specifies that outgoing calls are not to use reverse charging, when the interface default is that all outgoing calls are reverse charged.
- **profile** *profile*—Sets the X.3 PAD parameters as defined in the profile created by the **x29 profile** command.
- **reverse**—Provides reverse charging for X.25 on a per-call rather than a per-interface basis. Requests reverse charges on a specified X.121 address, even if the serial interface is not configured to request reverse charge calls. This is an outgoing option only.

SLIP and PPP translation options:

- **ip-pool**—Obtain an IP address from a DHCP proxy client or a local pool. If the **scope-name** option is not specified, the address is obtained from a DHCP proxy client. If the **scope-name** option is specified, the IP address is obtained from the specified local pool.
- **scope-name**—Specific local scope name from which to obtain an IP address. Can specify a range of IP addresses.
- **header-compression** [**passive**]—Implements header compression on IP packets only. The option **passive** for SLIP connections permits compression on outgoing packets only if incoming TCP packets on the same virtual asynchronous interface are compressed. The default (without the **passive** option) permits compression on all traffic.
- **routing**—Permits routing updates between connections. This option is required if the destination device is not on a subnet connected to one of the interfaces on the router.

- **mtu bytes**—Permits you to change the maximum transmission unit (MTU) of packets that the virtual asynchronous interface supports. The default MTU is 1500 bytes on a virtual asynchronous interface. The acceptable range is 64 through 1,000,000 bytes.

More PPP translation options:

- **keepalive number-of-seconds**—Permits you to specify the interval at which keepalive packets are sent on SLIP and PPP virtual asynchronous interfaces. By default, keepalive packets are enabled and are sent every 10 seconds. To shut off keepalive packets, use a value of 0. The active keepalive interval is 1 through 32767 seconds. When you do not change from the default of 10, the keepalive interval does not appear in **show running-config** or **show translate** output.
- **authentication {chap | pap}**—Use CHAP or PAP authentication for PPP on virtual asynchronous interfaces. If you specify both options, order is significant; the system will try to use the first authentication type, then the second.
- **ppp use-tacacs**—Enables TACACS authentication for CHAP or PAP on virtual asynchronous interfaces (for PPP only; TACACS authentication is not supported for SLIP).
- **ipx loopback number**—Permits clients running IPX–PPP over X.25 to connect through virtual terminal (VTY) lines on the router. The **loopback number** option specifies the loopback interface to be created. A loopback interface must have been created and configured with a Novell IPX network number before IPX–PPP can work on the VTY line. The VTY line is assigned to the loopback interface.

#### *global-options*

(Optional) Translation options that can be used by any connection type. It can be one or more of the following:

- **access-class number**—Allows the incoming call to be used by source hosts that match the access list parameters. The argument *number* is the number (integer) previously assigned to an access list. The standard access list is 1-99.
- **max-users number**—Limits the number of simultaneous users of the translation to *number* (an integer you specify).
- **local**—Allows Telnet protocol negotiations to *not* be translated.
- **rotor**—Provides a basic load sharing of the IP destinations.
- **login**—Requires that the user log in before the outgoing connection is made. This type of login is specified on the VTY lines with the **login** command.
- **quiet**—Suppresses printing of user-information messages.

## Default

No default translation parameters

## Command Mode

Global configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 11.2.

Table 8 provides a visual aid for understanding how to use the **translate tcp** command. As the table illustrates, you define the protocol translation connections by choosing a protocol keyword and supplying the appropriate address, host name, or service name. The protocol connection information is followed by optional features for that connection, as appropriate. For example, the **binary** option is only appropriate with TCP/IP connections. The global options, in general, apply to all the connection types, but there are exceptions.

**Table 8 Translate TCP Command Options**

Incoming Protocol	Options	Outgoing Protocol	Options	Global Options
<b>translate</b> <i>protocol incoming-address</i>	[ <i>in-options</i> ]	<i>protocol outgoing-address</i>	[ <i>out-options</i> ]	[ <i>global-options</i> ]
<b>tcp</b> <i>ip-address   host_name</i>	<b>port</b> <i>number</i>	<b>lat</b> <i>service-name</i>	<b>node</b> <i>node-name</i>	<b>access-class</b> <i>number</i>
	<b>binary</b>		<b>port</b> <i>port-name</i>	<b>max-users</b> <i>number</i>
	<b>stream</b>	<b>x25</b> <i>x.121-address</i>	<b>cud</b> <i>c-u-data</i>	<b>local</b>
	<b>printer</b>		<b>no-reverse</b>	<b>login</b>
			<b>profile</b> <i>profile</i>	<b>quiet</b>
			<b>reverse</b>	
		<b>slip</b> <i>ip-address</i>	<b>ip-pool</b> [ <i>scope-name name</i> ]	
			<b>headercompression</b> [ <i>passive</i> ]	
			<b>routing</b>	
			<b>keepalive</b> <i>number-of-seconds</i>	
			<b>mtu</b> <i>bytes</i>	
		<b>ppp</b> <i>ip-address</i>	<b>ip-pool</b>	
			<b>headercompression</b> [ <i>passive</i> ]	
			<b>routing</b>	
			<b>keepalive</b> <i>number-of-seconds</i>	
			<b>mtu</b> <i>bytes</i>	
			<b>authentication</b> { <b>pap</b>   <b>chap</b> }	
			<b>ppp use-tacacs</b>	

**Table 8 Translate TCP Command Options (Continued)**

Incoming Protocol	Options	Outgoing Protocol	Options	Global Options
			<i>ipx loopback number</i>	
		<b>autocommand</b> [ <i>arap   exec-string</i> ]		

**Examples**

The following example illustrates the use of the TCP incoming protocol option **printer** for an incoming TCP connection:

```
translate tcp 172.19.32.250 printer x25 5678
           incoming           option outgoing
```

The following example permits clients running IPX/PPP to connect through the device's VTY lines to a server running PPP:

```
interface loopback0
  no ip address
  ipx network 544
  ipx sap-interval 2000
!
translate tcp 172.21.14.67 port 1234 ppp 10.0.0.2 ipx loopback0
!           incoming                               outgoing option
```

**Related Commands**

You can use the master indexes or search online to find documentation of related commands.

