ADTRAN OPERATING SYSTEM (AOS)

Command Reference Guide

April 2003
61950860L1-35D

Trademarks

Any brand names and product names included in this manual are trademarks, registered trademarks, or trade names of their respective holders.

To the Holder of this Manual

The contents of this manual are current as of the date of publication. ADTRAN reserves the right to change the contents without prior notice.

In no event will ADTRAN be liable for any special, incidental, or consequential damages or for commercial losses even if ADTRAN has been advised thereof as a result of issue of this publication.

Notes provide additional useful information.

Cautions signify information that could prevent service interruption.

WARNING *Warnings provide information that could prevent damage to the equipment or endangerment to human life.*

901 Explorer Boulevard
P.O. Box 140000
Huntsville, AL 35814-4000
Phone: (256) 963-8000
www.adtran.com

Warranty and Customer Service

ADTRAN will replace or repair this product within five years from the date of shipment if it does not meet its published specifications or fails while in service. For detailed warranty, repair, and return information refer to the ADTRAN Equipment Warranty and Repair and Return Policy Procedure. Return Material Authorization (RMA) is required prior to returning equipment to ADTRAN. For service, RMA requests, or further information, contact one of the numbers listed at the end of this section.

Limited Product Warranty

ADTRAN warrants that for five (5) years from the date of shipment to Customer, all products manufactured by ADTRAN will be free from defects in materials and workmanship. ADTRAN also warrants that products will conform to the applicable specifications and drawings for such products, as contained in the Product Manual or in ADTRAN's internal specifications and drawings for such products (which may or may not be reflected in the Product Manual). This warranty only applies if Customer gives ADTRAN written notice of defects during the warranty period. Upon such notice, ADTRAN will, at its option, either repair or replace the defective item. If ADTRAN is unable, in a reasonable time, to repair or replace any equipment to a condition as warranted, Customer is entitled to a full refund of the purchase price upon return of the equipment to ADTRAN. This warranty applies only to the original purchaser and is not transferable without ADTRAN's express written permission. This warranty becomes null and void if Customer modifies or alters the equipment in any way, other than as specifically authorized by ADTRAN.

EXCEPT FOR THE LIMITED WARRANTY DESCRIBED ABOVE, THE FOREGOING CONSTITUTES THE SOLE AND EXCLUSIVE REMEDY OF THE CUSTOMER AND THE EXCLUSIVE LIABILITY OF ADTRAN AND IS IN LIEU OF ANY AND ALL OTHER WARRANTIES (EXPRESSED OR IMPLIED). ADTRAN SPECIFICALLY DISCLAIMS ALL OTHER WARRANTIES, INCLUDING (WITHOUT LIMITATION), ALL WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THIS EXCLUSION MAY NOT APPLY TO CUSTOMER.

In no event will ADTRAN or its suppliers be liable to Customer for any incidental, special, punitive, exemplary, or consequential damages experienced by either Customer or a third party (including, but not limited to, loss of data or information, loss of profits, or loss of use). ADTRAN is not liable for damages for any cause whatsoever (whether based in contract, tort, or otherwise) in excess of the amount paid for the item. Some states do not allow the limitation or exclusion of liability for incidental or consequential damages, so the above limitation or exclusion may not apply to Customer.

Customer Service, Product Support Information, and Training

ADTRAN will repair and return this product if within five years from the date of shipment the product does not meet its published specification or the product fails while in service. A return material authorization (RMA) is required prior to returning equipment to ADTRAN. For service, RMA requests, training, or more information, use the contact information which follows.

Repair and Return

If you determine that a repair is needed, please contact our Customer and Product Service (CAPS) department to have an RMA number issued. CAPS should also be contacted to obtain information regarding equipment currently in house or possible fees associated with repair:

CAPS Department (256) 963-8722

Identify the RMA number clearly on the package (below address), and return to the following address:

ADTRAN Customer and Product Service
901 Explorer Blvd. (East Tower)
Huntsville, Alabama 35806

RMA # _____

Pre-Sales Inquiries and Applications Support

Your reseller should serve as the first point of contact for support. If additional pre-sales support is needed, the ADTRAN Support web site provides a variety of support services such as a searchable knowledge base, latest product documentation, application briefs, case studies, and a link to submit a question to an Applications Engineer. All of this, and more, is available at:

http://support.adtran.com

When needed, further pre-sales assistance is available by calling our Applications Engineering Department:

Applications Engineering(800) 615-1176

Post-Sale Support

Your reseller should serve as the first point of contact for support. If additional support is needed, the ADTRAN Support web site provides a variety of support services such as a searchable knowledge base, updated firmware releases, latest product documentation, service request ticket generation, and trouble-shooting tools. All of this, and more, is available at:

http://support.adtran.com

When needed, further post-sales assistance is available by calling our Technical Support Center. Please have your unit serial number available when you call:

Technical Support (888) 4ADTRAN

Installation and Maintenance Support

The ADTRAN Custom Extended Services (ACES) program offers multiple types and levels of installation and maintenance services which allow you to choose the kind of assistance you need. This support is available at:

http://www.adtran.com/aces

For questions, call the ACES Help Desk:

ACES Help Desk	**(888) 874-ACES, ext. 2237**

Training

The Enterprise Network (EN) Technical Training Department offers training on our most popular products. These courses include overviews on product features and functions while covering applications of ADTRAN's product lines. ADTRAN provides a variety of training options, including customized training and courses taught at our facilities or at your site. For more information about training, please contact your Territory Manager or the Enterprise Training Coordinator:

Training Phone	**(800) 615-1176, ext. 7500**
Training Fax	**(256) 963-6700**
Training Email	**training@adtran.com**

ADTRAN OS COMMAND REFERENCE GUIDE

Contents

REFERENCE GUIDE INTRODUCTION

If you are new to the ADTRAN Operating System's Command Line Interface (CLI), take a few moments to review the information provided in the section which follows (*CLI Introduction*).

If you are already familiar with the CLI and you need information on a specific command or group of commands, proceed to *Command Descriptions* on page 13 of this guide.

CLI INTRODUCTION

This portion of the Command Reference Guide is designed to introduce you to the basic concepts and strategies associated with using the ADTRAN Operating System's Command Line Interface (CLI).

Accessing the CLI from your PC

All products using the ADTRAN OS are initially accessed by connecting a VT100 terminal (or terminal emulator) to the **CONSOLE** port located on the rear panel of the unit using a standard DB-9 (male) to DB-9 (female) serial cable. Configure the VT100 terminal or terminal emulation software to the following settings:

- 9600 baud
- 8 data bits
- No parity
- 1 stop bit
- No flow control

For more details on connecting to your unit, refer to the Quick Start Guide located on the ***NetVanta 3000 Series System Manual*** *CD provided with your unit.*

Understanding Command Security Levels

The ADTRAN CLI has two command security levels — **Basic** and **Enable**. Both levels support a specific set of commands. For example, all interface configuration commands are accessible only through the Enable security level. The following table contains a brief description of each level.

Table 1. Command Security Level Descriptions

Level	Access by...	Prompt	With this level you can...
Basic	beginning a session with your router.	`Router>`	• display system information • perform traceroute and ping functions • open a Telnet session
Enable	entering **`enable`** while in the Basic command security level as follows: `Router>`**`enable`**	`Router#`	• manage the startup and running configurations • use the debug commands • enter any of the configuration modes

To prevent unauthorized users from accessing the configuration functions of your router, immediately install an Enable-level password. Refer to the Quick Start Guide located on the ***NetVanta 3000 Series System Manual*** *CD provided with your unit for more information on configuring a password.*

Understanding Configuration Modes

The ADTRAN CLI has four configuration modes to organize the configuration commands – Global, Line, Router, and Interface. Each configuration mode supports a set of commands specific to the configurable parameters for the mode. For example, all frame relay configuration commands are accessible only through the Interface Configuration Mode (for the virtual frame relay interface). The following table contains a brief description of each level.

Table 2. Configuration Mode Descriptions

Mode	Access by...	Sample Prompt	With this mode you can...
Global	entering **`config`** while at the Enable command security level prompt. For example: `Router>enable` `Router#`**`config term`**	`Router(config)#`	• set the system's Enable-level password(s) • configure the system global IP parameters • configure the SNMP parameters • enter any of the other configuration modes

Line	specifying a line (console or Telnet) while at the Global Configuration Mode prompt.
	For example:
	`Router>enable`
	`Router#config term`
	`Router(config)#` **`line console 0`**
		`Router(config-con0)#`	• configure the console terminal settings (datarate, login password, etc.)
			• create Telnet logins and specify their parameters (login password, etc.)

Router	entering **`router rip`** **`router`** or **`router ospf`** while at the Global Configuration Mode prompt.
	For example:
	`Router>enable`
	`Router#config term`
	`Router(config)#` **`router rip`**
		`Router(config-rip)#`	• configure RIP or OSPF parameters
			• suppress route updates
			• redistribute information from outside routing sources (protocols)

Interface	specifying an interface (T1, Ethernet, frame relay, ppp, etc.) while in the Global Configuration Mode.
	For example:
	`Router>enable`
	`Router#config term`
	`Router(config)#`**`int eth 0/1`**
		`Router(config-eth 0/1)#`
		
		(The above prompt is for the Ethernet **LAN** interface located on the rear panel of the unit.)
			• configure parameters for the available LAN and WAN interfaces

Using CLI Shortcuts

The ADTRAN CLI provides several shortcuts which help you configure your router more easily. See the following table for descriptions.

Table 3. Command Line Shortcuts

Shortcut	Description
Up arrow key	To re-display a previously entered command, use the up arrow key. Continuing to press the up arrow key cycles through all commands entered starting with the most recent command.
Tab key	Pressing the <Tab> key after entering a partial (but unique) command will complete the command, display it on the command prompt line, and wait for further input.

?	The ADTRAN CLI contains help to guide you through the configuration process. Using the question mark, do any of the following:
	• Display a list of all subcommands in the current mode. For example:
	`Router(config-t1 1/1)#`**`coding ?`**
	`ami - Alternate Mark Inversion`
	`b8zs - Bipolar Eight Zero Substitution`
	• Display a list of available commands beginning with certain letter(s). For example:
	`Router(config)#`**`ip d?`**
	`default-gateway dhcp-server domain-lookup domain-name domain-proxy`
	• Obtain syntax help for a specific command by entering the command, a space, and then a question mark (?). The ADTRAN CLI displays the range of values and a brief description of the next parameter expected for that particular command. For example:
	`Router(config-eth 0/1)#`**`mtu ?`**
	`<64-1500> - MTU (bytes)`

<Ctrl> + A	Jump to the beginning of the displayed command line. This shortcut is helpful when using the **no** form of commands (when available). For example, pressing <Ctrl + A> at the following prompt will place the cursor directly after the #: `(config-eth 0/1)#` **`ip address 192.33.55.6`**
<Ctrl> + E	Jump to the end of the displayed command line. For example, pressing <Ctrl + E> at the following prompt will place the cursor directly after the **6**: `(config-eth 0/1)#` **`ip address 192.33.55.6`**
<Ctrl> + U	Clears the current displayed command line. The following provides an example of the <Ctrl + U> feature: `(config-eth 0/1)#` **`ip address 192.33.55.6`** (Press <Ctrl + U> here) `(config-eth 0/1)#`
auto finish	You need only enter enough letters to identify a command as unique. For example, entering **`int t1 1/1`** at the Global configuration prompt provides you access to the configuration parameters for the specified T1 interface. Entering **`interface t1 1/1`** would work as well, but is not necessary.

Performing Common CLI Functions

The following table contains descriptions of common CLI commands.

Table 4. Common CLI Functions

Command	Description

do	The **do** command provides a way to execute commands in other command sets without taking the time to exit the current command set and enter the desired one. The following example shows the **do** command used to view the frame relay interface configuration while currently in the T1 interface command set:
	
	For example:
	`(config)# interface t1 1/1`
	`(config-t1 1/1)# do show interfaces fr 7`

no	To undo an issued command or to disable a feature, enter **no** before the command. For example: `no shutdown t1 1/1`
`copy running-config startup-config`	When you are ready to save the changes made to the configuration, enter this command. This copies your changes to the unit's nonvolatile random access memory (NVRAM). Once the save is complete, the changes are retained even if the unit is shut down or suffers a power outage.
`show running config`	Displays the current configuration.
debug	Use the **debug** command to troubleshoot problems you may be experiencing on your network. These commands provide additional information to help you better interpret possible problems. For information on specific debug commands, refer to the section *Enable Mode Command Set* on page 24.
`undebug all`	To turn off any active debug commands, enter this command.

*The overhead associated with the **debug** command takes up a large portion of your router's resources and at times can halt other router processes. It is best to only use the **debug** command during times when the network resources are in low demand (non-peak hours, weekends, etc.).*

Understanding CLI Error Messages

The following table lists and defines some of the more common error messages given in the CLI.

Table 5. CLI Error Messages

Message	Helpful Hints
%Ambiguous command **%Unrecognized Command**	The command may not be valid in the current command mode, or you may not have entered enough correct characters for the command to be recognized. Try using the "?" command to determine your error. See Table 3 on page 10 for more information.
%Invalid or incomplete command	The command may not be valid in the current command mode, or you may not have entered all of the pertinent information required to make the command valid. Try using the "?" command to determine your error. See Table 3 on page 10 for more information.
%Invalid input detected at "^" marker	The error in command entry is located where the caret (^) mark appears. Enter a question mark at the prompt. The system will display a list of applicable commands or will give syntax information for the entry.

COMMAND DESCRIPTIONS

This portion of the guide provides a detailed listing of all available commands for the ADTRAN CLI (organized by command set). Each command listing contains pertinent information including the default value, the command security level required to access the command, a description of all sub-command parameters, functional notes for using the command, and a brief technology review.

To search for a particular command alphabetically, use the *Index on page 563*. To search for information on a group of commands within a particular command set, use the link references given below:

BASIC MODE COMMAND SET

To activate the Basic Mode command set, simply log in to the unit. After connecting the unit to a VT100 terminal (or terminal emulator) and activating a terminal session, the following prompt displays:

```
Router>
```

The following command is common to multiple command sets and is covered in a centralized section of this guide. For more information, refer to the section listed below:

exit on page 560

All other commands for this command set are described in this section in alphabetical order.

enable on page 15

logout on page 16

ping <*address*> on page 17

show clock on page 19

show snmp on page 20

show version on page 21

telnet <*address*> on page 22

traceroute <*address*> on page 23

enable

Use the **enable** command (at the Basic Command Mode prompt) to enter the Enable Command Mode. Use the **disable** command to exit the Enable Command Mode. See the section *Enable Mode Command Set* on page 24 for more information.

Syntax Description

No subcommands.

Default Values

No default value necessary for this command.

Command Modes

>	Basic Command Mode

Command History

Release 1.1	Command was introduced.

Functional Notes

The Enable Command Mode provides access to operating and configuration parameters and should be password protected to prevent unauthorized use. Use the **enable password** command (found in the Global Configuration command set) to specify an Enable Command Mode password. If the password is set, access to the Enable Commands (and all other "privileged" commands) is only granted when the correct password is entered. See *enable password [md5] <password>* on page 159 for more information.

Usage Examples

The following example enters the Enable Command Mode and defines an Enable Command Mode password:

> **enable**
configure terminal
(config)# **enable password ADTRAN**

At the next login, the following sequence must occur:

> **enable**
Password: ********
#

logout

Use the **logout** command to terminate the current session and return to the login screen.

Syntax Description

No subcommands.

Default Values

No defaults necessary for this command.

Command Modes

> or #	Basic or Enable Command Mode

Command History

Release 1.1	Command was introduced.

Usage Examples

The following example shows the logout command being executed in the Basic Mode:

Router> **logout**

Session now available

Press RETURN to get started.

ping <*address*>

Use the **ping** command (at the Basic Command Mode prompt) to verify IP network connectivity.

Syntax Description

<*address*> ***Optional*	Specifies the IP address of the system to ping. Entering the **ping** command with no specified address prompts the user with parameters for a more detailed **ping** configuration. See **Functional Notes** (below) for more information.

Default Values

No default value necessary for this command.

Command Modes

> or #	Basic or Enable Command Mode

Command History

Release 1.1	Command was introduced.

Functional Notes

The **ping** command helps diagnose basic IP network connectivity using the Packet InterNet Groper program to repeatedly bounce Internet Control Message Protocol (ICMP) Echo_Request packets off a system (using a specified IP address). The ADTRAN OS allows executing a standard **ping** request to a specified IP address or provides a set of prompts to configure a more specific **ping** configuration.

The following is a list of output messages from the **ping** command:

!	Success
-	Destination Host Unreachable
$	Invalid Host Address
X	TTL Expired in Transit
?	Unknown Host
*	Request Timed Out

The following is a list of available extended **ping** fields with descriptions:

Target IP address:	Specifies the IP address of the system to ping.
Repeat Count:	Number of ping packets to send to the system (valid range: 1 to 1000000).
Datagram Size:	Size (in bytes) of the ping packet (valid range: 1 to 1448).
Timeout in Seconds:	If a ping response is not received within the timeout period, the ping is considered unsuccessful (valid range: 1 to 5 seconds).
Extended Commands:	Specifies whether additional commands are desired for more ping configuration parameters.

Functional Notes (Continued)

Source Address (or interface):	Specifies the IP address to use as the source address in the ECHO_REQ packets.
Data Pattern:	Specify an alphanumerical string to use (the ASCII equivalent) as the data pattern in the ECHO_REQ packets.
Sweep Range of Sizes:	Varies the sizes of the ECHO_REQ packets transmitted.
Sweep Min Size:	Specifies the minimum size of the ECHO_REQ packet (valid range: 0 to 1448).
Sweep Max Size:	Specifies the maximum size of the ECHO_REQ packet (valid range: Sweep Min Size to 1448).
Sweep Interval:	Specifies the interval used to determine packet size when performing the sweep (valid range: 1 to 1448).
Verbose Output:	Specifies an extended results output.

Usage Examples

The following is an example of a successful **ping** command:

> **ping**
Target IP address:**192.168.0.30**
Repeat count[1-1000000]:**5**
Datagram Size [1-1000000]:**100**
Timeout in seconds [1-5]:**2**
Extended Commands? [y or n]:**n**
Type CTRL+C to abort.
Legend: '!' = Success '?' = Unknown host '$' = Invalid host address
'*' = Request timed out '-' = Destination host unreachable
'x' = TTL expired in transit

Pinging 192.168.0.30 with 100 bytes of data:
!!!!!
Success rate is 100 percent (5/5) round-trip min/avg/max = 19/20.8/25 ms

show clock

Use the **show clock** command to display the system time and date entered using the **clock set** command. See the section *clock set <time> <day> <month> <year>* on page 38 for more information.

Syntax Description

No subcommands.

Default Values

No default value necessary for this command.

Command Modes

> or #	Basic or Enable Command Mode

Command History

Release 1.1	Command was introduced.

Usage Examples

The following example displays the current time and data from the system clock:

>**show clock**

23:35:07 UTC Tue Aug 20 2002

show snmp

Use the **show snmp** command to display the system Simple Network Management Protocol (SNMP) parameters and current status of SNMP communications.

Syntax Description

No subcommands.

Default Values

No default value necessary for this command.

Command Modes

> or #	Basic or Enable Command Mode

Command History

Release 1.1	Command was introduced.

Usage Examples

The following is an example output using the **show snmp** command for a system with SNMP disabled and the default Chassis and Contact parameters:

> **show snmp**

```
Chassis: Chassis ID
Contact: Customer Service
SNMP logging is DISABLED
0 Rx SNMP packets
  0 Bad community names
  0 Bad community uses
  0 Bad versions
  0 Silent drops
  0 Proxy drops
  0 ASN parse errors
```


show version

Use the **show version** command to display the current ADTRAN OS version information.

Syntax Description

No subcommands.

Default Values

No default value necessary for this command.

Command Modes

> or #	Basic or Enable Command Mode

Command History

Release 1.1	Command was introduced.