- show translate**
- translate lat**
- translate x25**
- x29 access-list**
- x29 profile**

## translate tcp (virtual access interfaces)

When receiving a TCP connection request to a particular destination address or host name, the Cisco router can automatically translate the request to another outgoing protocol connection type. To set this up, use the **translate tcp** global configuration command.

The command syntax that follows shows how to apply a virtual interface template in place of outgoing **translate** options. If you are using virtual templates for protocol translation, all outgoing options are defined in the virtual interface template.

**translate tcp** *incoming-address* [*in-options*] **virtual-template** *number* [*global-options*]

### Syntax Description

<i>incoming-address</i>	TCP/IP Telnet and a standard IP address or host name. The argument <i>ip-address</i> is a standard, four-part dotted decimal IP address or the name of an IP host that can be resolved by the Domain Name System (DNS) or explicit specification in an <b>ip host</b> command.
<i>in-options</i>	<p>(Optional) Incoming connection request options. These arguments can have the following values:</p> <ul style="list-style-type: none"> <li>• <b>binary</b>—Negotiates Telnet binary mode on the Telnet connection. (This was the default in previous versions of the Cisco IOS software and is set automatically when you enter at <b>translate</b> command in the old format.)</li> <li>• <b>port number</b>—For incoming connections, enter the number of the port to match. The default is port 23 (any port). For outgoing connections, enter the number of the port to use. The default is port 23 (Telnet).</li> <li>• <b>printer</b>—Supports LAT and X.25 printing over a TCP network among multiple sites. This option causes the Cisco IOS software to delay the completion of an incoming Telnet connection until after the outgoing protocol connection (to LAT or X.25) has been successfully established. An unsuccessful outgoing connection attempt results in the TCP connection to the router being refused, rather than being accepted and then closed, which is the default behavior. Note that using this option will force the global option <b>quiet</b> to be applied to the translation.</li> <li>• <b>stream</b>—Performs stream processing, which enables a raw TCP stream with no Telnet control sequences. A stream connection does not process or generate any Telnet options, and prevents Telnet processing of the data stream as well. This option might be useful for connections to ports running UUCP or other non-Telnet protocols, or to ports connected to printers. For ports connected to printers using Telnet, the stream option prevents some of the usual problems associated with using Telnet for printers, such as strange things happening to bare carriage returns or line feeds and echoing of data back to VMS systems.</li> </ul>

- virtual-template** *number* Applies the virtual interface template specified by *number* in place of outgoing options.
- global-options* (Optional) Translation options that can be used by any connection type. It can be one or more of the following:
- **access-class** *number*—Allows the incoming call to be used by source hosts that match the access list parameters. The argument *number* is an integer value previously assigned to an access list. The standard access list range is from 1 to 99.
  - **local**—Allows Telnet protocol negotiations to *not* be translated.
  - **login**—Requires that the user log in before the outgoing connection is made. This type of login is specified on the VTY lines with the **login** command.
  - **max-users** *number*—Maximum number of simultaneous users of the translation.
  - **quiet**—Suppresses printing of user-information messages.
  - **rotor**—Provides a basic load sharing of the IP destinations.

#### Default

No default translation parameters

#### Command Mode

Global configuration

#### Usage Guidelines

This command first appeared before Cisco IOS Release 10.0.

You define the protocol translation connections by choosing a protocol keyword and supplying the appropriate address, host name, or service name. The protocol connection information is followed by optional features for that connection, as appropriate. For example, the **binary** option is only appropriate with TCP/IP connections. The global options, in general, apply to all the connection types, but there are exceptions.

#### Example

The following example illustrates the use of the TCP incoming option **printer** for an incoming TCP connection:

```
interface Virtual-Template1
 ip unnumbered Ethernet0
 peer default ip address 10.12.108.1
 ppp authentication chap

translate tcp 172.19.32.250 printer Virtual-Template1
                    incoming          option outgoing
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**show translate**  
**translate lat**  
**translate x25**  
**x29 access-list**  
**x29 profile**

## translate x25

When receiving an X.25 connection request to a particular destination address, the Cisco router can automatically translate the request to another outgoing protocol connection type. To set this up, use the **translate** global configuration command.

**translate x25** *incoming-address* [*in-options*] *protocol* *outgoing-address* [*out-options*]  
[*global-options*]

## Syntax Description

*incoming-address*

X.25 and an X.121 address. The X.121 address must conform to specifications provided in the *CCITT 1984 Red Book*. This number generally consists of a portion that is administered by the PDN and a portion that is locally assigned. You must be sure that the numbers that you assign agree with the addresses assigned to you by the X.25 service provider. The X.121 addresses will generally be subaddresses of the X.121 address for the X.25 network interface. Typically, the interface address will be a 12-digit number. Any additional digits are interpreted as a subaddress. The PDN still routes these calls to the interface, and the Cisco IOS software itself is responsible for dealing with the extra digits appropriately. Do not use the same address on the interface and for translation.

*in-options*

(Optional) Incoming connection request options. These arguments can have the following values:

- **accept-reverse**—Accepts reverse charged calls on an X.121 address even if the serial interface is not configured to accept reverse charged calls. This is an incoming option only.
- **cud** *c-u-data*—Sends the specified Call User Data (CUD) text (*c-u-data*) as part of an outgoing call request after the protocol identification bytes.
- **idle** *minutes*—Specifies the number of minutes the VC is idle. This option enables the protocol translation function to clear a switched virtual circuit (SVC) after a set period of inactivity, where *minutes* is the number of minutes in the period. Calls either originated or terminated are cleared. The maximum value of *minutes* is 255. The default value of *minutes* is zero.
- **printer**—Supports LAT and TCP printing over an X.25 network among multiple sites. Provides an “interlock mechanism” between the acceptance of an incoming X.25 connection and the opening of an outgoing LAT or TCP connection. The option causes the Cisco IOS software to delay the call confirmation of an incoming X.25 call request until the outgoing protocol connection (to TCP or LAT) has been successfully established. An unsuccessful outgoing connection attempt to the router results in the incoming X.25 connection being refused, rather than being confirmed and then cleared, which is the default behavior. Note that using this option will force the global option **quiet** to be applied to the translation.
- **profile** *profile*—Sets the X.3 PAD parameters as defined in the profile created by the **x29 profile** command.

*protocol outgoing-address* Name of a protocol followed by a service name, IP address, or host name. The host name is translated to an IP address during configuration, unless you use the TCP **host-name** option, which allows load balancing by dynamically resolving an IP address from a host name. These arguments can have the following values:

- **lat** *service-name*—LAT and a LAT service name. You must learn the service name, through LAT service advertisements, before you can use it.
- **tcp** *ip-address*—TCP/IP Telnet and a standard IP address or host name. The argument *ip-address* is a standard, four-part dotted decimal IP address or the name of an IP host that can be resolved by the Domain Name System (DNS) or explicit specification in an **ip host** command.
- **slip** *ip-address*—The argument *ip-address* is a standard, four-part dotted decimal IP address or the name of an IP host that can be resolved by the Domain Name System (DNS). The **slip** argument applies only to outgoing connections; SLIP is not supported on incoming protocol translation connections.
- **ppp** *ip-address*—The argument *ip-address* is a standard, four-part dotted decimal IP address or the name of an IP host that can be resolved by the Domain Name System (DNS). The **ppp** argument applies only to outgoing connections; PPP is not supported for incoming protocol translation connections.
- **autocommand**—Enables you to specify a string for an outgoing connection. The string executes upon connection to a host. If you want to enable ARA on an outgoing connection, you need to specify the **autocommand arap** string.

The **autocommand** option is necessary for ARA, because ARA does not use addressing, and **autocommand** permits you to invoke the **arap** string.

If the string following **autocommand** has one or more spaces as part of the string, you must place quotation marks (“ ”) around the string. For example, if you specify **autocommand tn3270 abracadabra**, you must enclose the **tn3270 abracadabra** string in quotes.

The **autocommand** option applies only to outgoing connections.

You can issue any EXEC command and any switch or host name as an argument to the **autocommand** option.

*out-options* (Optional) Outgoing connection request options. These arguments can have the following values:

- **use-map**—Applies **x25 map pad** command entry options (such as CUD and idle) and facilities (such as packet in, packet out, win in, and win out) to the outgoing protocol translation call. This application occurs when the protocol translation function searches the X.25 map PAD entries and finds a matching X.121 destination address. The X.25 map facilities applied to the outgoing translation can be viewed with the **show translation** command throughout the duration of the translation session.

LAT translation options:

- **node** *node-name*—Connects to the specified node (*node-name*) that offers a service. By default, the connection is made to the highest-rated node that offers the service.
- **port** *port-name*—Destination LAT port name (*port-name*) in the format of the remote system. This parameter is usually ignored in most timesharing systems but is used by terminal servers that offer reverse-LAT services.

Telnet TCP translation options:

- **port** *number*—For incoming connections, number of the port to match. The default is port 23 (any port). For outgoing connections, number of the port to use. The default is port 23 (Telnet).

SLIP and PPP translation options:

- **ip-pool**—Obtain an IP address from a DHCP proxy client or a local pool. If the **scope-name** option is not specified, the address is obtained from a DHCP proxy client. If the **scope-name** option is specified, the IP address is obtained from the specified local pool.
- **scope-name**—Specific local scope name from which to obtain an IP address. This option can specify a range of IP addresses.