Usage Examples

The following is a sample **show version** output:

>**show version**

```
ADTRAN OS version 03.00.18.adv
  Checksum: 1F0D5243 built on Fri Nov 08 13:12:06 2002
  Upgrade key: de76efcfeb4c8eeb6901188475dd0917
Boot ROM version 03.00.18
  Checksum: 7A3D built on: Fri Nov 08 13:12:25 2002
Copyright (c) 1999-2002 ADTRAN Inc.
Serial number C14C6308

UNIT_2 uptime is 0 days 4 hours 59 minutes 43 seconds

System returned to ROM by Warm Start
Current system image file is "030018adv.biz"
Boot system image file is "030018adv.biz"
```

telnet <*address*>

Use the **telnet** command to open a Telnet session (through the ADTRAN OS) to another system on the network.

Syntax Description

<*address*>	Specifies the IP address of the remote system.

Default Values

No default value necessary for this command.

Command Modes

> or #	Basic or Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following example opens a Telnet session with a remote system (**10.200.4.15**):

> **telnet 10.200.4.15**

User Access Login

Password:

traceroute <*address*>

Use the **traceroute** command to display the IP routes a packet takes to reach the specified destination.

Syntax Description

<*address*>	Specifies the IP address of the remote system to trace the routes to

Default Values

No default value necessary for this command.

Command Modes

> or #	Basic or Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following example performs a traceroute on the IP address 192.168.0.1:

```
# traceroute 192.168.0.1

Type CTRL+C to abort.
Tracing route to 192.168.0.1 over a maximum of 30 hops

  1   22ms   20ms   20ms    192.168.0.65
  2   23ms   20ms   20ms    192.168.0.1
#
```

ENABLE MODE COMMAND SET

To activate the Enable Mode command set, enter the **enable** command at the Basic Mode prompt. (If an enable password has been configured, a password prompt will display.) For example:

```
Router> enable
Password: XXXXXXX
Router#
```

The following command is common to multiple command sets and is covered in a centralized section of this guide. For more information, refer to the section listed below:

exit on page 560

All other commands for this command set are described in this section in alphabetical order.

clear access-list *<listname>*

Use the **clear access-list** command to clear all counters associated with all access lists (or a specified access list).

Syntax Description

<listname>	Specifies the name (label) of an access list
**Optional*	

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following example clears all counters for the access list labeled **MatchAll**:

> **enable**
#**clear access-list MatchAll**

clear arp-cache

Use the **clear arp-cache** command to remove all dynamic entries from the Address Resolution Protocol (ARP) cache table.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following example removes all dynamic entries from the ARP cache:

> **enable**
clear arp-cache

clear arp-entry *<address>*

Use the **clear arp-entry** command to remove a single entry from the Address Resolution Protocol (ARP) cache.

Syntax Description

<address>	Specifies the IP address of the entry to remove

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following example removes the entry for 10.200.4.56 from the ARP cache:

> **enable**
clear arp-entry 10.200.4.56

clear bridge <*group#*>

Use the **clear bridge** command to clear all counters associated with bridging (or for a specified bridge-group).

Syntax Description

<*group#*> **Optional*	Specifies a single bridge group (1-255).

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following example clears all counters for bridge group 17:

> **enable**
clear bridge 17

clear buffers max-used

Use the **clear buffers max-used** command to clear the maximum-used statistics for buffers displayed in the **show memory heap** command.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Usage Examples

> **enable**
clear buffers max-used

clear counters *<interface>*

Use the **clear counters** command to clear all interface counters (or the counters for a specified interface).

Syntax Description

<interface>	Specifies a single interface
**Optional*	

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following example clears all counters associated with the ethernet 0/1 interface:

> **enable**
clear counters ethernet 0/1

clear crypto ike sa *<policy priority>*

Use the **clear crypto ike sa** command to clear existing IKE security associations (SAs), including active ones.

Syntax Description

<policy priority> **Optional*	Clear out all existing IKE SAs associated with the designated policy priority. This number is assigned using the **crypto ike policy** command. See *crypto ike* on page 150 for more information.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 4.1	Command was introduced

Usage Examples

The following example clears the entire database of IKE SAs (including the active associations):

> **enable**
#**clear crypto ike sa**

clear crypto ipsec sa

Use the **clear crypto ipsec sa** command to clear existing IPSec security associations (SAs), including active ones.

Variations of this command include the following:
clear crypto ipsec sa
clear crypto ipsec sa entry <*ip address*> ah <*SPI*>
clear crypto ipsec sa entry <*ip address*> esp <*SPI*>
clear crypto ipsec sa map <*map name*>
clear crypto ipsec sa peer <*ip address*>

Syntax Description

entry <*ip address*>	Clear only the SAs related to a certain destination IP address.
ah <*SPI*>	Clear only a portion of the SAs by specifying the AH (authentication header) protocol and a security parameter index (SPI). You can determine the correct SPI value using the **show crypto ipsec sa** command.
esp <*SPI*>	Clear only a portion of the SAs by specifying the ESP (encapsulating security payload) protocol and a security parameter index (SPI). You can determine the correct SPI value using the **show crypto ipsec sa** command.
map <*map name*>	Clear only the SAs associated with the crypto map name given.
peer <*ip address*>	Clear only the SAs associated with the far-end peer IP address given.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 4.1	Command was introduced

Usage Examples

The following example clears all IPSec SAs:

> **enable**
#**clear crypto ipsec sa**

The following example clears the IPSec SA used for ESP traffic with the SPI of 300 to IP address 192.168.1.1:

> **enable**
#**clear crypto ipsec sa entry 192.168.1.1 esp 300**

clear event-history

Use the **clear event-history** command to clear all messages logged to the local event-history.

> **WARNING** *Messages cleared from the local event-history (using the **clear event-history** command) are no longer accessible.*

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following example clears all local event-history messages:

```
> enable
# clear event-history
```

clear ip policy-sessions

Use the **clear ip policy-sessions** command to clear policy class sessions. You may clear all the sessions or a specific session. Refer to the **show ip policy-sessions** for a current session listing. The following lists the complete syntax for the **clear ip policy-sessions** commands:

clear ip policy-sessions
clear ip policy-sessions <*classname*> [ahp | esp | gre | icmp | tcp | udp | <*protocol*>] <*source ip*> <*source port*><*dest ip*><*dest port*>
clear ip policy-sessions <*classname*> [ahp | esp | gre | icmp | tcp | udp | <*protocol*>] <*source ip*> <*source port*><*dest ip*><*dest port*> [destination | source] <*nat ip*><*nat port*>

Syntax Description

<*classname*>	Alphanumeric descriptor for identifying the configured access policy (access policy descriptors are not case-sensitive).
<*protocol*>	A specific protocol (valid range: 0-255).
<*source ip*>	Specifies the source IP address (format is A.B.C.D).
<*source port*>	Specifies the source port (in hex format for ahp, esp, and gre; decimal for all other protocols).
<*dest ip*>	Specifies the destination IP address (format is A.B.C.D).
<*dest port*>	Specifies the destination port (in hex format for ahp, esp, and gre; decimal for all other protocols).
[destination | source]	For NAT sessions, this specifies whether to select a NAT source or NAT destination session.
<*nat ip*>	For NAT sessions, this specifies the NAT IP address (format is A.B.C.D).
<*nat port*>	For NAT sessions, this specifies the NAT port (in hex format for ahp, esp, and gre; decimal for all other protocols).

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 2.1	Command was introduced

Functional Notes

The second half of this command, beginning with the source IP address may be copied and pasted from a row in the **show ip policy-sessions** table for easier use.

Usage Examples

The following example clears the Telnet association (TCP port 23) for policy class "pclass1" with source IP address 192.22.71.50 and destination 192.22.71.130:

> enable
 # clear ip policy-sessions pclass1 tcp 192.22.71.50 23 192.22.71.130 23

clear ip policy-stats <*classname*> **entry** <*policy class #*>

Use the **clear ip policy-stats** command to clear statistical counters for policy classes

Syntax Description

<*classname*> **Optional*	Specifies the policy class to clear. If no policy class is specified, statistics are cleared for all policies.
entry **Optional*	Use this optional keyword to clear statistics of a specific policy class entry
<*policy class #*> **Optional*	Specifies the policy class entry number.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 2.1	Command was introduced

Usage Examples

The following example clears statistical counters for all policy classes:

> **enable**
clear ip policy-stats

The following example clears statistical counters for the policy class **MatchALL**:

> **enable**
clear ip policy-stats MatchALL

clear ip route <*address*> <*subnet*> <*>

Use the **clear ip route** command to clear the IP route table (or remove a specified route).

Syntax Description

<*address*>	Specifies the IP address of the entry to remove from the route table.
<*subnet*>	Specifies the subnet mask of the entry to remove from the route table.
< * >	Removes all IP routes from the route table.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following example removes a route to the 10.200.3.0 (255.255.255.0) network from the route table:

```
> enable
# clear ip route 10.200.3.0 255.255.255.0
```

clock set *<time> <day> <month> <year>*

Use the **clock set** command to configure the system software clock. For the command to be valid, all fields must be entered. See the **Usage Example** below for an example.

Syntax Description

<time>	Sets the time of the system software clock in the format HH:MM:SS (hours:minutes:seconds).
<day>	Sets the current day of the month (valid range: 1 to 31).
<month>	Sets the current month (valid range: January to December). You need only enter enough characters to make the entry unique. This entry is not case-sensitive.
<year>	Sets the current year (valid range: 2000 to 2100).

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following example sets the system software clock for 3:42 pm, August 22 2002:

> **enable**
clock set 03:42:00 22 Au 2002

configure

Use the **configure** command to enter the Global Configuration Mode or to configure the system from memory. See *Global Configuration Mode Command Set* on page 135 for more information.

Syntax Description

terminal	Enter the Global Configuration Mode to configure the system from terminal inputs.
memory	Configure the active system with the commands located in the default configuration file stored in NVRAM.
network	Configure the system from a TFTP network host.
overwrite-network	Overwrite NVRAM memory from a TFTP network host.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following example enters the Global Configuration Mode from the Enable Command Mode:

```
> enable
# configure terminal
(config)#
```


copy <*source*> <*destination*>

Use the **copy** command to copy any file from a specified source to a specified destination.

Syntax Description

<*source*>	Specifies the current location of the file. Valid sources include: **running-config** (current running configuration file), **startup-config** (configuration file located in NVRAM), or a filename (located in FLASH memory).
<*destination*>	Specifies the destination of the copied file. Valid destinations include: **running-config** (current running configuration file), **startup-config** (configuration file located in NVRAM), or a filename (located in FLASH memory).

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following provides various sample **copy** commands:

> **enable**

Creates a copy of the file **myfile.biz** (located in FLASH memory) and names it **newfile.biz**:
copy myfile.biz newfile.biz

Creates a backup copy of the startup configuration file (and places in FLASH memory):
copy startup-config backup.bak

Copies the current running-configuration file to the startup configuration file located in NVRAM:
copy running-config startup-config

copy tftp *<destination>*

Use the **copy tftp** command to copy a file located on a network Trivial File Transfer Protocol (TFTP) server to a specified destination.

Syntax Description

<destination>	Specifies the destination of the file copied from the TFTP server. Valid destinations include: **flash** (FLASH memory), **startup-config** (the configuration file stored in NVRAM), or **running-config** (the current running configuration file).
*After entering **copy tftp** and specifying a destination, the ADTRAN OS prompts for the following information:*	
Address of remote host:	IP address of the TFTP server.
Source filename:	Name of the file to copy from the TFTP server.
Destination filename:	Specifies the filename to use when storing the copied file to FLASH memory. (Valid only for the **copy tftp flash** command.)

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following example copies myfile.biz from the TFTP server (10.200.2.4) to flash memory and labels it newfile.biz:

#**copy tftp flash**

Address of remote host?**10.200.2.4**
Source filename **myfile.biz**
Destination filename **newfile.biz**
Initiating TFTP transfer...
Received 45647 bytes.
Transfer Complete!
#

copy xmodem <*destination*>

Use the **copy xmodem** command to copy a file (using the XMODEM protocol) to a specified destination. XMODEM capability is provided in terminal emulation software such as HyperTerminal™.

Syntax Description

<*destination*>	Specifies the destination of the copied file. Valid destinations include: **flash** (FLASH memory), **startup-config** (the configuration file stored in NVRAM), or **running-config** (the current running configuration file).
*After entering **copy xmodem** and specifying a destination, the ADTRAN OS prompts for the following information:*	
Destination filename:	Specifies the filename to use when storing the copied file to FLASH memory. (Valid only for the **copy flash** command.)

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following example copies a .biz file to flash memory and labels it newfile.biz:

#**copy xmodem flash**
Destination filename **newfile.biz**
Begin the Xmodem transfer now...
Press CTRL+X twice to cancel
CCCCCC

The ADTRAN OS is now ready to accept the file on the **CONSOLE** port (using the XMODEM protocol). The next step in the process may differ depending on the type of terminal emulation software you are using. For HyperTerminal, you will now select **Transfer** > **Receive File** and browse to the file you wish to copy. Once the transfer is complete, information similar to the following is displayed:

Received 231424 bytes.
Transfer complete.

debug access-list *<listname>*

Use the **debug access-list** command to activate debug messages (for a specified list) associated with access list operation. Debug messages are displayed (real-time) on the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

Syntax Description

<listname>	Specifies a configured access list

Default Values

By default, all debug messages in the ADTRAN OS are disabled.

Command Modes

#	Enable Command Mode

Command History

Release 2.1	Command was introduced

Functional Notes

The **debug access-list** command provides debug messages to aid in troubleshooting access list issues.

Usage Examples

The following example activates debug messages for the access list labeled **MatchAll**:

```
> enable
# debug access-list MatchAll
```

debug crypto [ike | ike negotiation | ike client configuration | ipsec]

Use the **debug crypto** command to activate debug messages associated with IKE and IPSec functions. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

Syntax Description

ike	Displays all IKE debug messages.
ike negotiation	Displays only IKE key management debug messages (e.g., handshaking).
ike client configuration	Displays mode-config exchanges as they take place over the IKE SA. It is enabled independently from the **ike negotiation** debug described previously.
ipsec	Displays all IPSec debug messages.

Default Values

By default, all debug messages in the ADTRAN OS are disabled.

Command Modes

#	Enable Command Mode

Command History

Release 4.1	Command was introduced

Usage Examples

The following example activates the IPSec debug messages:

> **enable**
debug crypto ipsec

debug dial-backup

Use the **debug dial-backup** command to activate debug messages associated with dial-backup operation. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

Syntax Description

No subcommands

Default Values

By default, all debug messages in the ADTRAN OS are disabled.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced
Release 2.1	Additional debug messages were implemented for dial-backup operation to ADTRAN's IQ and Express series products

Functional Notes

The **debug dial-backup** command activates debug messages to aid in the troubleshooting of dial-backup links.

Usage Examples

The following example activates debug messages for dial-backup operation:

> **enable**
debug dial-backup

debug dialup-interfaces

Use the **debug dialup-interfaces** command to generate debug messages used to aid in troubleshooting problems with all dialup interfaces such as the modem or the BRI cards. Use the **no** version of this command to disable it.

Syntax Description

No subcommands

Default Values

By default, all debug messages in the ADTRAN OS are disabled.

Command Modes

#	Enable Command Mode

Command History

Release 2.1	Command was introduced

Functional Notes

When enabled, these messages provide status information on incoming calls, dialing and answering progress, etc. These messages also give information on why certain calls are dropped or rejected. It is beneficial to use this command when troubleshooting dial backup (in addition to the **debug dial-backup** command).

Usage Examples

The following example activates the debug messages for dialup interfaces:

> **enable**
debug dialup-interfaces

debug firewall

Use the **debug firewall** command to activate debug messages associated with the ADTRAN OS firewall operation. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

Syntax Description

No subcommands

Default Values

By default, all debug messages in the ADTRAN OS are disabled.

Command Modes

#	Enable Command Mode

Command History

Release 2.1	Command was introduced

Functional Notes

The **debug firewall** command activates debug messages to provide real-time information about the ADTRAN OS stateful inspection firewall operation.

Usage Examples

The following example activates the debug messages for the ADTRAN OS stateful inspection firewall:

```
> enable
# debug firewall
```

debug frame-relay [events | llc2 | lmi]

Use the **debug frame-relay** command to activate debug messages associated with the frame relay operation. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

Syntax Description

events	Activates debug messages for generic frame relay events (such as frame relay interface state)
llc2	Activates debug messages for the logical link control layer
lmi	Activates debug messages for the local management interface (such as DLCI status signaling state, etc.)

Default Values

By default, all debug messages in the ADTRAN OS are disabled.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Functional Notes

The **debug frame-relay** command activates debug messages to aid in the troubleshooting of frame relay links.

Usage Examples

The following example activates all possible debug messages associated with frame relay operation:

> **enable**
debug frame-relay events
debug frame-relay llc2
debug frame-relay lmi

debug interface [ethernet | shdsl]

Use the **debug interface** command to activate debug messages associated with the Ethernet or SHDSL interface. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

Syntax Description

ethernet	Activates debug messages for the ethernet network interface.
shdsl	Activates debug messages for the SHDSL (errors and events).

Default Values

By default, all debug messages in the ADTRAN OS are disabled.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Functional Notes

The **debug interface** command activates debug messages to aid in the troubleshooting of physical interfaces (ethernet and SHDSL).

Usage Examples

The following example activates all possible debug messages associated with the Ethernet port:

> **enable**
debug interface ethernet

debug ip dhcp-client

Use the **debug ip dhcp-client** command to activate debug messages associated with DHCP client operation in the ADTRAN OS. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

Syntax Description

No subcommands

Default Values

By default, all debug messages in the ADTRAN OS are disabled.

Command Modes

#	Enable Command Mode

Command History

Release 2.1	Command was introduced

Functional Notes

The **debug ip dhcp-client** command activates debug messages to provide information on DHCP client activity in the ADTRAN OS. The ADTRAN OS DHCP client capability allows interfaces to dynamically obtain an IP address from a network DHCP server.

Usage Examples

The following example activates debug messages associated with DHCP client activity:

> **enable**
debug ip dhcp-client

debug ip dhcp-server

Use the **debug ip dhcp-server** command to activate debug messages associated with DHCP server operation in the ADTRAN OS. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

Syntax Description

No subcommands

Default Values

By default, all debug messages in the ADTRAN OS are disabled.

Command Modes

#	Enable Command Mode

Command History

Release 2.1	Command was introduced

Functional Notes

The **debug ip dhcp-server** command activates debug messages to provide information on DHCP server activity in the ADTRAN OS. The ADTRAN OS DHCP server capability allows the ADTRAN OS to dynamically assign IP addresses to hosts on the network.

Usage Examples

The following example activates debug messages associated with DHCP server activity:

> **enable**
debug ip dhcp-server

debug ip dns-client

Use the **debug ip dns-client** command to activate debug messages associated with DNS (domain naming system) client operation in the ADTRAN OS. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

Syntax Description

No subcommands

Default Values

By default, all debug messages in the ADTRAN OS are disabled.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Functional Notes

The **debug ip dns-client** command activates debug messages to provide information on DNS client activity in the ADTRAN OS. The IP DNS capability allows for DNS-based host translation (name-to-address).

Usage Examples

The following example activates debug messages associated with DNS client activity:

> **enable**
debug ip dns-client

debug ip dns-proxy

Use the **debug ip dns-proxy** command to activate debug messages associated with DNS (domain naming system) proxy operation in the ADTRAN OS. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

Syntax Description

No subcommands

Default Values

By default, all debug messages in the ADTRAN OS are disabled.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Functional Notes

The **debug ip dns-proxy** command activates debug messages to provide information on DNS proxy activity in the ADTRAN OS. The IP DNS capability allows for DNS-based host translation (name-to-address).

Usage Examples

The following example activates debug messages associated with DNS proxy activity:

> **enable**
debug ip dns-proxy

debug ip icmp [send | recv]

Use the **debug ip icmp** command to show all ICMP messages as they come into the router or are originated by the router. If an optional keyword (**send** or **recv**) is not used, all results are displayed. Use the **no** form of this command to disable the debug messages.

Syntax Description

send **Optional*	Optional keyword which allows you to only display ICMP messages sent by the router.
recv **Optional*	Optional keyword which allows you to only display ICMP messages received by the router.