- **header-compression** [**passive**]—Implements header compression on IP packets only. The option **passive** for SLIP connections permits compression on outgoing packets only if incoming TCP packets on the same virtual asynchronous interface are compressed. The default (without the **passive** option) permits compression on all traffic.
- **routing**—Permits routing updates between connections. This option is required if the destination device is not on a subnet connected to one of the interfaces on the router.
- **mtu** *bytes*—Permits you to change the maximum transmission unit (MTU) of packets that the virtual asynchronous interface supports. The default MTU is 1500 bytes on a virtual asynchronous interface. The acceptable range is 64 to 1,000,000 bytes.

PPP translation options:

- **keepalive** *number-of-seconds*—Permits you to specify the interval at which keepalive packets are sent on SLIP and PPP virtual asynchronous interfaces. By default, keepalive packets are enabled and are sent every 10 seconds. To shut off keepalive packets, use a value of 0. The active keepalive interval is 1 to 32767 seconds. When you do not change from the default of 10, the keepalive interval does not appear in the **show running-config** or **show translate** command output.
- **authentication {chap | pap}**—Use CHAP or PAP authentication for PPP on virtual asynchronous interfaces. If you specify both options, order is significant; the system will try to use the first authentication type, then the second.
- **ppp use-tacacs**—Enables TACACS authentication for CHAP or PAP on virtual asynchronous interfaces (for PPP only; TACACS authentication is not supported for SLIP).
- **ipx loopback number**—Specifies the loopback interface to be created and permits clients running IPX-PPP over X.25 to connect through virtual terminal (VTY) lines on the router. A loopback interface must have been created and configured with a Novell IPX network number before IPX-PPP can work on the VTY line. The VTY line is assigned to the loopback interface.

*global-options*

(Optional) Translation options that can be used by any connection type. It can be one or more of the following:

- **access-class number**—Allows the incoming call to be used by source hosts that match the access list parameters. The argument *number* is the number (integer) previously assigned to an access list. The standard access list is 1 to 99.
- **max-users number**—Limits the number of simultaneous users of the translation to *number* (an integer you specify).
- **local**—Prevents Telnet protocol negotiations to from being translated.
- **login**—Requires that the user log in before the outgoing connection is made. This type of login is specified on the VTY lines with the **login** command.
- **rotor**—Provides a basic load sharing of the IP destinations.
- **quiet**—Suppresses printing of user-information messages.
- **swap**—Allows X.3 parameters to be set on the router by the host originating the X.25 call or by an X.29 profile. This configuration enables incoming and outgoing X.25 connections to be swapped so that the device is treated like a PAD when it accepts a call. By default, the router functions like a PAD for calls that it initiates, and like an X.25 host for calls it accepts. The **swap** keyword allows connections from an X.25 host that wants to connect to the router, and then treats it like a PAD. For X.25-to-TCP translations only.

- **pvc number** {[**interface serial number**] [**packetsize in-size out-size**] [**window size in-size out-size**]}—Specifies that the incoming or outgoing connection is actually a permanent virtual circuit (PVC). Only one session is allowed per PVC, where:

*number*—Specifies the virtual-circuit channel number of the incoming connection, which must be less than the virtual circuits assigned to the switched virtual circuits (SVC).

**interface serial number**—Specifies a PVC interface on which to set up the PVC connection.

**packetsize in-size out-size**—Specifies the input packet size (*in-size*) and output packet size (*out-size*) for the PVC. Following are valid packet size values:

16, 32, 64, 128, 256, 512 1024, 2048, or 4096

**window size in-size out-size**—Specifies the packet count for input windows (*in-size*) and output windows (*out-size*) for the outgoing translation. Values of *in-size* and *out-size* range 1 to 127 and must not be greater than the value set for the **x25 modulo** command. You must specify the same value for *in-size* and *out-size*.

## Default

No default translation parameters

## Command Mode

Global configuration

## Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

Table 9 provides a visual aid for understanding how the command syntax flows for the **translate x25** command. In the table, follow the steps horizontally (from left to right). As you travel from step to step, you can choose from a vertical list of options at each step. As the table illustrates, you define the protocol translation connections by choosing a protocol keyword and supplying the appropriate address or service name. The protocol connection information is followed by optional features for that connection, as appropriate. The global options, in general, apply to all the connection types, but there are exceptions. The **swap** keyword, for example, is for X.25 to TCP translations only. See the examples for more explanations on how to enter this command.

To use virtual templates with incoming X.25 translation, see the **translate x.25 (virtual access interfaces)** command.

**Table 9 Sequence of Steps for Translating a Protocol**

	<b>Incoming Address Step 1</b>	<b>Incoming Options Step 2</b>	<b>Protocol Outgoing Address Step 3</b>	<b>Outgoing Options Step 4</b>	<b>Global Options Step 5</b>
<b>translate x25</b>	<i>x.21 address</i>	<b>idle</b> <i>minutes</i>	<b>lat</b> <i>service-name</i>	<b>use-map</b>	<b>access-class</b> <i>number</i>
		<b>cu</b> <i>c-u-data</i>	<b>tcp</b> <i>ip-address</i>	<b>node</b> <i>node-name</i>	<b>max-users</b> <i>number</i>
		<b>profile</b> <i>profile</i>	<b>slip</b> <i>ip-address</i>	<b>port</b> <i>port-name</i>	<b>local</b>
		<b>accept-reverse</b>	<b>ppp</b> <i>ip-address</i>	<b>port</b> <i>number</i>	<b>login</b>
		<b>printer</b>	<b>autocommand</b> [ <b>arap</b>   <i>exec-string</i> ]	<b>host-name</b> <i>name</i>	<b>quiet</b>
		<i>idle minutes</i>		<b>multibyte-iac</b>	<b>swap</b>
				<b>ip-pool</b> [ <i>scope-name name</i> ]	<b>pvc</b> [ <i>number</i>   <b>interface</b> <i>serial-number</i> ] <b>packetsize</b> <i>in-size out-size</i> <b>window</b> <i>size in-size</i> <i>out-size</i>
				<b>headercompression</b> [ <b>passive</b> ]	
				<b>routing</b>	
				<b>keepalive</b> <i>number-of-seconds</i>	
				<b>mtu</b> <i>bytes</i>	
				<b>ip-pool</b>	
				<b>headercompression</b> [ <b>passive</b> ]	
				<b>routing</b>	
				<b>keepalive</b> <i>number-of-seconds</i>	
				<b>mtu</b> <i>bytes</i>	
				<b>authentication</b> { <b>pap</b>   <b>chap</b> }	
				<b>ppp use-tacacs</b>	
				<b>ipx loopback</b> <i>number</i>	

## Examples

The following example illustrates a simple X.25-to-TCP **translate x25** command. Packets coming in X.25 address 652365123 arrive via PVC 1 and are translated to TCP packets and transmitted out IP address 172.16.1.1.

```
translate x25 652365123 pvc 1 tcp 172.16.1.1
!          incoming      option outgoing
```

The following example illustrates a more complex configuration that calls an X.29 profile and swaps the default PAD operation of the router to that of an X.25 host. The name of the profile is *fullpackets*.

```
x29 profile fullpackets 2:0 3:0 4:100 7:21
translate x25 217536124 profile fullpackets tcp rubble port 4006 swap
!          incoming      option      outgoing option  global
```

The following example illustrates the use of the X.25 incoming protocol option **printer** for an incoming X.25 connection:

```
translate x25 55555 printer tcp 172.16.1.1
!          incoming      option outgoing
```

The following example translates X.25 packets to PPP. It enables routing updates between the two connections:

```
translate x25 12345678 ppp 10.0.0.2 routing
!          incoming      outgoing  option
```

The following example permits clients running ARA to connect through the devices' VTY lines to an AppleTalk network:

```
appletalk routing
translate x25 12345678 autocommand arap
!          incoming      outgoing
  arap enable
  arap dedicated
  arap timelimit 45
  arap warningtime 5
  arap noguest
  arap require-manual-password
  arap net-access-list 614
```

The following example specifies IP pooling from a DHCP server named *ludicrous*. It then specifies that incoming TCP traffic be translated to SLIP. The DHCP server will dynamically assign IP addresses on the outgoing sessions.

```
ip address-pool dhcp-proxy-client
ip dhcp-server ludicrous
translate x25 5467835 ppp ip-pool scope-name ludicrous
```

The following example specifies a local IP pool named *scandal* with IP addresses ranging from 172.18.10.10 to 172.18.10.110. It then specifies that incoming X.25 traffic be translated to PPP. The local IP pool *scandal* will be used to dynamically assign IP addresses on the outgoing sessions.

```
ip-pool scandal 172.18.10.10 172.18.10.110
translate x25 1234567 ppp ip-pool scope-name scandal
```

X.25 calls are cleared if they are idle for the configured time, as shown in the following example:

```
translate x25 1234 idle 2 lat shazam
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**show translate**  
**translate lat**  
**translate tcp**  
**x29 access-list**  
**x29 profile**

## translate x25 (virtual access interfaces)

When receiving a X.25 connection request to a particular destination address, the Cisco router can automatically translate the request to another outgoing protocol connection type. To set up this feature, use the **translate x25** global configuration command.

The command syntax that follows shows how to apply a virtual interface template in place of outgoing **translate x25** options. If you are using virtual templates for protocol translation, all outgoing options are defined in the virtual interface template. Table 10 lists all outgoing options and their corresponding interface configuration commands.

**translate x25** *incoming-address* [*in-options*] **virtual-template** *number* [*global-options*]

### Syntax Description

*incoming-address*

X.25 and an X.121 address. The X.121 address must conform to specifications provided in the *CCITT 1984 Red Book*. This number generally consists of a portion that is administered by the PDN and a portion that is locally assigned. You must be sure that the numbers that you assign agree with the addresses assigned to you by the X.25 service provider. The X.121 addresses will generally be subaddresses of the X.121 address for the X.25 network interface. Typically, the interface address will be a 12-digit number. Any additional digits are interpreted as a subaddress. The PDN still routes these calls to the interface, and the Cisco IOS software is responsible for appropriately dealing with the extra digits. Do not use the same address on the interface and for translation.

*in-options*

(Optional) Incoming connection request options. These arguments can have the following values:

- **accept-reverse**—Accepts reverse charged calls on an X.121 address even if the serial interface is not configured to accept reverse charged calls. This is an incoming option only.
- **cud** *c-u-data*—Sends the specified Call User Data (CUD) text (*c-u-data*) as part of an outgoing call request after the protocol identification bytes.
- **printer**—Supports LAT and TCP printing over an X.25 network among multiple sites. Provides an “interlock mechanism” between the acceptance of an incoming X.25 connection and the opening of an outgoing LAT or TCP connection. The option causes the Cisco IOS software to delay the call confirmation of an incoming X.25 call request until the outgoing protocol connection (to TCP or LAT) has been successfully established. An unsuccessful outgoing connection attempt to the router results in the incoming X.25 connection being refused, rather than being confirmed and then cleared, which is the default behavior. Note that using this option will force the global option **quiet** to be applied to the translation.

- **profile** *profile*—Sets the X.3 PAD parameters as defined in the profile created by the **x29 profile** command.
- **pvc** *number*—Specifies that the incoming connection (identified by the argument *number*) is actually a permanent virtual circuit (PVC).

**virtual-template** *number*

Apply the virtual interface template specified by *number* in place of outgoing options.

*global-options*

(Optional) Translation options that can be used by any connection type. It can be one or more of the following:

- **access-class** *number*—Allows the incoming call to be used by source hosts that match the access list parameters. The argument *number* is an integer in the range 1 to 99 that was previously assigned to an access list.
- **max-users** *number*—Limits the number of simultaneous users of the translation to *number* (an integer you specify).
- **local**—Allows Telnet protocol negotiations to *not* be translated.
- **login**—Requires that the user log in before the outgoing connection is made. This type of login is specified on the VTY lines with the **login** command.
- **rotor**—Provides a basic load sharing of the IP destinations.
- **quiet**—Suppresses printing of user-information messages.
- **swap**—Allows X.3 parameters to be set on the router by the host originating the X.25 call, or by an X.29 profile. This allows incoming and outgoing X.25 connections to be swapped so that the device is treated like a PAD when it accepts a call. By default, the router functions like a PAD for calls that it initiates, and like an X.25 host for calls it accepts. The **swap** keyword allows connections from an X.25 host that wants to connect to the router, and then treats it like a PAD. For X.25-to-TCP translations only.

Default

No default translation parameters

Command Mode

Global configuration

## Usage Guidelines

This command first appeared before Cisco IOS Release 10.0.