Default Values

By default, all debug messages in the ADTRAN OS are disabled.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Usage Examples

The following example activates the **debug ip icmp** send and receive messages for the ADTRAN OS:

> **enable**
debug ip icmp

ICMP SEND: From (0.0.0.0) to (172.22.14.229) Type=8 Code=0 Length=72 Details:echo request
ICMP RECV: From (172.22.14.229) to (10.100.23.19) Type=0 Code=0 Length=72 Details:echo reply
ICMP SEND: From (0.0.0.0) to (172.22.14.229) Type=8 Code=0 Length=72 Details:echo request
ICMP RECV: From (172.22.14.229) to (10.100.23.19) Type=0 Code=0 Length=72 Details:echo reply
ICMP RECV: From (172.22.255.200) to (10.100.23.19) Type=11 Code=0 Length=36 Details:TTL equals 0 during transit
ICMP RECV: From (172.22.14.229) to (10.100.23.19) Type=3 Code=3 Length=36 Details:port unreachable
ICMP RECV: From (172.22.14.229) to (10.100.23.19) Type=3 Code=3 Length=36 Details:port unreachable

debug ip ospf

Use the **debug ip ospf** command to activate debug messages associated with OSPF routing operations. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

Syntax Description

adj	Display OSPF adjacency events
events	Display OSPF events
flood	Display OSPF flooding
lsa-generation	Display OSPF link state advertisement generation
packet	Display OSPF packets
retransmission	Display OSPF retransmission events
spf	Display OSPF shortest-path-first calculations
tree	Display OSPF database tree
hello	Display OSPF hello events
database-timer	Display OSPF database timer

Default Values

By default, all debug messages in the ADTRAN OS are disabled.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Usage Examples

The following is an example of **debug ip ospf** command results:

> **enable**
debug ip ospf flood
OSPF: Update LSA: id=c0a8020d rtid=192.168.2.13 area=11.0.0.0 type=1
OSPF: Update LSA: id=0b003202 rtid=11.0.50.2 area=11.0.0.0 type=1
OSPF: Queue delayed ACK lasid=0b003202 lsartid=11.0.50.2 nbr=11.0.50.2
OSPF: Rx ACK lasid=c0a8020d lsartid=192.168.2.13 nbr=11.0.50.2
OSPF: Received LSA ACK LSA_ID=-64.-88.2.13 LSA_RT_ID=-64.-88.2.13
OSPF: Rx ACK lasid=00000000 lsartid=192.168.2.13 nbr=11.0.50.2
OSPF: Received LSA ACK LSA_ID=0.0.0.0 LSA_RT_ID=-64.-88.2.13
OSPF: Sending delayed ACK
OSPF: Update LSA: id=c0a8020d rtid=192.168.2.13 area=11.0.0.0 type=1
OSPF: Flooding out last interface
OSPF: Update LSA: id=0b003202 rtid=11.0.50.2 area=11.0.0.0 type=1

debug ip rip [events]

Use the **debug ip rip** command to activate debug messages associated with Routing Information Protocol (RIP) operation in the ADTRAN OS. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

Syntax Description

events ***Optional**	Use this optional keyword to display only RIP protocol events.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Functional Notes

The **debug ip rip** command activates debug messages to provide information on Routing Information Protocol (RIP) activity in the ADTRAN OS. RIP allows hosts and routers on a network to exchange information about routes.

Usage Examples

The following example activates debug messages associated with RIP activity:

> **enable**
debug ip rip

debug ip tcp events

Use the **debug ip tcp events** command to activate debug messages associated with significant TCP events such as state changes, retransmissions, session aborts, etc., in the ADTRAN OS. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

These debug events are logged for packets that are sent or received from the router. Forwarded TCP packets are not included.

Syntax Description

No subcommands.

Default Values

By default, all debug messages in the ADTRAN OS are disabled.

Command Modes

#	Enable Command Mode

Command History

Release 4.1	Command was introduced

Functional Notes

In the **debug ip tcp events** information, TCB stands for TCP task control block. The numbers which sometimes appear next to TCB (e.g., **TCB5** in the following example) represent the TCP session number. This allows you to differentiate debug messages for multiple TCP sessions.

Usage Examples

The following is sample output for this command:

```
> enable
# debug ip tcp events
2003.02.17 07:40:56 IP.TCP EVENTS TCP: Allocating block 5
2003.02.17 07:40:56 IP.TCP EVENTS TCB5: state change: FREE->SYNRCVD
2003.02.17 07:40:56 IP.TCP EVENTS TCB5: new connection from 172.22.75.246:3473 to
10.200.2.201:23
2003.02.17 07:40:56 IP.TCP EVENTS TCB5: state change: SYNRCVD->ESTABLISHED
[172.22.75.246:3473]
2003.02.17 07:41:06 IP.TCP EVENTS TCB5: Connection aborted -- error = RESET
2003.02.17 07:41:06 IP.TCP EVENTS TCB5: De-allocating tcb
```


debug ip udp

Use the **debug ip udp** command to activate debug messages associated with UDP send and receive events in the ADTRAN OS. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

These debug events are logged for packets that are sent or received from the router. Forwarded UDP packets are not included.

The overhead associated with this command takes up a large portion of your router's resources and at times can halt other router processes. It is best to only use the command during times when the network resources are in low demand (non-peak hours, weekends, etc.).

Syntax Description

No subcommands.

Default Values

By default, all debug messages in the ADTRAN OS are disabled.

Command Modes

#	Enable Command Mode

Command History

Release 4.1	Command was introduced

Functional Notes

In the **debug ip udp** information, the message **no listener** means that there is no service listening on this UDP port (i.e., the data is discarded).

Usage Examples

The following is sample output for this command:

> **enable**

debug ip udp

2003.02.17 07:38:48 IP.UDP RX: src=10.200.3.236:138, dst=10.200.255.255:138, 229 bytes , no listener
2003.02.17 07:38:48 IP.UDP RX: src=10.200.2.7:138, dst=10.200.255.255:138, 227 bytes , no listener
2003.02.17 07:38:48 IP.UDP RX: src=10.200.201.240:138, dst=10.200.255.255:138, 215 bytes , no listener

debug ppp [authentication | errors | negotiation | verbose]

Use the **debug ppp** command to activate debug messages associated with point-to-point protocol (PPP) operation in the ADTRAN OS. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

Syntax Description

authentication	Activates debug messages pertaining to PPP authentication (CHAP, PAP, EAP, etc.).
errors	Activates debug messages that indicate a PPP error was detected (mismatch in negotiation authentication, etc.).
negotiation	Activates debug messages associated with PPP negotiation.
verbose	Activates detailed debug messages for PPP operation.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Functional Notes

The **debug ppp** command activates debug messages to provide information on PPP activity in the system. PPP debug messages can be used to aid in troubleshooting PPP links.

Usage Examples

The following example activates debug messages associated with PPP authentication activity:

> **enable**
debug ppp authentication

debug sntp

Use the **debug sntp** command to enable debug messages associated with the Simple Network Time Protocol (SNTP). All SNTP Packet Exchanges and time decisions are displayed with these debugging events enabled. Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

Syntax Description

No subcommands

Default Values

By default, all debug messages in the ADTRAN OS are disabled.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Functional Notes

The **debug sntp** command activates debug messages to aid in troubleshooting SNTP protocol issues.

Usage Examples

The following is an example output for the **debug sntp** command:

> **enable**
debug sntp
#**config term**
(config)#**sntp server timeserver.localdomain**
2002.12.11 15:06:37 SNTP.CLIENT sent Version 1 SNTP time request to 192.168.1.1
2002.12.11 15:06:37 SNTP.CLIENT received SNTP reply packet from 192.168.1.1
2002.12.11 15:06:37 SNTP.CLIENT setting time to 12-11-2002 15:06:02 UTC
2002.12.11 15:06:37 SNTP.CLIENT waiting for 86400 seconds for the next poll interval

debug system

Use the **debug system** command to enable debug messages associated with system events (i.e., login, logouts, etc.). Debug messages are displayed (real-time) to the terminal (or Telnet) screen. Use the **no** form of this command to disable the debug messages.

Syntax Description

No subcommands

Default Values

By default, all debug messages in the ADTRAN OS are disabled.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Usage Examples

The following example activates debug messages associated with system information:

> **enable**
debug system

dir

Use the **dir** command to display a directory list of files on the system.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Usage Examples

The following is sample output from the **dir** command:

```
> enable
# dir
Files:
 988161 NV3200A-02-00-11.biz
   1152 startup-config
   1113 startup-config.bak
1739729 030018adv.biz
 231424 boot030015.biz
1352150 NV3200A-E03-00-17.biz
 232894 boot030018.biz
1812281 NV3200A-E03-00-20-adv.biz
6366976 bytes used, 335104 available, 6702080 total
```

disable

Use the **disable** command to exit the Enable Command Mode and enter the Basic Command Mode.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following example exits the Enable Command Mode and enters the Basic Command Mode:

```
# disable
>
```


erase [*<filename>* | **startup-config**]

Use the **erase** command to erase the specified file.

Syntax Description

<filename>	Specifies the name of the file (located in FLASH memory) to erase.
startup-config	Erases the startup configuration file stored in NVRAM.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following example erases the startup configuration file stored in NVRAM:

> **enable**
erase startup-config

If a new startup-configuration file is not specified before power-cycling the unit, the ADTRAN OS will initialize using a default configuration.

events

Use the **events** command to enable event reporting to the current CLI session. Use the **no** form of this command to disable all event reporting to the current CLI session.

Syntax Description

No subcommands

Default Values

By default, this command is enabled.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Usage Examples

The following example enables event reporting:

> **enable**
events

logout

Use the **logout** command to terminate the current session and return to the login screen.

Syntax Description

No subcommands

Default Values

No defaults necessary for this command.

Command Modes

> or #	Basic or Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following example shows the logout command being executed in Enable Mode:

logout

Session now available

Press RETURN to get started.

ping <*address*>

Use the **ping** command (at the Enable Command Mode prompt) to verify IP network connectivity.

Syntax Description

<*address*> ***Optional*	Specifies the IP address of the system to ping. Entering the **ping** command with no specified address prompts the user with parameters for a more detailed **ping** configuration. See **Functional Notes** (below) for more information.

Default Values

No default value necessary for this command.

Command Modes

> or #	Basic or Enable Command Mode

Command History

Release 1.1	Command was introduced

Functional Notes

The **ping** command helps diagnose basic IP network connectivity using the Packet InterNet Groper program to repeatedly bounce Internet Control Message Protocol (ICMP) Echo_Request packets off a system (using a specified IP address). The ADTRAN OS allows executing a standard **ping** request to a specified IP address or provides a set of prompts to configure a more specific **ping** configuration.

The following is a list of output messages from the **ping** command:

!	Success
-	Destination Host Unreachable
$	Invalid Host Address
X	TTL Expired in Transit
?	Unknown Host
*	Request Timed Out

The following is a list of available extended **ping** fields with descriptions:

Target IP address:	Specifies the IP address of the system to ping.
Repeat Count:	Number of ping packets to send to the system (valid range: 1 to 1000000).
Datagram Size:	Size (in bytes) of the ping packet (valid range: 1 to 1448).
Timeout in Seconds:	If a ping response is not received within the timeout period, the ping is considered unsuccessful (valid range: 1 to 5 seconds).
Extended Commands:	Specifies whether additional commands are desired for more ping configuration parameters.
Source Address (or interface):	Specifies the IP address to use as the source address in the ECHO_REQ packets.

Functional Notes (Continued)

Data Pattern:	Specifies an alphanumerical string to use (the ASCII equivalent) as the data pattern in the ECHO_REQ packets.
Sweep Range of Sizes:	Varies the sizes of the ECHO_REQ packets transmitted.
Sweep Min Size:	Specifies the minimum size of the ECHO_REQ packet (valid range: 0 to 999999).
Sweep Max Size:	Specifies the maximum size of the ECHO_REQ packet (valid range: Sweep Min Size to 1448).
Sweep Interval:	Specifies the interval used to determine packet size when performing the sweep (valid range: 1 to 1448).
Verbose Output:	Specifies an extended results output.

Usage Examples

The following is an example of a successful **ping** command:

#**ping**
Target IP address:**192.168.0.30**
Repeat count[1-1000000]:**5**
Datagram Size [1-1000000]:**100**
Timeout in seconds [1-5]:**2**
Extended Commands? [y or n]:**n**
Type CTRL+C to abort.
Legend: '!' = Success '?' = Unknown host '$' = Invalid host address
'*' = Request timed out '-' = Destination host unreachable
'x' = TTL expired in transit

Pinging 192.168.0.30 with 100 bytes of data:
!!!!!
Success rate is 100 percent (5/5) round-trip min/avg/max = 19/20.8/25 ms

reload [**cancel | in** <*delay*>]

Use the **reload** command to preform a manual reload of the ADTRAN OS.

> **WARNING** *Performing an ADTRAN OS **reload** disrupts data traffic.*

Syntax Description

cancel ***Optional**	Use the **cancel** keyword to deactivate a pending **reload** command.
in ***Optional**	Use the **in** keyword to specify a delay period the ADTRAN OS will wait before reloading.
<*delay*>	Specifies the delay period in minutes (mmm) or hours and minutes (hh:mm).

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following example reloads the ADTRAN OS software in 3 hours and 27 minutes:

> **enable**
reload in 03:27

The following example reloads the ADTRAN OS software in 15 minutes:
> **enable**
reload in 15

The following example terminates a pending reload command:
> **enable**
reload cancel

show access-lists *<listname>*

Use the **show access-lists** command to display all configured access lists in the system (or a specific list).

Syntax Description

<listname>	Specify a particular access list to display.
**Optional*	

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 2.1	Command was introduced

Functional Notes

The **show access-lists** command displays all configured access-lists in the system. All entries in the access list are displayed, and a counter indicating the number of packets matching the entry is listed.

Usage Examples

The following is a sample output from the **show access-lists** command:

```
> enable
# show access-lists

Standard access list MatchAll
   permit host 10.3.50.6 (0 matches)
   permit 10.200.5.0 wildcard bits 0.0.0.255 (0 matches)
Extended access list UnTrusted
   deny   icmp 10.5.60.0 wildcard bits 0.0.0.255  any  source-quench (0 matches)
   deny   tcp any  any   (0 matches)
```

show arp

Use the **show arp** command to display the Address Resolution Protocol (ARP) table.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following is a sample output of the **show arp** command:

> **enable**
show arp

ADDRESS	TTL (min)	MAC ADDRESS	LAST UPDATED (min)	INTERFACE
192.168.30.36	13	00:E0:7D:88:1A:B9	4260	eth 0/1
192.168.30.253	17	02:60:8C:DD:0A:CE	4264	eth 0/1
224.0.0.9	71578541	01:00:5E:00:00:09	0	eth 0/2

show bridge [**ethernet | frame-relay | ppp**] <*slot/port*> <*bridge group #*>

Use the **show bridge** command to display a list of all configured bridge groups (including individual members of each group). Enter an interface or a bridge number to display the corresponding list.

Syntax Description

ethernet <*slot/port*> **Optional*	Display all bridge groups associated with the Ethernet interface.
frame-relay <*slot/port*> **Optional*	Display all bridge groups associated with the frame relay virtual interface.
ppp <*slot/port*> **Optional*	Display all bridge groups associated with the PPP virtual interface.
<*bridgegroup#*> **Optional*	Display a specific bridge group.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following is a sample output from the **show bridge** command:

> **enable**
show bridge

Total of 300 station blocks 295 free

Address	Action	Interface	Age	Rx Count	Tx Count
00:04:51:57:4D:5A	forward	eth 0/1	0	7133392	7042770
00:04:5A:57:4F:2A	forward	eth 0/1	0	402365	311642
00:10:A4:B3:A2:72	forward	eth 0/1	4	2	0
00:A0:C8:00:8F:98	forward	eth 0/1	0	412367	231
00:E0:81:10:FF:CE	forward	fr 1.17	0	1502106	1486963

show buffers

Use the **show buffers** command to display the statistics for the buffer pools on the network server.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

> or #	Basic or Enable Command Mode

Command History

Release 3.1	Command was introduced

Usage Examples

The following is a sample output from the **show buffers** command:

#**show buffers**
Buffer handles: 119 of 2000 used.

Pool	Size	Total	Used	Available	Max. Used
0	1800	1894	119	1775	122
1	2048	64	0	64	0
2	4096	32	0	32	0
3	8192	4	0	4	0
4	16384	2	0	2	0
5	32768	2	0	2	0
6	65536	2	0	2	0

show buffers users

Use the **show buffers users** command to display a list of the top users of packet buffers. Typically, this command will only be used as a debug tool by ADTRAN personnel.

Syntax Description

No subcommands.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 4.1	Command was introduced

Usage Examples

The following is a sample from the **show buffers users** command:

> **enable**
show buffers users
Number of users: 7

Rank	User	Count
1	0x0052f4f8	59
2	0x0051a4fc	32
3	0x00528564	8
4	0x0053c1c8	7
5	fixedsize	5
6	0x001d8298	2
7	0x0010d970	1
8	0x00000000	0
9	0x00000000	0
10	0x00000000	0
11	0x00000000	0
12	0x00000000	0
13	0x00000000	0
14	0x00000000	0
15	0x00000000	0

show clock [detail]

Use the **show clock** command to display the system time and date entered using the **clock set** command. See *clock set <time> <day> <month> <year>* on page 38 for more information.

Syntax Description

detail **Optional*	Use this optional keyword to display more detailed clock information, including the time source.

Default Values

No default value necessary for this command.

Command Modes

> or #	Basic or Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following example displays the current time and data from the system clock:

>**show clock**

23:35:07 UTC Tue Aug 20 2002

show configuration

Use the **show configuration** command to display a text printout of the startup configuration file stored in NVRAM.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following is a sample output of the **show configuration** command:

```
> enable
# show configuration
!
!
no enable password
!
ip subnet-zero
ip classless
ip routing
!
event-history on
no logging forwarding
logging forwarding priority-level info
no logging email
!
ip policy-timeout tcp all-ports 600
ip policy-timeout udp all-ports 60
ip policy-timeout icmp 60
!
!
!
interface eth 0/1
speed auto
  no ip address
  shutdown
```


Usage Examples (Continued)

```
!
interface dds 1/1
  shutdown
!
interface bri 1/2
  shutdown
!
!
ip access-list standard Outbound
  permit host 10.3.50.6
  permit 10.200.5.0 0.0.0.255
!
!
ip access-list extended UnTrusted
  deny   icmp 10.5.60.0 0.0.0.255  any  source-quench
  deny   tcp any  any
!
no ip snmp agent
!
!
!
line con 0
  no login
!
line telnet 0
  login
line telnet 1
  login
line telnet 2
  login
line telnet 3
  login
line telnet 4
  login
!
```


show crypto ike

Use the **show crypto ike** command to display information regarding the IKE configuration.

Variations of this command include the following:
show crypto ike client configuration pool
show crypto ike client configuration pool <*poolname*>
show crypto ike policy
show crypto ike policy <*policy priority*>
show crypto ike remote-id <*remote-id*>
show crypto ike sa

Syntax Description

client configuration pool	Displays the list of all configured IKE client configuration pools.
<*poolname*>	Displays detailed information regarding the specified IKE client configuration pool.
policy	Displays information on all IKE policies. Indicates if client configuration is enabled for the IKE policies and displays the pool names.
< *policy priority*>	Displays detailed information on the specified IKE policy. This number is assigned using the **crypto ike policy** command. See *crypto ike* on page 150 for more information.
remote-id <*remote-id*>	Displays information on all IKE information regarding the remote-id. The remote-id value is specified using the **crypto ike remote-id** command (see *crypto ike remote-id* on page 154).
sa	Displays the configuration of active IKE security associations.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 4.1	Command was introduced

Usage Examples

The following is a sample from the **show crypto ike policy** command:

```
> enable
# show crypto ike policy
Crypto IKE Policy 100
  Main mode
  Using System Local ID Address
  Peers:
  192.168.1.2
initiate main
respond anymode
  Attributes:
    10
      Encryption: 3DES
      Hash: SHA
      Authentication: Pre-share
      Group: 1
      Lifetime: 900 seconds
```

show crypto ipsec

Use the **show crypto ipsec** command to display information regarding the IPSec configuration.