You define the protocol translation connections by choosing a protocol keyword and supplying the appropriate address or service name. The protocol connection information is followed by optional features for that connection, as appropriate. The global options, in general, apply to all the connection types, but there are exceptions. The **swap** keyword, for example, is for X.25 to TCP translations only. See the example for more explanations on how to enter this command.

Rather than specifying outgoing translation options in the **translate** command, configure these options as interface configuration commands under the virtual interface template, then apply the virtual interface template to the **translate** command. Table 10 maps outgoing **translate** command options to interface commands you can configure in the virtual interface template.

**Table 10 Mapping Outgoing Translate Command Options to Interface Commands**

Translate Command Options	Corresponding Interface Configuration Command
<b>ip-pool</b>	<b>peer default ip address {ip-address   dhcp   pool [poolname]}</b>
<b>header-compression</b>	<b>ip tcp header compression [on   off   passive]</b>
<b>routing</b>	<b>ip routing</b> or <b>ipx routing</b>
<b>mtu</b>	<b>mtu</b>
<b>keepalive</b>	<b>keepalive</b>
<b>authentication {chap   pap}</b>	<b>ppp authentication {chap   pap}</b>
<b>ppp use-tacacs</b>	<b>ppp use-tacacs</b>
<b>ipx loopback</b>	<b>ipx ppp-client loopback number</b>

## Example

The following example shows a virtual template with PPP encapsulation specified by default (not explicit). It also specifies CHAP authentication and an X.29 access list.

```
x29 access-list 1 permit ^5555
!
interface Virtual-Template1
 ip unnumbered Ethernet0
 peer default ip address 172.16.2.129
 ppp authentication chap
!
translate x25 5555667 virtual-template 1 access-class 1
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**interface virtual-template**  
**show translate**  
**translate lat**  
**translate tcp**  
**x29 access-list**  
**x29 profile**

## vty-async

To configure all virtual terminal lines on a router to support asynchronous protocol features, use the **vty-async** global configuration command. Use the **no** form of this command to disable asynchronous protocol features on virtual terminal lines.

```
vty-async  
no vty-async
```

### Syntax Description

This command has no arguments or keywords.

### Default

Asynchronous protocol features are not enabled by default on virtual terminal lines.

### Command Mode

Global configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 10.3.

The **vty-async** command extends asynchronous protocol features from physical asynchronous interfaces to virtual terminal lines. Normally, SLIP and PPP can function only on asynchronous interfaces, not on virtual terminal lines. However, extending asynchronous functionality to virtual terminal lines permits you to run SLIP and PPP on these *virtual asynchronous interfaces*. One practical benefit is the ability to tunnel SLIP and PPP over X.25 PAD, thus extending remote node capability into the X.25 area. You can also tunnel SLIP and PPP over Telnet or LAT on virtual terminal lines. To tunnel SLIP and PPP over X.25, LAT, or Telnet, you use the protocol translation feature in the Cisco IOS software.

To tunnel SLIP or PPP inside X.25, LAT, or Telnet, you can use two-step protocol translation or one-step protocol translation, as follows:

- If you are tunnelling SLIP or PPP using the two-step method, you need to first enter the **vty-async** command. Next, you perform two-step translation.
- If you are tunnelling SLIP or PPP using the one-step method, you do not need to enter the **vty-async** command. You only need to issue the **translate** command with the SLIP or PPP keywords, because the **translate** command automatically enables asynchronous protocol features on virtual terminal lines.

### Example

The following example enables asynchronous protocol features on virtual terminal lines:

```
vty-async
```

## Related Commands

You can use the master indexes or search online to find documentation of related commands.

**ppp**  
**slip**  
**translate**

## vty-async dynamic-routing

To enable dynamic routing on all virtual asynchronous interfaces, use the **vty-async dynamic-routing** global configuration command. Use the **no** form of this command to disable asynchronous protocol features on virtual terminal lines and, therefore, disable routing on virtual terminal lines.

```
vty-async dynamic-routing  
no vty-async
```

### Syntax Description

This command has no arguments or keywords.

### Default

Dynamic routing is not enabled on virtual asynchronous interfaces.

### Command Mode

Global configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 10.3.

This feature enables IP routing on virtual asynchronous interfaces. When you issue this command and a user later makes a connection to another host using SLIP or PPP, the user must specify **/routing** on the SLIP or PPP command line.

If you had not previously entered the **vty-async** command, the **vty-async dynamic-routing** command creates virtual asynchronous interfaces, then enables dynamic routing on them.

### Example

The following example enables dynamic routing on virtual asynchronous interfaces:

```
vty-async dynamic-routing
```

### Related Commands

You can use the master indexes or search online to find documentation of related commands.

**async dynamic routing**

## vty-async header-compression

To compress the headers of all TCP packets on virtual asynchronous interfaces, use the **vty-async header-compression** global configuration command. Use the **no** form of this command to disable virtual asynchronous interfaces and header compression.

```
vty-async header-compression [passive]  
no vty-async
```

### Syntax Description

**passive** (Optional) Specifies that outgoing packets to be compressed only if TCP incoming packets on the same virtual asynchronous interface are compressed. For SLIP, if you do not specify this option, the Cisco IOS software will compress all traffic. The default is no compression. For PPP, the Cisco IOS software always negotiates header compression.

### Default

Header compression is not enabled on virtual asynchronous interfaces.

### Command Mode

Global Configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 10.3.

This feature compresses the headers on TCP/IP packets on virtual asynchronous connections to reduce the size of the packets and to increase performance. This feature only compresses the TCP header, so it has no effect on UDP packets or other protocol headers. The TCP header compression technique, described fully in RFC 1144, is supported on virtual asynchronous interfaces using SLIP or PPP encapsulation. You must enable compression on both ends of a connection.

### Example

The following example compresses outgoing TCP packets on virtual asynchronous interfaces only if incoming TCP packets are compressed:

```
vty-async header-compression passive
```

### Related Commands

You can use the master indexes or search online to find documentation of related commands.

**async dynamic routing**

## vty-async ipx ppp-client loopback

To enable IPX–PPP on virtual terminal (VTY) lines, use the **vty-async ipx ppp-client loopback** global configuration command. Use the **no** form of this command to disable IPX–PPP sessions on VTYs.