Variations of this command include the following:
show crypto ipsec sa
show crypto ipsec sa address <*ip address*>
show crypto ipsec sa map <*mapname*>
show crypto ipsec transform-set
show crypto ipsec transform-set <*transform-set name*>

Syntax Description

sa	Displays all IPSec security associations.
address <*ip address*>	Displays all IPSec security associations associated with the designated peer IP address.
map <*mapname*>	Displays all IPSec security associations associated with the designated crypto map name.
transform-set	Displays all defined transform-sets.
<*transform-set name*>	Displays information for a specific transform-set.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 4.1	Command was introduced

Usage Examples

The following are samples from the **show crypto ipsec transform-set** command:

```
> enable
# show crypto ipsec transform-set

Transform Set "MySet"
  ah-md5-hmac
  mode tunnel

Transform Set "Set1"
  esp-3des esp-sha-hmac
  mode tunnel

Transform Set "esp-des"
  esp-des
  mode tunnel
```


show crypto map

Use the **show crypto map** command to display information regarding crypto map settings.

Variations of this command include the following:
show crypto map
show crypto map interface ethernet <*slot/port*>
show crypto map interface frame-relay <*port number*>
show crypto map interface loopback <*port number*>
show crypto map interface ppp <*port number*>
show crypto map <*map name*>
show crypto map <*map name*> <*map number*>

Syntax Description

interface	Displays the map settings for the specified interface. Valid interfaces include: Ethernet, frame-relay, frame-relay sublinks, loopback, or PPP.
<*slot/port*>	For Ethernet interfaces, designate the slot and port number.
<*port number*>	For frame-relay, loopback, and PPP ports, enter the port number (range is 1-1024). For frame-relay sublinks, the syntax is <port#.sublink#> (range for sublinks is 1-1007).
<*map name*>	Enter a specific crypto map name.
<*map number*>	Enter a specific crypto map number.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 4.1	Command was introduced

Usage Examples

The following is a sample from the **show crypto map** command:

```
> enable
#show crypto map testMap

Crypto Map "testMap" 10 ipsec-ike
  Extended IP access list NewList
  Peers:
    192.168.1.1
  Transform sets:
    esp-des
  Security-association lifetimes:
    0 kilobytes
    86400 seconds
  No PFS group configured
  Interfaces using crypto map testMap:
    eth 0/1
```


show debugging

Use the **show debugging** command to display a list of all activated debug message categories.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following is a sample output from the **show debugging** command:

> **enable**
show debugging

debug access-list MatchAll
debug firewall
debug ip rip
debug frame-relay events
debug frame-relay llc2
debug frame-relay lmi

show dial-backup interfaces

Use the **show dial-backup interfaces** command to display all configured dial-backup interfaces and the associated parameters for each.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following example enters the Enable Command Mode and uses the show command to display dial-backup interface information:

> **enable**
show dial-backup interfaces

Dial-backup interfaces...
fr 1.16 backup interface:

Backup state:	idle
Remote DLCI:	16
Backup Protocol:	Frame Relay
IQ handshake:	disabled
Call mode:	originate and answer always
Auto-backup:	enabled
Auto-restore:	enabled
Priority:	60
Backup delay:	10 seconds
Restore delay:	10 seconds
Connect timeout:	60 seconds
Redial retries:	unlimited
Redial Delay:	10 seconds

Usage Examples (Continued)

Backup enabled all day on the following days:
Sunday Monday Tuesday Wednesday Thursday Friday Saturday

Backup phone number list:

Number	Call Type	min/max DS0s
3332222	64K	1/1

show dialin interfaces

Use the **show dialin interfaces** command to display information regarding remote console dialin.

.

Syntax Description

No subcommands.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 4.1	Command was introduced

Usage Examples

The following is sample output from the **show dialin interfaces** command:

```
> enable
# show dialin interfaces
Dialin interfaces...
modem 1/3 dialin interface:
  Connection Status: Connected
Caller id info : name-John Smith number-5551212 time-14:23:10 2/17/2003
```


show event-history

Use the **show event-history** command to display all entries in the current local event-history log.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The event history provides useful information regarding the status of the system and individual port states. Use the event history as a troubleshooting tool when identifying system issues. The following is a sample event history log.

```
>enable
# show event-history
Using 526 bytes
2002.07.12 15:34:01 T1.t1 1/1 Yellow
2002.07.12 15:34:01 INTERFACE_STATUS.t1 1/1 changed state to down.
2002.07.12 15:34:02 T1.t1 1/1 No Alarms
2002.07.12 15:34:02 INTERFACE_STATUS.t1 1/1 changed state to up.
2002.07.12 15:34:03 INTERFACE_STATUS.eth 0/1 changed state to up.
2002.07.12 15:34:10 OPERATING_SYSTEM Warm Start
2002.07.12 15:34:12 PPP.NEGOTIATION LCP up
2002.07.12 15:34:12 PPP.NEGOTIATION IPCP up
```

show flash

Use the **show flash** command to display a list of all files currently stored in FLASH memory.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following is a sample **show flash** output:

```
> enable
# show flash

Files:
 245669 010100boot.biz
1141553 new.biz
    821 startup-config
   1638 startup-config.old
1175679 020016.biz
    821 startup-config.bak
2572304 bytes used 4129776 available 6702080 total
```


show frame-relay [**lmi** | **pvc** <*interface*>] **frame-relay** <*interface*>

Use the **show frame-relay** command to display configuration and status parameters for configured virtual frame relay interfaces.

Syntax Description

lmi	Displays LMI (Link Management Interface) statistics for each virtual frame relay interface
pvc	Displays PVC (Permanent Virtual Circuit) configuration and statistics for all virtual frame relay interfaces (or a specified interface)
<*interface*>	Specifies the virtual frame relay interface (for example fr 1)
frame-relay	Optional keyword used to display frame relay PVC statistics for a specific frame relay interface.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following are sample outputs from various **show frame-relay** commands:

> **enable**
show frame-relay lmi

LMI statistics for interface FR 1 LMI TYPE = ANSI
Num Status Enq. Sent 79 Num Status Msgs Rcvd 71
Num Update Status Rcvd 12 Num Status Timeouts 5

show frame-relay pvc

Frame Relay Virtual Circuit Statistics for interface FR 1

	Active	Inactive	Deleted	Static
local	2	0	0	2

DLCI = 16 DLCI USAGE = LOCAL PVC STATUS = ACTIVE INTERFACE = FR 1.16

MTU: 1500

input pkts: 355	output pkts: 529	in bytes: 23013
out bytes: 115399	dropped pkts: 13	in FECN pkts: 0

Usage Examples (Continued)

```
in BECN pkts: 0          in DE pkts: 0          out DE pkts: 0
pvc create time: 00:00:00:12          last time pvc status changed: 00:00:13:18
DLCI = 20 DLCI USAGE = LOCAL  PVC STATUS = ACTIVE INTERFACE = FR 1.20
MTU: 1500
input pkts: 0          output pkts: 44          in bytes: 0
out bytes: 22384          dropped pkts: 11          in FECN pkts: 0
in BECN pkts: 0          in DE pkts: 0          out DE pkts: 0
pvc create time: 00:00:01:25          last time pvc status changed: 00:00:13:18
```

show hosts

Use the **show hosts** command to display information such as the domain name, name lookup service, a list of name server hosts, and the cached list of host names and addresses on the network to which you can connect.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Functional Notes

The list below describes the fields contained in the host table:

- Flags: Indicate whether the entry is permanent (P) or temporary (T) and if the entry is OK or expired (EXP).
- Age: Indicates how old the entry is.
- Type: Shows the protocol type.
- Address: Displays the IP address for the entry.

Usage Examples

The following is sample output from the **show hosts** command:

>**enable**
#**show hosts**
Name/address lookup uses domain name service
DNS Proxy is disabled
Default domain is not set
Name servers are 1.1.1.1 2.2.2.2
Host

	Flags	Age	Type	Address
Example1	(P OK)	- -	IP	1.1.1.1
Example2	(P OK)	- -	IP	2.2.2.2

show interfaces *<interface>*

Use the **show interfaces** command to display configuration parameters and current statistics for all interfaces (or a specified interface).

Syntax Description

<interface>	Specific interface to display. Type **show interfaces ?** for a complete list of valid interfaces.
**Optional*	

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following are samples from various **show interfaces** commands:

> **enable**
show interfaces t1 1/1

```
t1 1/1 is UP
  T1 coding is B8ZS framing is ESF
  Clock source is line FDL type is ANSI
  Line build-out is 0dB
  No remote loopbacks No network loopbacks

  DS0 Status: 12345678901234567890123 4
              NNNNNNNNNNNNNNNNNNNNNNNN

  Line Status: -- No Alarms --

  Current Performance Statistics:
    0 Errored Seconds 0 Bursty Errored Seconds
    0 Severely Errored Seconds 0 Severely Errored Frame Seconds
    0 Unavailable Seconds 0 Path Code Violations
    0 Line Code Violations 0 Controlled Slip Seconds
    0 Line Errored Seconds 0 Degraded Minutes
```


Usage Examples (Continued)

```
# show interfaces fr 1
TDM group 10 line protocol is UP
Encapsulation FRAME-RELAY (fr 1)
   463 packets input 25488 bytes 0 no buffer
   0 runts 0 giants 0 throttles
   0 input errors 0 CRC 0 frame
   0 abort 0 ignored 0 overruns
   864 packets output 239993 bytes 0 underruns
   0 input clock glitches 0 output clock glitches
   0 carrier lost 0 cts lost

  Line Status: -- No Alarms --

  Current Performance Statistics:
   0 Errored Seconds 0 Bursty Errored Seconds
   0 Severely Errored Seconds 0 Severely Errored Frame Seconds
   0 Unavailable Seconds 0 Path Code Violations
   0 Line Code Violations 0 Controlled Slip Seconds
   0 Line Errored Seconds 0 Degraded Minutes

# show interfaces modem 1/2
modem 1/2 is UP
  Line status: on-hook
  Caller ID will be used to route incoming calls
   0 packets input 0 bytes 0 no buffer
   0 runts 0 giants 0 throttles
   0 input errors 0 CRC 0 frame
   0 abort 0 ignored 0 overruns
   0 packets output 0 bytes 0 underruns
   0 input clock glitches 0 output clock glitches
   0 carrier lost 0 cts lost

# show interfaces eth 0/1
Ip address is 10.200.1.50
  Netmask is 255.255.0.0
  MTU is 1500
  Fastcaching is Enabled
  RIP Authentication is Disabled
  RIP Tx uses global version value
  RIP Rx uses global version value
```

Usage Examples (Continued)

```
# show interfaces dds 1/1
dds 1/1 is UP line protocol is UP
  Encapsulation FRAME-RELAY (fr 1)
  Loop rate is set to 56000 actual rate is 56000
  Clock source is line
  Data scrambling is disabled
  No Loopbacks
    75 packets input 6108 bytes 0 no buffer
    0 runts 0 giants 0 throttles
    0 input errors 0 CRC 0 frame
    0 abort 0 ignored 0 overruns
    81 packets output 11496 bytes 0 underruns
    0 input clock glitches 0 output clock glitches
    0 carrier lost 0 cts lost
```


show interfaces shdsl

Use the **show interfaces shdsl** command to display configuration parameters and current statistics for the SHDSL interfaces (or a specified interface).

Variations of this command include the following:
show interfaces shdsl <*slot/port*>
show interfaces shdsl <*slot/port*> performance-statistics
show interfaces shdsl <*slot/port*> performance-statistics 24-hour
show interfaces shdsl <*slot/port*> performance-statistics <*x-y*>
show interfaces shdsl <*slot/port*> version

Syntax Description

performance statistics **Optional*	Displays the current 15-minute interval, the current 24-hour totals, and all 96 stored intervals.
performance-statistics 24-hour **Optional*	Displays the current 24-hour totals and the past seven 24-hour intervals.
performance-statistics <*x-y*>	Shows the current 15-minute interval, the current 24-hour totals, and all intervals from x through y. This command is basically the same thing as the **performance-statistics** command with the added function of allowing you to specify a particular interval (or range of intervals) to display rather than displaying all 96. *Note: If you wish to display the 24th interval, enter **show interface shdsl 1/1 performance-statistics 24-24**. Entering **show interface shdsl 1/1 performance-statistics 24** results in displaying the 24-hour statistics. Any number other than 24 (between 1 and 96) results in the correct display of the selected interval (e.g., **show interface** **shdsl 1/1 performance-statistics 4** shows the 4th interval).*
version **Optional*	Displays current version information (e.g., model and list number, software version, etc.) for the SHDSL interface.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 4.1	Command was introduced

Functional Notes

The following is a list of output messages from the **show interfaces shdsl** command:

Equipment Type	Shows whether the unit is operating in CPE (NT) mode or CO (LT) mode.
Line Rate	Shows the current line rate. The line rate is the data rate + 8 kbps. Therefore, a rate of 2056 kbps implies an actual data rate of 2048 kbps.
Alarms	Shows the current alarm conditions. Possible alarms are: • LOS • LOSW - Loss of synchronization word (related to frame sync) • loop attenuation (loop attenuation margin threshold has been reached or exceeded; this threshold is user selectable and disabled by default) • SNR margin (SNR margin threshold has been reached or exceeded; this threshold is also user programmable) • CRC • segment defect • segment anomaly
Loop Status	Shows additional information about the loop status as well as the Embedded Operations Channel (EOC). Possible messages are: • SHDSL training complete (marginal signal quality). Establishing EOC... • SHDSL training complete (marginal signal quality). EOC is up. • SHDSL training complete. EOC is down. • SHDSL training complete. EOC is up. • SHDSL training in progress.
Loopback State	Shows the state of local and remote loopbacks. Possible local loopback messages are: • Local dual-sided loopback • Local customer transparent loopback • Local customer non-transparent loopback • Local transparent network loopback • Local non-transparent network loopback • No local loopbacks Possible remote loopback messages are: • Remote dual-sided loopback • Remote customer transparent loopback • Remote customer non-transparent loopback • Remote transparent network loopback • Remote non-transparent network loopback • No remote loopbacks
SNR margin	Shows the current, minimum, and maximum Signal-to-Noise Ratio of the line. These may be cleared using the **clear counters shdsl <slot/port>** command.

Functional Notes (Continued)

Loop Attenuation	Shows the current, minimum, and maximum loop attenuation of the line. These may be cleared using the **clear counters shdsl <slot/port>** command.
Performance Stats	Shows current interval line statistics. These statistics may be cleared through the use of the **clear counters shdsl <slot/port>** command, but the number of elapsed seconds will continue running and accumulating time.

Usage Examples

The following is sample output from the **show interfaces shdsl** command:

```
> enable
# show interfaces shdsl 1/1

shdsl 1/1 is UP, line protocol is DOWN
  Encapsulation FRAME-RELAY IETF (fr 1)
  Equipment type is cpe
  Line rate is 2056kbps
  No alarms.
  SHDSL training complete.  EOC is up.
  No local loopbacks, No remote loopbacks
  SNR margin is 18dB currently, 15dB minimum, 30dB maximum
  Loop attenuation is 1dB currently, 1dB minimum, 1dB maximum

  Current 15-minute performance statistics (115 seconds elapsed):
    0 code violations, 0 loss of sync word seconds
    0 errored seconds, 0 severely errored seconds
    0 unavailable seconds

  Packet Statistics:
    0 packets input, 0 bytes, 0 no buffer
    0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame
    0 abort, 0 ignored, 0 overruns
    32 packets output, 0 bytes, 0 underruns
    0 input clock glitches, 0 output clock glitches
    0 carrier lost, 0 cts lost
```

Technology Review

A network loopback loops data toward the network (away from the unit). A customer loopback loops data toward the router. The router does not instigate customer-side loopbacks, only network loopbacks (remote or local). The reason for this is that the customer interface is internal to the router. There is little use for looping back router data on itself.

A transparent loopback is one in which the unit loops back one side (i.e., network) and also allows the same incoming data to be passed through to the customer side. A non-transparent loopback is one which loops back one side of the interface (network) but sends idle codes to the other side (customer). The AOS defaults to non-transparent loopbacks. The reason for this is that sending test patterns into the IP stack could cause unpredictable behavior. However, it is still possible for the network to send a transparent loopback request. Such requests will be accepted.

show ip access-lists *<listname>*

Use the **show ip access-lists** command to display all configured IP access lists in the system.

Syntax Description

<listname>	Specify a particular access list to display.
**Optional*	

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 2.1	Command was introduced

Functional Notes

The **show ip access-lists** command displays all configured IP access-lists in the system. All entries in the access list are displayed, and a counter indicating the number of packets matching the entry is listed.

Usage Examples

The following is a sample output from the **show ip access-lists** command:

```
> enable
# show ip access-lists

Standard IP access list MatchAll
   permit host 10.3.50.6 (0 matches)
   permit 10.200.5.0 wildcard bits 0.0.0.255 (0 matches)
Extended IP access list UnTrusted
   deny   icmp 10.5.60.0 wildcard bits 0.0.0.255  any  source-quench (0 matches)
   deny   tcp any  any   (0 matches)
```

show ip arp

Use the **show ip arp** command to display the Address Resolution Protocol (ARP) table.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following is a sample output of the **show ip arp** command:

> **enable**
show ip arp

ADDRESS	TTL (min)	MAC ADDRESS	LAST UPDATED (min)
192.168.30.36	13	00:E0:7D:88:1A:B9	4260
192.168.30.253	17	02:60:8C:DD:0A:CE	4264
224.0.0.9	71578541	01:00:5E:00:00:09	0

show ip dhcp-client lease *<interface>*

Use the **show ip dhcp-client lease** command to display all Dynamic Host Client Protocol (DHCP) lease information for interfaces that have dynamically assigned IP addresses.

Syntax Description

<interface> **Optional*	Displays the information for the specified interface Valid interfaces include: Ethernet (eth 0/1) and virtual frame relay interfaces (fr 1)

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 2.1	Command was introduced

Usage Examples

The following is a sample output from the **show dhcp-client lease** command:

```
> enable
# show dhcp-client lease

Interface: ethernet 0/1
  Temp IP address: 10.100.23.64 Mask: 0.0.0.0
    DHCP Lease server: 10.100.23.207 State: Bound (3)
    Lease: 120 seconds
  Temp default gateway address: 0.0.0.0
    Client-ID: N/A
```

show ip dhcp-server binding *<client ip address>*

Use the **show ip dhcp-server binding** command to display the Dynamic Host Client Protocol (DHCP) server client table with associated information.

Syntax Description

<client ip address>	Specify a particular client IP address.
**Optional*	

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 2.1	Command was introduced

Usage Examples

The following is a sample output from the **show ip dhcp-server binding** command:

> **enable**
show ip dhcp-server binding

```
IP Address      Client Id            Lease Expiration      Client Name
10.100.23.64    01:00:a0:c8:00:8f:b3  Aug 15 2002 11:02 AM  Router
```


show ip interfaces [<*interface*> **| brief**]

Use the **show ip interfaces** command to display the status information for all IP interfaces (or a specific interface).

Syntax Description

<*interface*> **Optional*	Enter a specific interface to view its status information. If no interface is entered, status information for all interfaces is displayed. Valid entries include: **FrameRelay**, **ethernet**, **loopback**, and **ppp**.
brief	Use this optional keyword to display an abbreviated version of interface statistics for all IP interfaces.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following is a sample output of the **show ip interfaces** command:

> **enable**
show ip interfaces

ADDRESS	TTL (min)	MAC ADDRESS	LAST UPDATED (min)
192.168.30.36	13	00:E0:7D:88:1A:B9	4260
192.168.30.253	17	02:60:8C:DD:0A:CE	4264
224.0.0.9	71578541	01:00:5E:00:00:09	0

show ip ospf

Use the **show ip ospf** command to display general information regarding OSPF processes.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Usage Examples

The following is a sample output from the **show ip ospf** command:

```
> enable
# show ip ospf

Summary of OSPF Process with ID: 192.2.72.101
  Supports only single Type Of Service routes (TOS 0)
  SPF delay timer: 5 seconds, Hold time between SPFs: 10 seconds
  LSA interval: 240 seconds
  Number of external LSAs: 0, Checksum Sum: 0x0
  Number of areas: 0, normal: 0, stub: 0, NSSA: 0
```


show ip ospf border-routers

Use the **show ip ospf border-routers** command to display the internal OSPF routing table entries for area border routers (ABRs) and autonomous system boundary routers (ASBRs).