```
vty-async ipx ppp-client loopback number  
no vty-async
```

### Syntax Description

<i>number</i>	Number of the loopback interface configured for IPX to which the VTY lines are assigned.
---------------	--

### Default

IPX over PPP is not enabled on VTY lines.

### Command Mode

Global configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 11.0.

This command enables users to log into the router from a device running a virtual terminal protocol, then issue the PPP command at the EXEC prompt to connect to a remote device.

A loopback interface must already have been defined and an IPX network number must have been assigned to the loopback interface before the **vty-async ipx ppp-client loopback** command will permit IPX–PPP on VTY lines.

### Example

The following example enables IPX over PPP on VTY lines:

```
ipx routing ramana  
interface loopback0  
  ipx network 12345  
vty-async ipx ppp-client loopback0
```

### Related Commands

You can use the master indexes or search online to find documentation of related commands.

```
interface loopback  
ipx network
```

## vty-async keepalive

To change the frequency of keepalive packets on all virtual asynchronous interfaces, use the **vty-async keepalive** global configuration command. Use the **no vty-async** command to disable asynchronous protocol features on virtual terminal lines, or the **vty-async keepalive 0** command to disable keepalive packets on virtual terminal lines.

```
vty-async keepalive seconds  
no vty-async  
vty-async keepalive 0
```

### Syntax Description

<i>seconds</i>	The frequency, in seconds, with which the Cisco IOS software sends keepalive messages to the other end of a virtual asynchronous interface. To disable keepalive packets, use a value of 0. The active keepalive interval is 1 to 32767 seconds. The default is 10 seconds.
----------------	---

### Default

10 seconds

### Command Mode

Global configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 10.3.

Use this command to change the frequency of keepalive updates on virtual asynchronous interfaces from the default of 10, or to disable keepalive updates. If you do not change from the default of 10, the keepalive interval does not appear in **show running-config** or **show translate** output.

A connection is declared down after three update intervals have passed without receiving a keepalive packet.

### Examples

In the following example, the keepalive interval is set to 30 seconds.

```
vty-async keepalive 30
```

In the following example, the keepalive interval is set to 0 (off), and the sample output for **show running-config** is shown.

```
vty-async keepalive 0  
  
router# show running-config  
no vty-async keepalive
```

### Related Commands

You can use the master indexes or search online to find documentation of related commands.

**keepalive**

## vty-async mtu

To set the maximum transmission unit (MTU) size on virtual asynchronous interfaces, use the **vty-async mtu** global configuration command. Use the **no** form of this command to disable asynchronous protocol features on virtual terminal lines.

**vty-async mtu** *bytes*  
**no vty-async**

### Syntax Description

*bytes* MTU size of IP packets that the virtual asynchronous interface can support. The default MTU is 1500 bytes, the minimum MTU is 64 bytes, and the maximum is 1,000,000 bytes.

### Default

1500 bytes

### Command Mode

Global configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 10.3.

Use this command to modify the maximum transmission unit (MTU) for packets on a virtual asynchronous interfaces. You might want to change to a smaller MTU size for IP packets transmitted on a virtual terminal line configured for asynchronous functions for any of the following reasons:

- The SLIP or PPP application at the other end only supports packets up to a certain size.
- You want to ensure a shorter delay by using smaller packets.
- The host echoing takes longer than 0.2 seconds.

Do not change the MTU size unless the SLIP or PPP implementation running on the host at the other end of the virtual asynchronous interface supports reassembly of IP fragments. Because each fragment occupies a spot in the output queue, it might also be necessary to increase the size of the SLIP or PPP hold queue if your MTU size is such that you might have a high amount of packet fragments in the output queue.

### Example

The following example sets the MTU for IP packets to 256 bytes:

```
vty-async mtu 256
```

### Related Commands

You can use the master indexes or search online to find documentation of related commands.

**mtu**

## vty-async ppp authentication

To enable PPP authentication on virtual asynchronous interfaces, use the **vty-async ppp authentication {chap | pap}** global configuration command. Use the **no** form of this command to disable PPP authentication.

```
vty-async ppp authentication {chap | pap}  
no vty-async ppp authentication {chap | pap}
```

### Syntax Description

<b>chap</b>	Enable CHAP on all virtual asynchronous interfaces.
<b>pap</b>	Enable PAP on all virtual asynchronous interfaces.

### Default

No CHAP or PAP authentication for PPP.

### Command Mode

Global configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 10.3.

This command configures the virtual asynchronous interface to either authenticate CHAP or PAP while running PPP. After you have enabled CHAP or PAP, the local router requires a password from remote devices. If the remote device does not support CHAP or PAP, no traffic will be passed to that device.

### Example

The following example enables CHAP authentication for PPP sessions on virtual asynchronous interfaces:

```
vty-async ppp authentication chap
```

### Related Commands

You can use the master indexes or search online to find documentation of related commands.

```
ppp authentication chap  
ppp authentication pap  
ppp use-tacacs  
vty-async  
vty-async ppp use-tacacs
```

## vty-async ppp use-tacacs

To enable TACACS authentication for PPP on virtual asynchronous interfaces, use the **vty-async ppp use-tacacs** global configuration command. Use the **no** form of this command to disable TACACS authentication on virtual asynchronous interfaces.