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Usage Examples

> **enable**
show ip ospf border-routers

show ip ospf database

Use the **show ip ospf database** command to display information from the OSPF database regarding a specific router. There are several variations of this command which you can use to obtain information about different OSPF link state advertisements. The variations are shown below:

show ip ospf <*area-id*> database
show ip ospf <*area-id*> database adv-router <*ip address*>
show ip ospf <*area-id*> database asbr-summary <*link-state-id*>
show ip ospf <*area-id*> database asbr-summary <*link-state-id*> adv-router <*ip address*>
show ip ospf <*area-id*> database database-summary
show ip ospf <*area-id*> database external <*link-state-id*>
show ip ospf <*area-id*> database external <*link-state-id*> adv-router <*ip address*>
show ip ospf <*area-id*> database network <*link-state-id*>
show ip ospf <*area-id*> database network <*link-state-id*> adv-router <*ip address*>
show ip ospf <*area-id*> database router <*link-state-id*>
show ip ospf <*area-id*> database router <*link-state-id*> adv-router <*ip address*>
show ip ospf <*area-id*> database summary <*link-state-id*>
show ip ospf <*area-id*> database summary <*link-state-id*> adv-router <*ip address*>
show ip ospf <*area-id*> database asbr-summary <*link-state-id*> self-originate <*link-state-id*>
show ip ospf <*area-id*> database external <*link-state-id*> self-originate <*link-state-id*>
show ip ospf <*area-id*> database network <*link-state-id*> self-originate <*link-state-id*>
show ip ospf <*area-id*> database router <*link-state-id*> self-originate <*link-state-id*>
show ip ospf <*area-id*> database summary <*link-state-id*> self-originate <*link-state-id*>
show ip ospf <*area-id*> database self-originate <*link-state-id*>

Syntax Description

<*area id*> **Optional*	Area ID number associated with the OSPF address range. This range is defined in the network router configuration command used to define the particular area. See *network <ip address> <wildcard> area <area id>* on page 548 for more information.
<*link-state-id*> **Optional*	This ID number identifies the portion of the internet environment that is being described by the advertisement. The value needed in this field is tied to the advertisement's LS type.
<*ip address*>	Enter in the form <A.B.C.D>.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Functional Notes

The link-state-id differs depending on whether the link state advertisement in question describes a network or a router.

If describing a network, this ID is one of the following:

- The network's IP address. This is true for type 3 summary link advertisements and in autonomous system external link advertisements.
- An address obtained from the link state ID. If the network link advertisement's link state ID is masked with the network's subnet mask, this will yield the network's IP address.

If describing a router, this ID is always the router's OSPF router ID.

Usage Examples

> **enable**
show ip ospf database

show ip ospf interface <*interface type*> <*interface number*>

Use the **show ip ospf interface** command to display OSPF information for a specific interface.

Syntax Description

<*interface type*> **Optional*	Enter the interface type (i.e., **eth, ppp, loopback,** etc.).
<*interface number*> **Optional*	Enter the interface number.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Usage Examples

> **enable**
show ip ospf interface ppp 1

show ip ospf neighbor *<interface type> <interface number> <neighbor id>* [**detail**]

Use the **show ip ospf neighbor** command to display OSPF neighbor information for a specific interface.

Syntax Description

<interface type> **Optional*	Enter the interface type (i.e., **eth, ppp,** etc.).
<interface number> **Optional*	Enter the interface number.
<neighbor id> **Optional*	Enter a specific neighbor's router ID.
detail **Optional*	Enter this keyword to display details on all neighbors.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Usage Examples

> **enable**
show ip ospf neighbor

show ip ospf summary-address

Use the **show ip ospf summary-address** command to display a list of all summary address redistribution information for the system.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Usage Examples

> **enable**
show ip ospf summary-address

show ip policy-class <*policyname*>

Use the **show ip policy-class** command to display a list of currently configured access policies. See *ip policy-class <policyname> max-sessions* on page 208 for information on configuring access policies.

Syntax Description

<*policyname*> **Optional*	Enter a specific policy class name to display information for a single policy.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Usage Examples

The following is a sample output from the **show ip policy-class** command:

```
> enable
# show ip policy-class

ip policy-class max-sessions 0

Policy-class "Trusted":
  0 current sessions (6000 max)
  Entry 1 - allow list MatchAll
```

show ip policy-sessions *<policyname>*

Use the **show ip policy-sessions** command to display a list of current policy class associations. See *ip policy-class <policyname> max-sessions* on page 208 for information on configuring access policies.

Syntax Description

<policyname> **Optional*	Enter a specific policy class name to display information for a single policy.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Usage Examples

> **enable**
show ip policy-sessions

show ip policy-stats *<policyname>*

Use the **show ip policy-stats** command to display a list of current policy class statistics. See *ip policy-class <policyname> max-sessions* on page 208 for information on configuring access policies.

Syntax Description

<policyname> **Optional*	Enter a specific policy class name to display information for a single policy.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Usage Examples

> **enable**
show ip policy-stats

show ip protocols

Use the **show ip protocols** command to display IP routing protocol parameters and statistics.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Usage Examples

The following is a sample output from the **show ip protocols** command:

```
> enable
# show ip protocols
Sending updates every 30 seconds, next due in 8 seconds
  Invalid after 180 seconds, hold down time is 120 seconds
  Redistributing: rip
  Default version control: send version 2, receive version 2
  Interface     Send Ver.  Rec Ver.
    eth 0/1     2          2
    ppp 1       2          2
  Routing for networks:
    1.1.1.0/24
```


show ip route [connected| ospf | rip | static | table]

Use the **show ip route** command to display the contents of the IP route table.

Syntax Description

connected ***Optional**	Displays only the IP routes for directly connected networks.
ospf ***Optional**	Displays only the IP routes associated with OSPF.
rip ***Optional**	Displays only the IP routes that were dynamically learned through RIP.
static ***Optional**	Displays only the IP routes that were statically entered.
table ***Optional**	Displays a condensed version of the IP route table.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following is a sample output from the **show ip route** command:

> **enable**
show ip route rip

Codes: C - connected S - static R - RIP O - OSPF IA - OSPF inter area
N1 - OSPF NSSA external type 1 N2 - OSPF NSSA external type 2
E1 - OSPF external type 1 E2 - OSPF external type 2

Gateway of last resort is 10.200.254.254 to network 0.0.0.0

show ip traffic

Use the **show ip traffic** command to display all IP traffic statistics.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Usage Examples

> **enable**
show ip traffic

show memory heap

Use the **show memory heap** command to display statistics regarding memory including memory allocation and buffer use statistics.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Usage Examples

The following is a sample output from the **show memory heap** command:

> **enable**
show memory heap

Memory Heap:
HeapFree: 2935792
HeapSize: 8522736

Block Managers:

Mgr	Size	Used	Free	Max-Used
0	0	58	0	58
1	16	1263	10	1273
2	48	1225	2	1227
3	112	432	2	434
4	240	140	3	143
5	496	72	2	74
6	1008	76	1	77
7	2032	25	1	26
8	4080	2	1	3
9	8176	31	1	32
10	16368	8	0	8
11	32752	5	1	6
12	65520	3	0	3
13	131056	0	0	0

show output-startup

Use the **show output-startup** command to display startup configuration output line-by-line. This output can be copied into a text file and then used as a configuration editing tool.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Usage Examples

The following is a sample output from the **show output-startup** command:

> **enable**
show output-startup

```
!
#!
#hostname "UNIT_2"
UNIT_2#no enable password
UNIT_2#!
UNIT_2#ip subnet-zero
UNIT_2#ip classless
UNIT_2#ip routing
UNIT_2#!
UNIT_2#event-history on
UNIT_2#no logging forwarding
UNIT_2#logging forwarding priority-level info
UNIT_2#no logging email
etc....
```

show processes cpu

Use the **show processes cpu** command to display information regarding any processes that are currently active.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Usage Examples

The following is a sample output from the **show processes cpu** command:

> **enable**
show processes cpu

```
processes cpu
System load: 7.07%     Min: 0.00%     Max 85.89%
Context switch load: 0.21%
Task  Task                  Invoked  Exec Time    Runtime     Load %
Id    Name          PRI STAT   (count)  (usec)       (usec)      (1sec)
0     Idle            0 W       129689       1971     927923      92.79
1     FrontPanel     249 W         9658        165       3202       0.32
3     Stack Usage     11 W          485        305        325       0.03
4     Q Test 1        10 W           50          4          0       0.00
5     Q Test 2        11 W           50          6          0       0.00
10    Clock           20 W         1443         24         55       0.01
11    PacketRouting  250 W        31656         10       3871       0.39
12    Thread Pool     50 W          161        159          0       0.00
13    IKE             10 W            2        341          0       0.00
14    RouteTableTick  50 W           49        874        874       0.09
....etc.
```

show running-config [verbose | checksum]

Use the **show running-config** command to display a text print of all the non-default parameters contained in the current running configuration file. Use the **verbose** keyword to display a text print of the entire configuration (including parameters in their default state).

Syntax Description

verbose ***Optional**	Using the **verbose** keyword displays the entire running configuration to the terminal screen (versus only the non-default values)
checksum ***Optional**	Using the **checksum** keyword displays the encrypted Message Digest 5 (md5) version of the running configuration.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following is a sample output from the **show running-config** command:

```
> enable
# show running-config
Building configuration...
!
no enable password
!
ip subnet-zero
ip classless
ip routing
!
event-history on
no logging forwarding
logging forwarding priority-level info
no logging email
!
ip policy-timeout tcp all-ports 600
ip policy-timeout udp all-ports 60
ip policy-timeout icmp 60
!
interface eth 0/1.....
```

show snmp

Use the **show snmp** command to display the system Simple Network Management Protocol (SNMP) parameters and current status of SNMP communications.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

> or #	Basic or Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following is an example output using the **show snmp** command for a system with SNMP disabled and the default Chassis and Contact parameters:

> **show snmp**

```
Chassis: Chassis ID
Contact: Customer Service
SNMP logging is DISABLED
0 Rx SNMP packets
  0 Bad community names
  0 Bad community uses
  0 Bad versions
  0 Silent drops
  0 Proxy drops
  0 ASN parse errors
```

show sntp

Use the **show sntp** command to display the system Simple Network Time Protocol (SNTP) parameters and current status of SNTP communications.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

> or #	Basic or Enable Command Mode

Command History

Release 3.1	Command was introduced

Usage Examples

> **show sntp**

show spanning-tree <*bridgegroup#*>

Use the **show spanning-tree** command to display the latest spanning-tree calculations, including priority path cost root path information.

Syntax Description

<*bridgegroup#*> **Optional*	Display spanning-tree for a specific bridge group.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following is an example output using the **show spanning-tree** command:

> **enable**
 # show spanning-tree

```
Bridge group 1 is executing the ieee compatible Spanning Tree protocol
 Bridge Identifier has priority 32768 address 00:A0:C8:08:CA:A2
 Configured hello time 2 max age 20 forward delay 15
 Current root has priority 32768 address 00:A0:C8:00:8F:98
 Root port is 1 (eth 0/1) cost of root path is 100
 Number of topology changes 5 last change occurred 22:31:12 ago
 from fr 1.17
 Times:  hold 1 topology change 35 notification 2
 Port 1 (eth 0/1) of Bridge group 1 is forwarding
   Port path cost 100 Port priority 128 Port Identifier 128.1.
   Designated root has priority 32768 address 00:A0:C8:00:8F:98
   Designated bridge has priority 32768 address 00:A0:C8:00:8F:98
   Designated port id is 128.1 designated path cost 0
   BPDU: sent 0 received 41107
Port 2 (fr 1.16) of Bridge group 1 is down
   Port path cost 1302 Port priority 128 Port Identifier 128.2.
   Designated root has priority 32768 address 00:A0:C8:00:8F:98
   Designated bridge has priority 32768 address 00:A0:C8:08:CA:A2
   Designated port id is 128.2 designated path cost 100
   BPDU: sent 0 received 0
```

Usage Examples (Continued)

```
Port 3 (fr 1.17) of Bridge group 1 is forwarding
  Port path cost 1302 Port priority 128 Port Identifier 128.3.
  Designated root has priority 32768 address 00:A0:C8:00:8F:98
  Designated bridge has priority 32768 address 00:A0:C8:08:CA:A2
  Designated port id is 128.3 designated path cost 100
  BPDU: sent 37014 received 23
```


show startup-config

Use the **show startup-config** command to display a text printout of the startup configuration file stored in NVRAM.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following is a sample output of the **show startup-config** command:

```
> enable
# show startup-config
!
!
no enable password
!
ip subnet-zero
ip classless
ip routing
!
event-history on
no logging forwarding
logging forwarding priority-level info
no logging email
!
ip policy-timeout tcp all-ports 600
ip policy-timeout udp all-ports 60
ip policy-timeout icmp 60
!
!
!
interface eth 0/1
speed auto
  no ip address
  shutdown
```

Usage Examples (Continued)

```
!
interface dds 1/1
  shutdown
!
interface bri 1/2
  shutdown
!
!
ip access-list standard MatchAll
  permit host 10.3.50.6
  permit 10.200.5.0 0.0.0.255
!
!
ip access-list extended UnTrusted
  deny   icmp 10.5.60.0 0.0.0.255  any  source-quench
  deny   tcp any  any
!
no ip snmp agent
!
!
!
line con 0
  no login
!
line telnet 0
  login
line telnet 1
  login
line telnet 2
  login
line telnet 3
  login
line telnet 4
  login
!
```


show tcp info <*control block*>

Use the **show tcp info** command to display TCP control block information in the ADTRAN OS. This information is for troubleshooting and debug purposes only. For more detailed information, you can optionally specify a particular TCP control block. When a particular TCP control block is specified, the system provides additional information regarding crypto map settings that the **show tcp info** command does not display.

Syntax Description

<*control block*> **Optional*	Specify a particular TCP control block for more detailed information.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 4.1	Command was introduced

Usage Examples

The following is a sample from the **show tcp info** command:

> **enable**
#**show tcp info**

TCP TCB Entries

ID	STATE	LSTATE	OSTATE	TYPE	FLAGS	RPORT	LPORT	SWIN	SRT	INTERFACE
0	FREE	FREE	FREE	SRVR	0	0	0	0	0	NONE
1	LISTEN	FREE	FREE	CONN	0	0	21	0	0	NONE
2	LISTEN	FREE	FREE	CONN	0	0	80	0	0	NONE
3	LISTEN	FREE	FREE	CONN	0	0	23	0	0	NONE
4	LISTEN	FREE	FREE	CONN	0	0	5761	0	0	NONE
5	FREE	FREE	FREE	SRVR	0	0	0	0	0	NONE
.										
.										
31	FREE	FREE	FREE	SRVR	0	0	0	0	0	NONE

show version

Use the **show version** command to display the current ADTRAN OS version information.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

> or #	Basic or Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following is a sample **show version** output:

```
>enable
# show version

ADTRAN OS version: 02.01.00
Checksum: 1505165C Built on: Fri Aug 23 10:23:13 2002
 Upgrade key: 420987gacs9097gbdsado
BootROM version: 02.01.00
 Checksum: DB85 Built on: Mon Aug 19 10:33:03 2002
Copyright 1999-2002 ADTRAN Inc.
Serial number b104

Router uptime is 0 days 3 hours 9 minutes 54 seconds
System returned to ROM by External Hard Reset
System image file is "020100.biz"
```

telnet <*address*>

Use the **telnet** command to open a Telnet session (through the ADTRAN OS) to another system on the network.

Syntax Description

<*address*>	Specifies the IP address of the remote system.

Default Values

No default value necessary for this command.

Command Modes

> or #	Basic or Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following example opens a Telnet session with a remote system (**10.200.4.15**):

>**enable**
telnet 10.200.4.15

User Access Login

Password:

traceroute <*address*>

Use the **traceroute** command to display the IP routes a packet takes to reach the specified destination.

Syntax Description

<*address*> **Optional*	Specifies the IP address of the remote system to trace the routes to.

Default Values

No default value necessary for this command.

Command Modes

> or #	Basic or Enable Command Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following is a sample traceroute output:

>**enable**
traceroute 192.168.0.1

Type CTRL+C to abort.
Tracing route to 192.168.0.1 over a maximum of 30 hops

```
  1   22ms   20ms   20ms    192.168.0.65
  2   23ms   20ms   20ms    192.168.0.1
#
```


undebug all

Use the **undebug all** command to disable all activated debug messages.

Syntax Description

No subcommands

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 2.1	Command was introduced

Usage Examples

The following example disabled all activated debug messages:

> **enable**
undebug all

write [erase | memory | network | terminal]

Use the **write** command to save the running configuration to the unit's NVRAM or a TFTP server. Also use the **write** command to clear NVRAM or to display the running configuration on the terminal screen. Entering the **write** command with no other arguments copies your configuration changes to the unit's nonvolatile random access memory (NVRAM). Once the save is complete, the changes are retained even if the unit is shut down or suffers a power outage.

Syntax Description

erase **Optional*	Erase the configuration files saved to the unit's nonvolatile access memory (NVRAM).
memory **Optional*	Save the current configuration to NVRAM. See *copy <source> <destination>* on page 40 for more information.
network **Optional*	Save the current configuration to the network TFTP server. See *copy tftp <destination>* on page 41 for more information.
terminal **Optional*	Display the current configuration on the terminal screen.

Default Values

No default value necessary for this command.

Command Modes

#	Enable Command Mode

Command History

Release 3.1	Command was introduced

Usage Examples

The following example saves the current configuration to the unit's NVRAM:

> **enable**
write memory

GLOBAL CONFIGURATION MODE COMMAND SET

To activate the Global Configuration Mode command set, enter the **configuration** command at the Enable security mode prompt. For example:

```
Router> enable
Router# configure terminal
Router(config)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

All other commands for this command set are described in this section in alphabetical order.

banner [**exec | login | motd**] <*character*> <*message*> <*character*>

Use the **banner** command to specify messages to be displayed in certain situations. Use the **no** form of this command to delete a previously configured banner.

Syntax Description

exec	This command creates a message to be displayed when any exec-level process takes place.
login	This command creates a message to be displayed before the username and password login prompts.
motd	This message creates a message-of-the-day (MOTD) banner.
<*character*>	Banner text delimiter character. Press **Enter** after the delimiter to begin input of banner text.
<*message*>	Enter the text message you wish to display. End with the character that you chose as your delimiter.

Default Values

By default, no banners are configured.

Command Modes

(config)#	Global Configuration Mode

Command History

Release 2.1	Command was introduced

Functional Notes

Banners appear in the following order (if configured):

- MOTD banner appears at initial connection.
- Login banner follows the MOTD banner.
- Exec banner appears after successful log in.

Usage Examples

The following example configures the system to display a message of the day:

(config)# **banner motd *The system will be shut down today from 7PM to 11PM***

boot system flash <*filename*> [**no-backup** | <*backup filename*>]

Use the **boot system flash** command to specify the system image loaded at startup.

Syntax Description

<*filename*>	Specifies the filename (located in flash memory) of the image (filenames are case-sensitive) - image files should have a .biz extension
no-backup	Specify that no backup image is to be saved to the system.
<*backup filename*>	Specify a name for the backup image.

Default Values

No default value necessary for this command.

Command Modes

(config)#	Global Configuration Mode

Command History

Release 1.1	Command was introduced

Functional Notes

Detailed instructions for upgrading the ADTRAN OS and loading files into flash memory are found on the ***NetVanta 3000 Series System Manual*** CD. Please refer to these instructions when upgrading your unit.

Usage Examples

The following example specifies the file “myimage.biz” as the startup image:

(config)# **boot system flash myimage.biz**

bridge <*group#*> **address** <*MAC address*> **discard** <*interface*>

Use the **bridge address discard** command to filter frames on a specific interface that contain the specified hardware address in either the source or destination field. Using the **discard** version of this command limits the processing of unwanted frames on the specified interface. Multiple bridge address commands may be entered for custom filtering applications. Use the **no** form of this command to delete the configured discard filter.

Syntax Description

<*group#*>	Bridge group number (1 to 255) specified using the **bridge-group** command
<*MAC address*>	Unique hardware address (48-bit) in the form of six hexadecimal pairs delimited by either . or : (Example: 00:DE:AD:00:88:28 = 00.DE.AD.00.88.28)
<*interface*>	Defines the interface on which the configured rule will be applied Valid interfaces include: Ethernet (eth 0/1), virtual frame relay interfaces with a specified DLCI (fr 1.16), and virtual PPP interfaces (ppp 1).

Default Values

By default, all packets meeting the configured bridge criteria are forwarded unless otherwise specified.

Command Modes

(config)#	Global Configuration Mode

Command History

Release 1.1	Command was introduced

Functional Notes

Creating custom filter schemes for bridged networks limits the amount of unnecessary traffic processed and distributed by the bridging equipment. Use multiple bridge address discard commands to develop the filter scheme.

Usage Examples

The following example discards all packets from (or to) a host with the MAC address 00:DE:AD:00:88:28 destined for bridge group 17 on the ethernet 0/1 interface:

(config)# **bridge 17 address 00:DE:AD:00:88:28 discard eth 0/1**

Technology Review

A bridged network can provide excellent traffic management to reduce collisions and limit the amount of bandwidth wasted with unnecessary transmissions. Transparent bridges working at Layer 2 (of the OSI Model) forward packets on network segments only when it is necessary. The decision process involves only Layer 2 hardware addresses. Conversely, routers operate at Layer 3, using network addresses and route tables to make packet forwarding decisions.

bridge <*group#*> **address** <*MAC address*> **forward** <*interface*>

Use the **bridge address forward** command to filter frames on a specific interface that contain the specified hardware address in either the source or destination field. Using the **forward** version of this command predefines the path for specified packets. Multiple bridge address commands may be entered for custom filtering applications. Use the **no** form of this command to delete the configured forward filter.

Syntax Description

<*group#*>	Bridge group number (1 to 255) specified using the **bridge-group** command
<*MAC address*>	Unique hardware address (48-bit) in the form of six hexadecimal pairs delimited by either . or : (Example: 00:DE:AD:00:88:28 = 00.DE.AD.00.88.28)
<*interface*>	Defines the interface on which the configured rule will be applied Valid interfaces include: Ethernet (eth 0/1), virtual frame relay interfaces with a specified DLCI (fr 1.16), and virtual PPP interfaces (ppp 1).

Default Values

By default, all packets meeting the configured bridge criteria are forwarded unless otherwise specified.

Command Modes

(config)#	Global Configuration Mode

Command History

Release 1.1	Command was introduced

Functional Notes

Creating custom filter schemes for bridged networks limits the amount of unnecessary traffic processed and distributed by the bridging equipment. Use multiple bridge address forward commands to develop the filter scheme.

Usage Examples

The following example forwards all packets from (or to) a host with the MAC address 00:DE:AD:00:88:28 destined for bridge group 17 on the ethernet 0/1 interface:

(config)# **bridge 17 address 00:DE:AD:00:88:28 forward eth 0/1**

Technology Review

A bridged network can provide excellent traffic management to reduce collisions and limit the amount of bandwidth wasted with unnecessary transmissions.

bridge <*group#*> **aging-time** <*seconds*>

Use the **bridge aging-time** command to set the length of time a dynamic entry may remain in the bridge table from the time the entry was created or last updated. To return to the default aging-time interval, use the **no** form of this command.

Syntax Description

<*group#*>	Bridge group number (1 to 255) specified using the **bridge-group** command
<*seconds*>	Number of seconds the entry will live (valid range: 0 to 1000000)

Default Values

<*seconds*>	300 seconds

Command Modes

(config)#	Global Configuration Mode

Command History

Release 1.1	Command was introduced

Functional Notes

When bridged networks are relatively static (with very few moving hosts), increasing the aging time value may reduce the possibility of flooding unnecessary network traffic to all subnets. If hosts on the network frequently change, decreasing the aging time value allows the bridge to adapt to the changing network configuration.

Usage Examples

The following example configures an aging time of 450 seconds for bridge 17:

(config)# **bridge 17 aging-time 450**

Technology Review

A bridged network can provide excellent traffic management to reduce collisions and limit the amount of bandwidth wasted with unnecessary transmissions. Configuring an appropriate aging time (allowing the bridge forwarding table to remain updated) reduces the number of flooded broadcast messages.

bridge <*group#*> **forward-time** <*seconds*>

Use the **bridge forward-time** command to specify the delay interval (in seconds) when forwarding bridge packets. Use the **no** form of this command to return to the default interval. The default value may be restored using the **no** form of this command.

Syntax Description

<*group#*>	Bridge group number (1 to 255) specified using the **bridge-group** command
<*seconds*>	Forward delay interval in seconds (valid range: 4 to 20)

Default Values

<*seconds*>	15 seconds

Command Modes

(config)#	Global Configuration Mode

Command History

Release 1.1	Command was introduced

Functional Notes

When a new bridging interface is added to a network topology, change information is broadcasted and used to update spanning-tree calculations. The forward time interval specifies the number of seconds a configured bridge will wait (and listen) for the topology update information before forwarding packets to the new interface. If multiple forward time intervals exist within a bridge group, the entire bridge group follows the forward time interval specified in the root bridge.

Usage Examples

The following example configures a forward time interval of 15 seconds for bridge 17:

(config)# **bridge 17 forward-time 15**

Technology Review

A bridged network can provide excellent traffic management to reduce collisions and limit the amount of bandwidth wasted with unnecessary transmissions. Configuring an appropriate forward time delay interval (allowing the bridge to wait for the new host information) reduces the amount of information requests transmitted on the interfaces while maintaining a current forwarding table.

bridge <*group#*> **hello-time** <*seconds*>

Use the **bridge hello-time** command to specify the delay interval (in seconds) between hello bridge protocol data units (BPDUs). To return the default interval, use the **no** form of this command.

Syntax Description

<*group#*>	Bridge group number (1 to 255) specified using the **bridge** command
<*seconds*>	Delay interval (in seconds) between hello BPDUs (valid range: 0 to 1000000)

Default Values

<*seconds*>	2 seconds

Command Modes

(config)#	Global Configuration Mode

Command History

Release 1.1	Command was introduced

Functional Notes

A bridged network can provide excellent traffic management to reduce collisions and limit the amount of bandwidth wasted with unnecessary transmissions. Configuring an appropriate hello-time interval helps reduce the number of Bridge Protocol Data Units (BPDUs) transmitted on the network.

Usage Examples

The following example configures a hello-time interval of 10000 seconds for bridge-group 17:

(config)# **bridge 17 hello-time 10000**

bridge <*group#*> **max-age** <*seconds*>

Use the **bridge max-age** command to specify the interval (in seconds) the bridge will wait to receive Bridge Protocol Data Units (BPDUs) from the root bridge before assuming the network has changed, thus re-evaluating the spanning-tree topology. Use the **no** form of this command to return to the default interval.

Syntax Description

<*group#*>	Bridge group number (1 to 255) specified using the **bridge** command
<*seconds*>	Wait interval (in seconds) between received BPDUs (from the root bridge) (valid range: 6 to 200)

Default Values

<*seconds*>	20 seconds

Command Modes

(config)#	Global Configuration Mode

Command History

Release 1.1	Command was introduced

Functional Notes

A bridged network can provide excellent traffic management to reduce collisions and limit the amount of bandwidth wasted with unnecessary transmissions. Configuring an appropriate max-age interval helps reduce the number of Bridge Protocol Data Units (BPDUs) transmitted on the network.

Usage Examples

The following example configures a max-age interval of 45 seconds for bridge-group 17:

(config)# **bridge 17 max-age 45**

bridge <*group#*> **priority** <*value*>

Use the **bridge priority** command to set the priority for bridging interfaces in the specified bridge group. The lower the priority value, the higher the likelihood the configured bridge interface will be the root for the bridge group. To return to the default bridge priority value, use the **no** version of this command.

Syntax Description

<*group#*>	Bridge group number (1 to 255) specified using the **bridge-group** command
<*value*>	Priority value for the bridge interface. Configuring this value to a low number increases the interface's chance of being the root. Therefore, the maximum priority level would be 0.

Default Values

<*value*>	32768

Command Modes

(config)#	Global Configuration Mode

Command History

Release 1.1	Command was introduced

Functional Notes

A root bridge is necessary for each bridging physical interface for the spanning-tree to be computed. The specified value is inversely proportional to the likelihood the bridge interface will be the root path. Set the priority value lower to increase the chance the interface will be the root.

Usage Examples

The following example globally sets the maximum priority for the bridge interface 1:

(config)# **bridge 1 priority 0**

Technology Review

Priority values are used by the spanning-tree protocol when analyzing a bridged network for root and redundant paths. In addition to priority values, spanning-tree protocol uses assigned path costs to identify and rank valid data paths.

bridge <*group#*> protocol ieee

The **bridge protocol ieee** command configures a bridge group for the IEEE Spanning Tree Protocol. Use the **no** form of this command (with the appropriate arguments) to delete this setting.

Syntax Description

<*group#*>	Bridge group number (1 to 255) specified using the bridge command
ieee	IEEE 802.1 Ethernet spanning-tree protocol

Default Values

*By default, all configured bridge interfaces implement **ieee** spanning-tree protocol.*

Command Modes

(config)#	Global Configuration Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following example deletes the bridge protocol setting for bridge-group 17:

(config)# **no bridge 17 protocol ieee**

cross-connect *<#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>*

Use the **cross-connect** command to create a cross-connect map from a created tdm-group on an interface to another physical or virtual interface.

*Changing **cross-connect** settings could potentially result in service interruption.*

Syntax Description

<#>	Number descriptor or label for identifying the cross-connect (useful in systems that allow multiple cross connects)
<from interface>	Specifies the interface (physical of virtual) on one end of the cross-connect Valid interfaces include: T1 (t1 1/1), DDS (dds 1/1), serial (ser 1/1), and shdsl (shdsl 1/1)
<slot/port>	Used when a physical interface is specified in the <from interface> subcommand (For example: specifying the T1 port of a T1 module would be t1 1/1).
<tdm-group#>	Specifies which configured tdm-group to use for this cross-connect. This subcommand only applies to T1 physical interfaces.
<to interface>	Specifies the interface (physical or virtual) on the other end of the cross-connect.
<slot/port>	Used when a physical interface is specified in the <to interface> subcommand. (For example, specifying the primary T1 port of a T1 module would be t1 1/1).

Default Values

By default, there are no configured cross-connects.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

Cross-connects provide the mechanism for connecting a configured virtual (layer 2) endpoint with a physical (layer 1) interface. Supported layer 2 protocols include frame relay and point-to-point protocol (PPP).

Usage Examples

The following example creates a frame relay endpoint and connects it to the t1 1/1 physical interface:

1. Create the frame relay virtual endpoint and set the signaling method:

(config)# **interface frame-relay 1**
(config-fr 1)# **frame-relay lmi-type cisco**

2. Create the sub-interface and configure the PVC parameters (including DLCI and IP address):

(config-fr 1)# **interface fr 1.1**
(config-fr 1.1)# **frame-relay interface-dlci 17**
(config-fr 1.1)# **ip address 168.125.33.252 255.255.255.252**

3. Create the tdm-group of 12 DS0s (64K) on the t1 physical interface:
(THIS STEP IS ONLY VALID FOR T1 INTERFACES.)

(config)# **interface t1 1/1**
(config-t1 1/1)# **tdm-group 1 timeslots 1-12 speed 64**
(config-t1 1/1)# **exit**

4. Connect the frame relay sub-interface with port t1 1/1:

(config)# **cross-connect 1 t1 1/1 1 fr 1**

Technology Review

Creating an endpoint that uses a layer 2 protocol (such as frame relay) is generally a four-step process:

Step 1:
Create the frame relay virtual endpoint (using the **interface frame-relay** command) and set the signaling method (using the **frame-relay lmi-type** command). Also included in the frame relay virtual endpoint command set are all the applicable frame relay timers logging thresholds, encapsulation types, etc. Generally, most frame relay virtual interface parameters should be left at their default state. For example, the following creates a frame relay interface labeled **7** and sets the signaling method to **ansi**.

(config)# **interface frame-relay 7**
(config-fr 7)# **frame-relay lmi-type ansi**

Step 2:
Create the sub-interface and configure the PVC parameters. Using the sub-interface command set, apply access policies to the interface, create bridging interfaces, configure dial-backup, assign an IP address, and set the PVC data-link control identifier (DLCI). For example, the following creates a frame relay sub-interface labeled **22,** sets the DLCI to **30,** and assigns an IP address of **193.44.69.253** to the interface.

(config-fr 7)# **interface fr 7.22**
(config-fr 7.22)# **frame-relay interface-dlci 30**
(config-fr 7.22)# **ip address 193.44.69.253 255.255.255.252**

Technology Review (Continued)

Step 3: (VALID ONLY FOR T1 INTERFACES)
Specify the group of DS0s used for signaling on the T1 interface by creating a **tdm-group**. Group any number of aggregate DS0s together to create a data pipe for layer 2 signaling. Also use the **tdm-group** command to specify the per-DS0 signaling rate on the interface. For example, the following creates a tdm-group labeled **9** containing 20 DS0s (each DS0 having a data rate of 56 kbps).

(config)# **interface t1 1/1**
(config-t1 1/1)# **tdm-group 9 timeslots 1-20 speed 56**
(config-t1 1/1)# **exit**

Step 4:
Make the association between the layer 2 endpoint and the physical interface using the **cross-connect** command. Supported layer 2 protocols include frame relay and point-to-point protocol (PPP). For example, the following creates a cross-connect (labeled **5**) to make an association between the frame relay virtual interface (**fr 7**) and the tdm-group configured on interface t1 1/1 (**tdm-group 9**).

(config)# **cross-connect 5 t1 1/1 9 fr 7**

crypto ike

Use the **crypto ike** command to define the system-level local ID for IKE negotiations and to enter the IKE Client or IKE Policy command sets.

Variations of this command include the following:
crypto ike client configuration pool <*poolname*>
crypto ike local-id address
crypto ike policy <*policy priority*>

Syntax Description

client configuration pool <*poolname*>	Creates a local pool named the <*poolname*> of your choice and enters the IKE Client command set. Clients that connect via an IKE policy that specifies this pool-name will be assigned values from this pool. See the section *IKE Client Command Set* on page 266 for more information.
local-id address	Sets the local ID during IKE negotiation to be the IP address of the interface from which the traffic exits. This setting can be overridden on a per-policy basis using the **local-id** command in the IKE Policy command set (see *local-id [address | fqdn | user-fqdn] <ipaddress or domain name>* on page 261 for more information).
policy <*policy priority*>	Creates an IKE policy with the <*policy priority*> of your choice and enters the IKE Policy command set. See *IKE Policy Command Set* on page 254 for more information.

Default Values

There are no default settings for this command.

Command Modes

(config)#	Global Configuration Mode

Command History

Release 4.1	Command was introduced

Usage Examples

The following example creates an IKE policy with a policy priority setting of 1 and enters the IKE Policy command set for that policy:

(config)# **crypto ike policy 1**
(config-ike)#

Technology Review

The following example configures an ADTRAN OS product for VPN using IKE aggressive mode with pre-shared keys. The ADTRAN OS product can be set to initiate IKE negotiation in main mode or aggressive mode. The product can be set to respond to IKE negotiation in main mode, aggressive mode, or any mode. In this example, the device is configured to initiate in aggressive mode and to respond to any mode.

This example assumes that the ADTRAN OS product has been configured with a WAN IP Address of 192.168.1.1 on interface **ethernet 0/1** and a LAN IP Address of 10.10.10.254 on interface **ethernet 0/2**. The Peer Private IP Subnet is 10.10.20.0.

For more detailed information on VPN configuration, refer to the technical support note *Configuring VPN* located on the ***NetVanta 3000 Series System Manual*** CD provided with your unit.

Step 1:
Enter the Global configuration mode (i.e., config terminal mode).

```
>enable
#configure terminal
```

Step 2:
Enable VPN support using the **ip crypto** command. This command allows crypto maps to be applied to interfaces, and enables the IKE server to listen for IKE negotiation sessions on UDP port 500.

```
(config)#ip crypto
```

Step 3:
Set the local ID. During IKE negotiation, local-ids are exchanged between the local device and the peer device. In the ADTRAN OS, the default setting for all local-ids is configured by the **crypto ike local-id** command. The default setting is for all local-ids to be the IPv4 address of the interface over which the IKE negotiation is occurring. In the future, a unique system-wide Hostname or Fully Qualified Domain Name could be used for all IKE negotiation.

```
(config)#crypto ike local-id address
```

Step 4:
Create IKE policy. In order to use IKE negotiation, an IKE policy must be created. Within the system, a list of IKE policies is maintained. Each IKE policy is given a priority number in the system. That priority number defines the position of that IKE policy within the system list. When IKE negotiation is needed, the system searches through the list, starting with the policy with priority of 1, looking for a match to the peer IP address.

An individual IKE policy can override the system local-id setting by having the **local-id** command specified in the IKE policy definition. This command in the IKE policy is used to specify the type of local-id and the local-id data. The type can be of IPv4 address, Fully Qualified Domain Name, or User-Specified Fully Qualified Domain Name.

An IKE policy may specify one or more peer IP addresses that will be allowed to connect to this system. To specify multiple unique peer IP addresses, the **peer A.B.C.D** command is used multiple times within a single IKE policy. To specify that all possible peers can use a default IKE policy, the **peer any** command is given instead of the **peer A.B.C.D** command inside of the IKE policy. The policy with the **peer any** command specified will match to any peer IP address (and therefore should be given the highest numerical priority number). This will make the policy the last one to be compared against during IKE negotiation.

Technology Review (Continued)

```
(config)#crypto ike policy 10
(config-ike)#no local-id
(config-ike)#peer 192.168.1.2
(config-ike)#initiate aggressive
(config-ike)#respond anymode
(config-ike)#attribute 10
(config-ike-attribute)#encryption 3des
(config-ike-attribute)#hash sha
(config-ike-attribute)#authentication pre-share
(config-ike-attribute)#group 1
(config-ike-attribute)#lifetime 900
```

Step 5:
Define the remote-id settings. The **crypto ike remote-id** command is used to define the remote-id for a peer connecting to the system, specify the preshared-key associated with the specific remote-id, and (optionally) determine that the peer matching this remote-id should not use mode config (by using the **no-mode-config** keyword). See *crypto ike remote-id* on page 154 for more information.

```
(config)#crypto ike remote-id address 192.168.1.2 preshared-key
mysecret123
```

Step 6:
Define the transform-set. A transform-set defines the encryption and/or authentication algorithms to be used to secure the data transmitted over the VPN tunnel. Multiple transform-sets may be defined in a system. Once a transform-set is defined, many different crypto maps within the system can reference it. In this example, a transform-set named **highly_secure** has been created. This transform-set defines ESP with Authentication implemented using 3DES encryption and SHA1 authentication.

```
(config)#crypto ipsec transform-set highly_secure esp-3des esp-sha-hmac
(cfg-crypto-trans)#mode tunnel
```

Step 7:
Define an ip-access list. An Extended Access Control List is used to specify which traffic needs to be sent securely over the VPN tunnel. The entries in the list are defined with respect to the local system. The source IP address will be the source of the traffic to be encrypted. The destination IP address will be the receiver of the data on the other side of the VPN tunnel.

```
(config)#ip access-list extended corporate_traffic
(config-ext-nacl)#permit ip 10.10.10.0 0.0.0.255 10.10.20.0 0.0.0.255 log
  deny ip any any
```

Step 8:
Create crypto map. A Crypto Map is used to define a set of encryption schemes to be used for a given interface. A crypto map entry has a unique index within the crypto map set. The crypto map entry will specify whether IKE is used to generate encryption keys or if manually specified keys will be used. The crypto map entry will also specify who will be terminating the VPN tunnel, as well as which transform-set or sets will be used to encrypt and/or authenticate the traffic on that VPN tunnel. It also specifies the lifetime of all created IPSec Security Associations.

```
(config)#crypto map corporate_vpn 1 ipsec-ike
(config-crypto-map)#match address corporate_traffic
(config-crypto-map)#set peer 192.168.1.2
(config-crypto-map)#set transform-set highly_secure
(config-crypto-map)#set security-association lifetime kilobytes 8000
(config-crypto-map)#set security-association lifetime seconds 86400
(config-crypto-map)#no set pfs
```

Technology Review (Continued)

Step 9:
Configure public interface. This process includes configuring the IP address for the interface and applying the appropriate crypto map to the interface. Crypto maps are applied to the interface on which encrypted traffic will be transmitted.