```
vty-async ppp use-tacacs  
no vty-async ppp use-tacacs
```

### Syntax Description

This command has no arguments or keywords.

### Default

TACACS for PPP is disabled.

### Command Mode

Global configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 10.3.

This command requires the extended TACACS server.

After you have enabled TACACS, the local router requires a password from remote devices.

This feature is useful when integrating TACACS with other authentication systems that require a clear-text version of a user's password. Such systems include one-time password systems and token card systems.

If the username and password are contained in the CHAP password, then the CHAP secret is not used by the router. Because most PPP clients require that a secret be specified, you can use any arbitrary string; the Cisco IOS software ignores it.

You cannot enable TACACS authentication for SLIP on asynchronous or virtual asynchronous interfaces.

### Example

The example enables TACACS authentication for PPP sessions:

```
vty-async ppp use-tacacs
```

### Related Commands

You can use the master indexes or search online to find documentation of related commands.

```
ppp use-tacacs  
vty-async ppp authentication
```

## vty-async virtual-template

To configure virtual terminal (VTY) lines to support asynchronous protocol functions based on the definition of a virtual interface template, use the **vty-async virtual-template** global configuration command. Use the **no** form of this command to disable virtual interface templates for asynchronous functions on virtual terminal lines.

```
vty-async virtual-template number  
no vty-async
```

### Syntax Description

*number*            The virtual interface number.

### Default

Asynchronous protocol features are not enabled by default on virtual terminal lines.

### Command Mode

Global configuration

### Usage Guidelines

The **vty-async** command first appeared in Cisco IOS Release 10.3. The **vty-async virtual-template** command first appeared in Cisco IOS Release 11.3.

The **vty-async virtual-template** command enables you to support tunneling of SLIP or PPP across X.25, TCP, or LAT networks by using two-step protocol translation.

Before issuing the **vty-async virtual-template** command, create and configure a virtual interface template by using the **interface virtual-template** command. Configure this virtual interface as a regular asynchronous serial interface. That is, assign the virtual interface template the IP address of the Ethernet interface, and configure addressing, just as on an asynchronous interface. You can also enter commands in interface configuration mode that compress TCP headers or configure CHAP authentication for PPP.

After creating a virtual interface template, apply it by issuing the **vty-async virtual-template** command. When a user dials in through a VTY line, the router creates a virtual access interface, which is a temporary interface that supports the asynchronous protocol configuration specified in the virtual interface template. This virtual access interface is created dynamically, and is freed up as soon as the connection drops.

Before virtual templates were implemented, you could use the **vty-async** command to extend asynchronous protocol functions from physical asynchronous interfaces to VTY lines. However, in doing so, you created a virtual asynchronous interface, rather than the virtual access interface. The difference is that the virtual asynchronous interfaces are allocated permanently, whereas the virtual access interfaces are created dynamically when a user calls in and closed down when the connection drops.

You can have up to 25 virtual templates interfaces, but you can apply only one template to vty-async interfaces on a router. There can be up to 300 virtual access interfaces on a router.

### Example

The following example enables asynchronous protocol features on virtual terminal lines:

```
vty-async
vty-async Virtual-Template 1
vty-async dynamic-routing
vty-async header-compression
!
interface Virtual-Template1
 ip unnumbered Ethernet0
 encapsulation ppp
 no peer default ip address
 ppp authentication chap
```

### Related Commands

You can use the master indexes or search online to find documentation of related commands.

**ppp**  
**slip**  
**translate lat**  
**translate tcp**  
**translate x25**  
**interface virtual-template**

## x25 host

Use the **x25 host** global configuration command to define a static host name-to-address mapping. Use the **no** form of this 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>	X.121 address.
<b>cu</b> d <i>call-user-data</i>	(Optional) Specifies the Call User Data (CUD) field in the X.25 Call Request packet.

### Default

No static address mapping is defined.

### Command Mode

Global configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

This command permits you to map an X.121 address to an easily recognizable name. You can later use this host name instead of the X.121 address when you issue the **translate** command for X.25.

### 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
```

The following example specifies static address mapping from the X.121 address 12345678 to the host name masala. It then uses the name masala in the **translate** command in place of the X.121 address when translating from the X.25 host to the PPP host with address 10.0.0.2.

```
x25 host masala 12345678  
translate x25 masala ppp 10.0.0.2 routing
```

### Related Commands

You can use the master indexes or search online to find documentation of related commands.

**translate**

## x29 access-list

To limit access to the access 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 {permit | deny} rx121-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 router.
<i>x121-address</i>	If applied as an inbound access class, specifies the X.121 address that can or cannot have access (with or without regular expression pattern-matching characters). The X.121 address is the source address of the incoming packet.  If applied as an outbound access class, then the address specifies a destination to where connections are allowed.

### Default

No access lists are defined.

### Command Mode

Global configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

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 “X.3 PAD Parameters” and “Regular Expressions” appendixes later in this publication 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 {default | name} parameter:value [parameter:value]
```

### Syntax Description

<b>default</b>	Specifies default profile script.
<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

The default PAD profile script is used. The default for inbound connections is:

```
2:0, 4:1, 15:0, 7:21
```

### Command Mode

Global configuration

### Usage Guidelines

This command first appeared in Cisco IOS Release 10.0.

When an X.25 connection is established, the router 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 access server accordingly.

For incoming PAD connections, the Protocol Translator uses a default PAD profile to set the remote X.3 PAD parameters unless a profile script is defined with the **translate** command.

### 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** global configuration command to effect use of this script.

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

To override the default PAD profile, create a PAD profile script named “default” by using the following command:

```
x29 profile default 2:1 4:1, 15:0, 4:0
```

### Related Commands

You can use the master indexes or search online to find documentation of related commands.

**translate**