```
(config)#interface ethernet 0/1
(config-eth 0/1)#ip address 192.168.1.1 255.255.255.0
(config-eth 0/1)#crypto map corporate_vpn
(config-eth 0/1)#no shutdown
```

Step 10:
Configure private interface and add a static route to allow all traffic destined for the VPN tunnel to be routed to the appropriate gateway.

```
(config)#interface ethernet 0/2
(config-eth 0/2)#ip address 10.10.10.254 255.255.255.0
(config-eth 0/2)#no shutdown
(config-eth 0/2)#exit
(config)#ip route 10.10.20.0 255.255.255.0 192.168.1.2
```

crypto ike remote-id

Use the **crypto ike remote-id** command to specify the remote ID and to associate a pre-shared key with the remote ID.

Variations of this command include the following:
crypto ike remote-id address <*IPv4 address*> [no-mode-config]
crypto ike remote-id address <*IPv4 address*> preshared-key <*keyname*> [no-mode-config]
crypto ike remote-id fqdn <*fqdn*> [no-mode-config]
crypto ike remote-id fqdn <*fqdn*> preshared-key <*keyname*> [no-mode-config]
crypto ike remote-id user-fqdn <*fqdn*> [no-mode-config]
crypto ike remote-id user-fqdn <*fqdn*> preshared-key <*keyname*> [no-mode-config]

For VPN configuration example scripts, refer to the technical support note ***Configuring VPN*** *located on the* ***NetVanta 3000 Series System Manual*** *CD provided with your unit.*

Syntax Description

address <*IPv4 address*>	Specifies a remote ID of IPv4 type.
fqdn <*fqdn*>	Specifies a fully qualified domain name (e.g., adtran.com) as the remote ID.
user-fqdn <*fqdn*>	Specifies a user fully qualified domain name (e.g., user1@adtran.com) as the remote ID.
preshared-key <*keyname*>	Associates a pre-shared key with this remote ID.
no-mode-config	Optional keyword used to specify that the peer matching this remote-id should not use mode config.

Default Values

There are no default settings for this command.

Command Modes

(config)#	Global Configuration Mode

Command History

Release 4.1	Command was introduced

Usage Examples

The following example assigns a remote ID of 192.168.1.1 and associates the pre-shared key **mysecret** with the remote ID:

(config)# **crypto ike remote-id address 192.168.1.1 preshared-key mysecret**

crypto ipsec transform-set *<setname> <parameters>*

Use the **crypto ipsec transform-set** command to define the transform configuration for securing data (e.g., esp-3des, esp-sha-hmac, etc.). The transform-set is then assigned to a crypto map using the map's **set transform-set** command. See *set transform-set <setname1 - setname6>* on page 282.

*For VPN configuration example scripts, refer to the technical support note **Configuring VPN** located on the **NetVanta 3000 Series System Manual** CD provided with your unit.*

Syntax Description

<setname>	Assign a name to the transform-set you are about to define.
<parameters>	Assign a combination of up to three security algorithms. This field is a valid combination of the following: • ah-md5-hmac, ah-sha-hmac • esp-des, esp-3des, esp-aes-128-cbc, esp-aes-192-cbc, esp-aes-256-cbc, esp-null • esp-md5-hmac, esp-sha-hmac

Default Values

There are no default settings for this command.

Command Modes

(config)#	Global Configuration Mode

Command History

Release 4.1	Command was introduced

Functional Notes

Crypto map entries do not directly contain the transform configuration for securing data. Instead, the crypto map is associated with transform sets which contain specific security algorithms.

If no transform-set is configured for a crypto map, the entry is incomplete and will have no effect on the system.

Usage Examples

The following example first creates a transform-set (**Set1**) consisting of two security algorithms (up to three may be defined), and then assigns the transform-set to a crypto map (**Map1**):

(config)#**crypto ipsec transform-set Set1 esp-3des esp-sha-hmac**
(cfg-crypto-trans)#**exit**

(config)#**crypto map Map1 1 ipsec-ike**
(config-crypto-map)#**set transform-set Set1**

crypto map

Use the **crypto map** command to define crypto map names and numbers and to enter the associated command set (either Crypto Map IKE or Crypto Map Manual).

Variations of this command include the following:
crypto map <*mapname*> <*mapindex*> ipsec-ike
crypto map <*mapname*> <*mapindex*> ipsec-manual

*For VPN configuration example scripts, refer to the technical support note **Configuring VPN** located on the **NetVanta 3000 Series System Manual** CD provided with your unit.*

Syntax Description

<*mapname*>	Name the crypto map. You can assign the same name to multiple crypto maps, as long as the map index numbers are unique.
<*mapindex*>	Assign a crypto map sequence number.
ipsec-ike	Enter the Crypto Map IKE command set (see *Crypto Map IKE Command Set* on page 276). This command set supports IPSec entries that will use IKE to negotiate keys.
ipsec-manual	Enter the Crypto Map Manual command set (see *Crypto Map Manual Command Set* on page 283). This command set supports manually configured IPSec entries.

Default Values

There are no default settings for this command.

Command Modes

(config)#	Global Configuration Mode

Command History

Release 4.1	Command was introduced

Functional Notes

Crypto map entries do not directly contain the transform configuration for securing data. Instead, the crypto map is associated with transform sets which contain specific security algorithms (see *crypto ipsec transform-set <setname> <parameters>* on page 155).

Crypto map entries do not directly contain the selectors used to determine which data to secure. Instead, the crypto map entry refers to an access control list. An access control list is assigned to the crypto map using the **match address** command (see *match address <listname>* on page 277).

If no transform-set or access-list is configured for a crypto map, the entry is incomplete and will have no effect on the system.

When you apply a crypto map to an interface (using the **crypto map** command within the interface's command set), you are applying all crypto maps with the given map name. This allows you to apply multiple crypto maps if you have created maps which share the same name but have different map index numbers.

Usage Examples

The following example creates a new IPSec IKE crypto map called **testMap** with a map index of **10**:

(config)# **crypto map testMap 10 ipsec-ike**
(config-crypto-map)#

Technology Review

A crypto map entry is a single policy that describes how certain traffic is to be secured. There are two types of crypto map entries: ipsec-manual and ipsec-ike. Each entry is given an index, which is used to sort the ordered list. When a non-secured packet arrives on an interface, the crypto map set associated with that interface is processed in order. If a crypto map entry matches the non-secured traffic, the traffic is discarded.

When a packet is to be transmitted on an interface, the crypto map set associated with that interface is processed in order. The first crypto map entry that matches the packet will be used to secure the packet. If a suitable SA (security association) exists, that is used for transmission. Otherwise, IKE is used to establish an SA with the peer. If no SA exists, and the crypto map entry is "respond only", the packet is discarded.

When a secured packet arrives on an interface, its SPI (security parameter index) is used to look up an SA. If an SA does not exist, or if the packet fails any of the security checks (bad authentication, traffic does not match SA selectors, etc.), it is discarded. If all checks pass, the packet is forwarded normally.

enable password [**md5**] *<password>*

Use the **enable password** command to define a password (with optional encryption) for accessing the Enable Mode command set. Use the **no enable password** command to remove a configured password.

To prevent unauthorized users from accessing the configuration functions of your AOS device, immediately install an Enable-level password.

Syntax Description

md5 ***Optional*	Specifies Message Digest 5 (md5) as the encryption protocol to use when displaying the enable password during **show** commands. If the **md5** keyword is not used, encryption is not used when displaying the enable password during **show** commands
<password>	String (up to 30 characters in length) to use as the Enable Security Mode password.

Default Values

By default, there is no configured enable password.

Command Modes

#	Enable Security Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

To provide extra security, the ADTRAN OS can encrypt the enable password when displaying the current configuration. For example, the following is a **show configuration** printout (password portion) with an unencrypted enable password (ADTRAN):

```
!
enable password ADTRAN
!
```

Alternately, the following is a **show configuration** printout (password portion) with an enable password of ADTRAN using md5 encryption:

```
!
enable password md5 encrypted 5aa5fbae7d01a90e79fb57705ce74676
!
```

event-history on

Use the **event-history on** command to enable event logging for the ADTRAN OS system. Event log messages will not be recorded unless this command has been issued (regardless of the **event-history priority** configured). The event log may be displayed using the **show event-history** command. Use the **no** form of this command to disable the event log.

Syntax Description

No subcommands

Default Values

By default, the ADTRAN OS event logging capabilities are disabled.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

The event history provides useful information regarding the status of the system and individual port states. Use the event history as a troubleshooting tool when identifying system issues. The following is a sample event history log.

Router# **show event-history**
Using 526 bytes
2002.07.12 15:34:01 T1.t1 1/1 Yellow
2002.07.12 15:34:01 INTERFACE_STATUS.t1 1/1 changed state to down.
2002.07.12 15:34:02 T1.t1 1/1 No Alarms
2002.07.12 15:34:02 INTERFACE_STATUS.t1 1/1 changed state to up.
2002.07.12 15:34:03 INTERFACE_STATUS.eth 0/1 changed state to up.
2002.07.12 15:34:10 OPERATING_SYSTEM Warm Start
2002.07.12 15:34:12 PPP.NEGOTIATION LCP up
2002.07.12 15:34:12 PPP.NEGOTIATION IPCP up

Usage Examples

The following example enables the ADTRAN OS event logging feature:
(config)# **event-history on**

event-history priority *<priority>*

Use the **event-history priority** command to set the threshold for events stored in the event history. All events with the specified priority or higher will be kept for viewing in the local event log. The event log may be displayed using the **show event-history** command. Use the **no** form of this command to keep specified priorities from being logged.

Syntax Description

<priority>	Sets the minimum priority threshold for logging messages to the event history The following priorities are available (ranking from lowest to highest): **4 - Information** When priority 4 is selected, all events (priorities 0 through 4) are logged. **3 - Notice** When priority 3 is selected, events with priority 3, 2, 1, or 0 are logged. **2 - Warning** When priority 2 is selected, events with priority 2, 1, or 0 are logged. **1 - Error** When priority 1 is selected, events with priority 1 or 0 are logged. **0 - Fatal** When priority 0 is selected, only events with priority 0 are logged.

Default Values

By default, no event messages are logged to the event history.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

The event history provides useful information regarding the status of the system and individual port states. Use the event history as a troubleshooting tool when identifying system issues. The following is a sample event history log.

Router# **show event-history**
Using 526 bytes

2002.07.12 15:34:01 T1.t1 1/1 Yellow
2002.07.12 15:34:01 INTERFACE_STATUS.t1 1/1 changed state to down.
2002.07.12 15:34:02 T1.t1 1/1 No Alarms
2002.07.12 15:34:02 INTERFACE_STATUS.t1 1/1 changed state to up.
2002.07.12 15:34:03 INTERFACE_STATUS.eth 0/1 changed state to up.
2002.07.12 15:34:10 OPERATING_SYSTEM Warm Start
2002.07.12 15:34:12 PPP.NEGOTIATION LCP up
2002.07.12 15:34:12 PPP.NEGOTIATION IPCP up

Usage Examples

The following example logs all events to the event history (specifying priority 4):

(config)# **event-history priority 4**

hostname *<name>*

Creates a name used to identify the unit. This alphanumeric string should be used as a unique description for the unit. This string will be displayed in all prompts.

Syntax Description

<name>	Alphanumeric string up to 32 characters used to identify the unit

Default Values

<name>	Router

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example creates a hostname for the AOS device of **ATL_RTR** to identify the system as the Atlanta router:

```
(config)# hostname ATL_RTR
ATL_RTR(config)#
```

interface <*port-type*> <*slot/port*>

Activates the Interface Configuration Mode for the listed physical interface.

Syntax Description

<*port-type*>	Identifies the physical port type of the installed Network Interface Module (NIM) Dial-Backup Interface Module (DIM) or Ethernet port. Type **interface ?** for a complete list of valid interfaces.
<*slot/port*>	Specifies an interface based on its physical location (slot and port). For example, if you have a T1/DSX-1 NIM installed in Slot 1 of an AOS product: • The **WAN-T1** port would be specified in the CLI as **t1 1/1**. • The DSX-1 port would be specified as **t1 1/2**. • If (for example) a BRI DIM is also installed, then the **DBU** port of the NIM card would be specified as **bri 1/3**. If you are specifying a port that is built into the base unit (e.g., the Ethernet port), the slot number is **0**. For example, the Ethernet (**LAN**) port would be specified as **ethernet 0/1**.

Default Values

No default values required for this command.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example enters the serial interface command set for a serial module installed in slot 1:

(config)# **interface serial 1/1**

interface frame-relay <*label*> point-to-point

Use the **interface frame-relay** command to create a virtual frame relay interface (or sublink, if specified) that is identified using the entered number label. In addition, entering this command activates the frame relay interface command set. The **point-to-point** keyword (optional) can be used to identify the frame relay endpoint as a point-to-point link (versus multipoint). Use the **no** form of this command to delete a configured virtual frame relay interface.

To specify a virtual frame relay sub-interface, the following syntax applies:

interface frame-relay <*label*>**.**<*sublink label*>

Syntax Description

<*label*>	Specifies the numerical virtual frame relay interface identifying label (valid range: 1 to 1024)
<*sublink label*>	Numerical label for the virtual sublink (valid range: 1-255)
point-to-point ***Optional**	Identifies the frame relay interface as a point-to-point link (versus multilink) By default, all created frame relay interfaces are point-to-point.

Default Values

By default, there are no configured virtual frame relay interfaces or sublinks.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

Creating an endpoint that uses a layer 2 protocol (such as frame relay) is generally a four-step process:

Step 1:
Create the frame relay virtual endpoint (using the **interface frame-relay** command) and set the signaling method (using the **frame-relay lmi-type** command). Also included in the frame relay virtual endpoint command set are all the applicable frame relay timers, logging thresholds, encapsulation types, etc. Generally, most frame relay virtual interface parameters should be left at their default state. For example, the following creates a frame relay interface labeled **7** and sets the signaling method to **ansi**.

(config)# **interface frame-relay 7**
(config-fr 7)# **frame-relay lmi-type ansi**

Functional Notes (Continued)

Step 2:
Create the sub-interface and configure the PVC parameters. Using the sub-interface command set, apply access policies to the interface, create bridging interfaces, configure dial-backup, assign an IP address, and set the PVC data-link control identifier (DLCI). For example, the following creates a frame relay sub-interface labeled **22,** sets the DLCI to **30,** and assigns an IP address of **193.44.69.1/30** to the interface.

(config-fr 7)# **interface fr 7.22**
(config-fr 7.22)# **frame-relay interface-dlci 30**
(config-fr 7.22)# **ip address 193.44.69.1 255.255.255.252**

Step 3: (VALID ONLY FOR T1 INTERFACES)
Specify the group of DS0s used for signaling on the T1 interface by creating a **tdm-group**. Group any number of aggregate DS0s together to create a data pipe for layer 2 signaling. Also use the **tdm-group** command to specify the per-DS0 signaling rate on the interface. For example, the following creates a tdm-group labeled **9** containing 20 DS0s (each DS0 having a data rate of 56 kbps).

(config)# **interface t1 1/1**
(config-t1 1/1)# **tdm-group 9 timeslots 1-20 speed 56**
(config-t1 1/1)# **exit**

Step 4:
Make the association between the layer 2 endpoint and the physical interface using the **cross-connect** command. Supported layer 2 protocols include frame relay and point-to-point protocol (PPP). For example, the following creates a cross-connect (labeled **5**) to make an association between the frame relay virtual interface (**fr 7**) and the tdm-group configured on interface t1 1/1 (**tdm-group 9**).

(config)# **cross-connect 5 t1 1/1 9 fr 7**

Usage Examples

The following example creates a frame relay virtual interface (labeled **1**) and enters the Frame Relay Interface Configuration Mode:

(config)# **interface fr 1**
(config-fr 1)#

interface loopback *<label>*

Use the **interface loopback** command to create a virtual interface that can be assigned layer 3 and higher properties and is always up unless the router is shut down. Use the **no** form of this command to delete a configured loopback interface.

Syntax Description

<label>	Specifies the numerical virtual loopback interface identifying label (valid range: 1 to 1024)

Default Values

By default, there are no configured loopback interfaces.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 3.1	Command was introduced

Usage Examples

The following example creates a loopback virtual interface (labeled 1) and enters the Loopback Interface Configuration Mode:

```
(config)# interface loopback 1
(config-loop 1)#
```

interface ppp *<label>*

Use the **interface ppp** command to create a virtual point-to-point protocol (PPP) interface that is identified using the entered number label. In addition, entering this command activates the PPP interface command set. Use the **no** form of this command to delete a configured virtual PPP interface.

Syntax Description

<label>	Specifies the numerical virtual PPP interface identifying label (valid range: 1 to 1024)

Default Values

By default, there are no configured PPP interfaces.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

Creating an endpoint that uses a layer 2 protocol (such as PPP) is generally a four-step process:

Step 1:
Create the PPP virtual endpoint (using the **interface ppp**) command and enter the PPP command set.

(config)# **interface ppp 7**
(config-ppp 7)#

Step 2:
Configure the interface parameters to apply access policies to the interface, create bridging interfaces, configure dial-backup, and assign an IP address. For example, the following assigns an IP address of **193.44.69.1/30** to the interface.

(config-ppp 7)# **ip address 193.44.69.1 255.255.255.252**

Functional Notes (Continued)

Step 3: (VALID ONLY FOR T1 INTERFACES)
Specify the group of DS0s used for signaling on the T1 interface by creating a **tdm-group**. Group any number of aggregate DS0s together to create a data pipe for layer 2 signaling. Also use the **tdm-group** command to specify the per-DS0 signaling rate on the interface. For example, the following creates a tdm-group labeled **9** containing 20 DS0s (each DS0 having a data rate of 56 kbps).

(config)# **interface t1 1/1**
(config-t1 1/1)# **tdm-group 9 timeslots 1-20 speed 56**
(config-t1 1/1)# **exit**

Step 4:
Make the association between the layer 2 endpoint and the physical interface using the **cross-connect** command. Supported layer 2 protocols include frame relay and point-to-point protocol (PPP). For example, the following creates a cross-connect (labeled **5**) to make an association between the PPP virtual interface (**ppp 7**) and the tdm-group configured on interface t1 1/1 (**tdm-group 9**).

(config)# **cross-connect 5 t1 1/1 9 ppp 7**

Usage Examples

The following example creates a PPP virtual interface (labeled **1**) and enters the PPP Interface Configuration Mode:

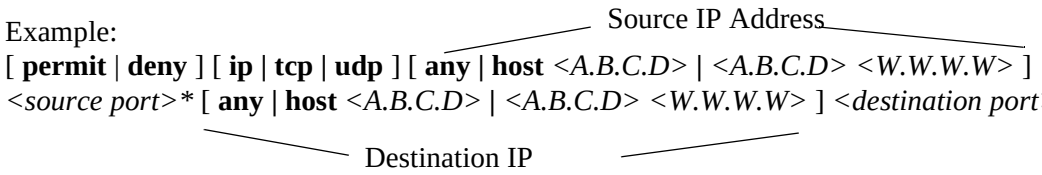
(config)# **interface ppp 1**
(config-ppp 1)#

ip access-list extended <*listname*>

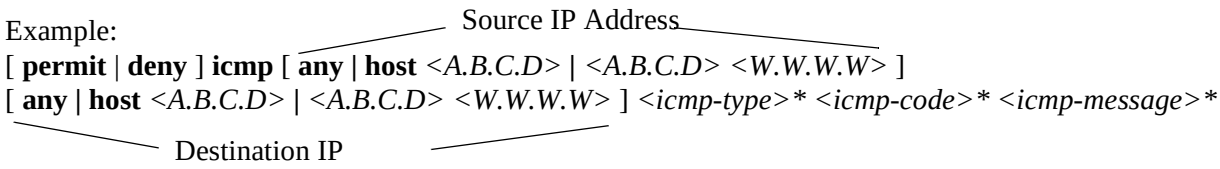
Use the **ip access-list extended** command to create an empty access list and enter the extended access-list command set. Use the **no** form of this command to delete an access list and all the entries contained in it. The following lists the complete syntax for the **ip access-list extended** commands:

<action> <protocol> <source IP> <source port> <destination ip> <destination port>

Example:



Example:



* = optional

Syntax Description

<*listname*>	Alphanumeric descriptor for identifying the configured access list (all access list descriptors are case-sensitive)
<*protocol*>	Specifies the data protocol such as ip, icmp, tcp, udp, or a specific protocol (0-255)
<*source ip*>	Specifies the source IP address used for packet matching IP addresses can be expressed in one of three ways: 1. Using the keyword **any** to match any IP address. For example, entering **deny any** will effectively shut down the interface that uses the access list because all traffic will match the **any** keyword. 2. Using the **host** <A.B.C.D> to specify a single host address. For example, entering **permit 196.173.22.253** will allow all traffic from the host with an IP address of 196.173.22.253. 3. Using the <A.B.C.D> <wildcard> format to match all IP addresses in a "range". Wildcard masks work in reverse logic from subnet mask. Specifying a one in the wildcard mask equates to a "don't care". For example, entering **deny 192.168.0.0 0.0.0.255** will deny all traffic from the 192.168.0.0/24 network.

Syntax Description (Continued)

<source port>
Optional

The source port is used only when *<protocol>* is **tcp** or **udp**

The following keywords and port numbers are supported for the *<source port>* field:

any
Match any destination port

eq *<port number>*
Match only packets on a given port number

gt *<port number>*
Match only packets with a port number higher than the one listed

host *<port number>*
Match a single destination host

lt *<port number>*
Match only packets with a port number lower than the one listed

neq *<port number>*
Match only packets that do not contain the specified port number

range *<port number>*
Match only packets that contain a port number specified in the listed range

The *<port number>* may be specified using the following syntax: <0-65535>. Specifies the port number used by TCP or UDP to pass information to upper layers. All ports below 1024 are considered well-known ports and are controlled by the Internet Assigned Numbers Authority (IANA). All ports above 1024 are dynamically assigned ports that include registered ports for vendor-specific applications

<port list>
The ADTRAN OS provides a condensed list of port numbers that may be entered using a text name

The following is the list of UDP port numbers that may be identified using the text name (in **bold**):

biff (Port 512)
bootpc (Port 68)
bootps(Port 67)
discard (Port 9)
dnsix (Port 195)
domain (Port 53)
echo (Port 7)
isakmp (Port 500)
mobile-ip (Port 434)
nameserver (Port 42)
netbios-dgm (Port 138)
netbios-ns (Port 137)
netbios-ss (Port 139)
ntp (Port 123)
pim-auto-rp (496)
rip (Port 520)
snmp (Port 161)
snmptrap (Port 162)
sunrpc (Port 111)
syslog (Port 514)
tacacs (Port 49)
talk (Port 517)
tftp (Port 69)
time (Port 37)
who (Port 513)
xdmcp (Port 177)

Syntax Description (Continued)

The following is the list of TCP port numbers that may be identified using the text name (in **bold**):

bgp (Port 179)	**lpd** (Port 515)
chargen (Port 19)	**nntp** (Port 119)
cmd (Port 514)	**pim-auto-rp** (Port 496)
daytime (Port 13)	**pop2** (Port 109)
discard (Port 9)	**pop3** (Port 110)
domain (Port 53)	**smtp** (Port 25)
echo (Port 7)	**sunrpc** (Port 111)
exec (Port 512)	**syslog** (Port 514)
finger (Port 79)	**tacacs** (Port 49)
ftp (Port 21)	**talk** (Port 517)
gopher (Port 70)	**tftp** (Port 69)
hostname (Port 101)	**telnet** (Port 23)
ident (Port 113)	**time** (Port 37)
irc (Port 194)	**uucp** (Port 540)
klogin (Port 543)	**whois** (Port 43)
kshell (Port 544)	**www** (Port 80)
login (Port 513)	

<destination ip>

Specifies the destination IP address used for packet matching

IP addresses can be expressed in one of three ways:
1. Using the keyword **any** to match any IP address. For example, entering **deny any** will effectively shut down the interface that uses the access list because all traffic will match the **any** keyword.
2. Using the **host** <A.B.C.D> to specify a single host address. For example, entering **permit 196.173.22.253** will allow all traffic from the host with an IP address of 196.173.22.253.
3. Using the <A.B.C.D> <wildcard> format to match all IP addresses in a “range”. Wildcard masks work in reverse logic from subnet mask. Specifying a one in the wildcard mask equates to a “don’t care”. For example, entering **deny 192.168.0.0 0.0.0.255** will deny all traffic from the 192.168.0.0/24 network.

<destination port>
Optional

Only valid when *<protocol>* is **tcp** or **udp** (See previously listed *<source port>* for more details)

<icmp-type>
Optional

Filter packets using ICMP defined (and numbered) messages carried in IP datagrams (used to send error and control information). Valid range is 0 to 255.

Syntax Description (Continued)

<icmp-code> ***Optional*	ICMP packets that are filtered using the ICMP message type (using the *<icmp-type>* keyword) may also be filtered using the ICMP message code (valid range: 0 to 255). An *<icmp-type>* must be specified when entering an *<icmp-code>.*
<icmp-message> ***Optional*	Filter packets using ICMP descriptive message rather than the corresponding type and code associations.

Default Values

By default, all ADTRAN OS security features are disabled and there are no configured access lists.

Command Modes

(config)#	Global Configuration Mode

Command History

Release 2.1	Command was introduced

Functional Notes

Access control lists (ACLs) are used as packet selectors by other ADTRAN OS systems; by themselves they do nothing. ACLs are composed of an ordered list of entries with an implicit **deny all** at the end of each list. An ACL entry contains two parts: an action (permit or deny) and a packet pattern. A permit ACL is used to allow packets (meeting the specified pattern) to enter the router system. A deny ACL is used to block entry to the network for specified criteria. The ADTRAN OS provides two types of ACLs: standard and extended. Standard ACLs allow source IP address packet patterns only. Extended ACLs may specify patterns using most fields in the IP header and the TCP or UDP header.

ACLs are performed in order from the top of the list down. Generally, the most specific entries should be at the top and the most general at the bottom.

The following commands are contained in the access-list extended command set:

remark
Use the **remark** command to associate a descriptive tag (up to 80 alphanumeric characters encased in quotation marks) to the access-list. Enter a functional description for the list such as “This list blocks all outbound web traffic”.

log
Using the **log** keyword logs a message (if **debug access-list** is enabled for this access list) when the access list finds a packet match.

Usage Examples

The following example creates an access list **AllowIKE** to allow all IKE (UDP Port 500) packets from the 190.72.22.55.0/24 network:

(config)# **ip access-list extended AllowIKE**
(config-ext-nacl)# **permit udp 190.72.22.55.0 0.0.0.255 eq 500 any eq 500**

For more details, refer to the ***NetVanta 3000 Series System Manual*** CD or the ADTRAN website (www.adtran.com) for technical support notes regarding access-list configuration.

Technology Review

Creating access policies and lists to regulate traffic through the routed network is a four-step process:

Step 1:

Enable the security features of the ADTRAN OS using the **ip firewall** command.

Step 2:

Create an access control list (using the **ip access-list** command) to permit or deny specified traffic. Standard access lists provide pattern matching for source IP addresses only. (Use extended access lists for more flexible pattern matching.) IP addresses can be expressed in one of three ways:

1. Using the keyword **any** to match any IP address. For example, entering **deny any** will effectively shut down the interface that uses the access list because all traffic will match the **any** keyword.
2. Using the **host** <A.B.C.D> to specify a single host address. For example, entering **permit 196.173.22.253** will allow all traffic from the host with an IP address of 196.173.22.253.
3. Using the <A.B.C.D> <wildcard> format to match all IP addresses in a “range”. Wildcard masks work in reverse logic from subnet mask. Specifying a one in the wildcard mask equates to a “don’t care”. For example, entering **deny 192.168.0.0 0.0.0.255** will deny all traffic from the 192.168.0.0/24 network.

Step 3:

Create an access control policy (using the **ip policy-class** command) that uses a configured access list. ADTRAN OS access policies are used to allow, discard, or manipulate (using NAT) data for each physical interface. Each ACP consists of a selector (access list) and an action (allow, discard, NAT). When packets are received on an interface, the configured ACPs are applied to determine whether the data will be processed or discarded. Possible actions performed by the access policy are as follows:

allow list <*access list names*>
All packets passed by the access list(s) entered will be allowed to enter the router system.

discard list <*access list names*>
All packets passed by the access list(s) entered will be dropped from the router system.

allow list <*access list names*> **dest-policy** <*access policy name*>
All packets passed by the access list(s) entered and destined for the interface using the access policy listed will be permitted to enter the router system. This allows for configurations to permit packets to a single interface and not the entire system.

Technology Review (Continued)

discard list <*access list names*> **dest-policy** <*access policy name*>
All packets passed by the access list(s) entered and destined for the interface using the access policy listed will be blocked from the router system. This allows for configurations to deny packets on a specified interface.

nat source list <*access list names*> **address** <*IP address*> **overload**
All packets passed by the access list(s) entered will be modified to replace the source IP address with the entered IP address. The **overload** keyword allows multiple source IP addresses to be replaced with the single IP address entered. This hides private IP addresses from outside the local network.

nat source list <*access list names*> **interface** <*interface*> **overload**
All packets passed by the access list(s) entered will be modified to replace the source IP address with the primary IP address of the listed interface. The **overload** keyword allows multiple source IP addresses to be replaced with the single IP address of the specified interface. This hides private IP addresses from outside the local network.

nat destination list <*access list names*> **address** <*IP address*>
All packets passed by the access list(s) entered will be modified to replace the destination IP address with the entered IP address. The **overload** keyword is not an option when performing NAT on the destination IP address; each private address must have a unique public address. This hides private IP addresses from outside the local network.

WARNING

Before applying an access control policy to an interface, verify your Telnet connection will not be affected by the policy. If a policy is applied to the interface you are connecting through and it does not allow Telnet traffic, your connection will be lost.

Step 4:

Apply the created access control policy to an interface. To assign an access policy to an interface, enter the interface configuration mode for the desired interface and enter **access policy** <*policy name*>. The following example assigns access policy **MatchAll** to the ethernet 0/1 interface:

(config)# **interface ethernet 0/1**
(config-eth 0/1)# **access-policy MatchAll**

ip access-list standard *<listname>*

Use the **ip access-list standard** command to create an empty access list and enter the standard access-list command set. Use the **no** form of this command to delete an access list and all the entries contained in it. The following lists the complete syntax for the **ip access-list standard** commands:

ip access-list standard *<listname>*
[**permit** or **deny**] **any**
[**permit** or **deny**] **host** *<ip address>*
[**permit** or **deny**] *<ip address> <wildcard>*

Syntax Description

<listname>	Alphanumeric descriptor for identifying the configured access list (all access list descriptors are case-sensitive).
<action>	Permit or deny entry to the routing system for specified packets.
<source ip>	Specifies the source IP address used for packet matching. IP addresses can be expressed in one of three ways: 1. Using the keyword **any** to match any IP address. For example, entering **deny any** will effectively shut down the interface that uses the access list because all traffic will match the **any** keyword. 2. Using the **host** <A.B.C.D> to specify a single host address. For example, entering **permit 196.173.22.253** will allow all traffic from the host with an IP address of 196.173.22.253. 3. Using the <A.B.C.D> <wildcard> format to match all IP addresses in a "range". Wildcard masks work in reverse logic from subnet mask. Specifying a one in the wildcard mask equates to a "don't care". For example, entering **deny 192.168.0.0 0.0.0.255** will deny all traffic from the 192.168.0.0/24 network.

Default Values

By default, all ADTRAN OS security features are disabled and there are no configured access lists.

Command Modes

(config)#	Global Configuration Mode

Command History

Release 2.1	Command was introduced

Functional Notes

Access control lists are used as packet selectors by access policies (ACPs); by themselves they do nothing. ACLs are composed of an ordered list of entries with an implicit **deny all** at the end of each list. An ACL entry contains two parts: an action (permit or deny) and a packet pattern. A permit ACL is used to allow packets (meeting the specified pattern) to enter the router system. A deny ACL is used to block entry to the network for specified criteria. The ADTRAN OS provides two types of ACLs: standard and extended. Standard ACLs allow source IP address packet patterns only. Extended ACLs may specify patterns using most fields in the IP header and the TCP or UDP header.

ACLs are performed in order from the top of the list down. Generally the most specific entries should be at the top and the most general at the bottom.

The following commands are contained in the access-list standard command set:

remark
Use the **remark** command to associate a descriptive tag (up to 80 alphanumeric characters encased in quotation marks) to the access-list. Enter a functional description for the list such as “This list blocks all outbound web traffic”.

log
use the **log** keyword to log a message (if **debug access-list** is enabled for this access list) when the access list finds a packet match.

permit or **deny any**
Use the **any** keyword to match any IP address received by the access list. For example, the following allows all packets through the configured access list:

(config)# **ip access-list standard MatchAll**
(config-std-nacl)# **permit any**

permit or **deny host** <*ip address*>
Use the **host** <A.B.C.D> keyword to specify a single host address. For example, the following allows all traffic from the host with an IP address of 196.173.22.253.

(config)# **ip access-list standard MatchHost**
(config-std-nacl)# **permit 196.173.22.253**

permit or **deny** <*ip address*> <*wildcard*>
Use the <A.B.C.D> <wildcard> format to match all IP addresses in a “range”. Wildcard masks work in reverse logic from subnet mask. Specifying a one in the wildcard mask equates to a “don’t care”. For example, the following denies all traffic from the 192.168.0.0/24 network:

(config)# **ip access-list standard MatchNetwork**
(config-std-nacl)# **deny 192.168.0.0 0.0.0.255**

Usage Examples

The following example creates an access list **UnTrusted** to deny all packets from the 190.72.22.248/30 network:

(config)# **ip access-list standard UnTrusted**
(config-std-nacl)# deny 190.72.22.248 0.0.0.3

For more details, refer to the ***NetVanta 3000 Series System Manual*** CD or the ADTRAN website (www.adtran.com) for technical support notes regarding access-list configuration.

Technology Review

Creating access policies and lists to regulate traffic through the routed network is a four-step process:

Step 1:

Enable the security features of the ADTRAN OS using the **ip firewall** command.

Step 2:

Create an access list to permit or deny specified traffic. Standard access lists provide pattern matching for source IP addresses only. (Use extended access lists for more flexible pattern matching.) IP addresses can be expressed in one of three ways:

1. Using the keyword **any** to match any IP address. For example, entering **deny any** will effectively shut down the interface that uses the access list because all traffic will match the **any** keyword.
2. Using the **host** <A.B.C.D> to specify a single host address. For example, entering **permit 196.173.22.253** will allow all traffic from the host with an IP address of 196.173.22.253.
3. Using the <A.B.C.D> <wildcard> format to match all IP addresses in a “range”. Wildcard masks work in reverse logic from subnet mask. Specifying a one in the wildcard mask equates to a “don’t care”. For example, entering **deny 192.168.0.0 0.0.0.255** will deny all traffic from the 192.168.0.0/24 network.

Step 3:

Create an access policy that uses a configured access list. ADTRAN OS access policies are used to allow, discard, or manipulate (using NAT) data for each physical interface. Each ACP consists of a selector (access list) and an action (allow, discard, NAT). When packets are received on an interface, the configured ACPs are applied to determine whether the data will be processed or discarded. Possible actions performed by the access policy are as follows:

allow list <*access list names*>
All packets passed by the access list(s) entered will be allowed to enter the router system.

discard list <*access list names*>
All packets passed by the access list(s) entered will be dropped from the router system.

allow list <*access list names*> **dest-policy** <*access policy name*>
All packets passed by the access list(s) entered and destined for the interface using the access policy listed will be permitted to enter the router system. This allows for configurations to permit packets to a single interface and not the entire system.

Technology Review (Continued)

discard list <*access list names*> **dest-policy** <*access policy name*>
All packets passed by the access list(s) entered and destined for the interface using the access policy listed will be blocked from the router system. This allows for configurations to deny packets on a specified interface.

nat source list <*access list names*> **address** <*IP address*> **overload**
All packets passed by the access list(s) entered will be modified to replace the source IP address with the entered IP address. The **overload** keyword allows multiple source IP addresses to be replaced with the single IP address entered. This hides private IP addresses from outside the local network.

nat source list <*access list names*> **interface** <*interface*> **overload**
All packets passed by the access list(s) entered will be modified to replace the source IP address with the primary IP address of the listed interface. The **overload** keyword allows multiple source IP addresses to be replaced with the single IP address of the specified interface. This hides private IP addresses from outside the local network.

nat destination list <*access list names*> **address** <*IP address*>
All packets passed by the access list(s) entered will be modified to replace the destination IP address with the entered IP address. The **overload** keyword is not an option when performing NAT on the destination IP address; each private address must have a unique public address. This hides private IP addresses from outside the local network.

WARNING

Before applying an access control policy to an interface, verify your Telnet connection will not be affected by the policy. If a policy is applied to the interface you are connecting through and it does not allow Telnet traffic, your connection will be lost.

Step 4:

Apply the created access policy to an interface. To assign an access policy to an interface, enter the interface configuration mode for the desired interface and enter **access policy** <*policy name*>. The following example assigns access policy **MatchAll** to the ethernet 0/1 interface:

(config)# **interface ethernet 0/1**
(config-eth 0/1)# **access-policy MatchAll**

ip classless

Use the **ip classless** command to forward classless packets to the best supernet route available. A classless packet is a packet addressed for delivery to a subnet of a network with no default network route.

Syntax Description

No subcommands

Default Values

By default, this command is enabled.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

AOS products only function in classless mode. You cannot disable this feature.

Usage Examples

The following example enables the system to forward classless packets:

(config)# **ip classless**

ip crypto

Use the **ip crypto** command to enable ADTRAN OS VPN functionality and allow crypto maps to be added to interfaces. Use the **no** form of this command to disable the VPN functionality.

*Disabling the ADTRAN OS security features (using the **no ip crypto** command) does not affect VPN configuration settings (with the exception of the removal of all crypto maps from the interfaces). All other configuration parameters will remain intact, and VPN functionality will be disabled.*

*For VPN configuration example scripts, refer to the technical support note **Configuring VPN** located on the **NetVanta 3000 Series System Manual** CD provided with your unit.*

Syntax Description

No subcommands

Default Values

By default, all ADTRAN OS VPN functionality is disabled.

Command Modes

(config)#	Global Configuration Mode

Command History

Release 4.1	Command was introduced

Functional Notes

VPN-related settings will not go into effect until you enable VPN functionality using the **ip crypto** command. The AOS allows you to perform all VPN-related configuration prior to enabling **ip crypto**, with the exception of assigning a **crypto map** to an interface. The **no ip crypto** command removes all crypto maps from the interfaces.

Enabling **ip crypto** enables the IKE server on UDP port 500. The **no** form of this command disables the IKE server on UDP port 500.

Usage Examples

The following example enables VPN functionality:

(config)# **ip crypto**

ip default-gateway *<ip address>*

Use the **ip default-gateway** command to specify a default gateway if (and only if) IP routing is NOT enabled on the unit. Use the **ip route** command to add a default route to the route table when using IP routing functionality. See *ip route <ip address> <subnet mask> <interface or ip address>* on page 213 for more information.

Syntax Description

<ip address>	Specifies the default gateway IP address in the form of dotted decimal notation (example: 192.22.71.50).

Default Values

By default, there is no configured default-gateway.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

Only use the **ip default-gateway** when IP routing is disabled on the router. For all other cases, use the **ip route 0.0.0.0 0.0.0.0** *<ip address>* command.

Usage Examples

The following example disables IP routing and configures a default gateway for 192.22.71.50:

(config)# **no ip routing**
(config)# **ip default gateway 192.22.71.50**

ip dhcp-server excluded-address *<start ip> <end ip>*

Use the **ip dhcp-server excluded-address** command to specify IP addresses that cannot be assigned to DHCP clients. Use the **no** form of this command to remove a configured IP address restriction.

Syntax Description

<start ip>	Specifies the lowest IP address (using dotted decimal notation) in the range OR a single IP address to be excluded.
<end ip> **Optional*	Specifies the highest IP address (using dotted decimal notation) in the range. This field is not required when specifying a single IP address.

Default Values

By default, there are no excluded IP addresses.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 2.1	Command was introduced

Functional Notes

The ADTRAN OS DHCP server (by default) allows all IP addresses for the DHCP pool to be assigned to requesting clients. This command is used to ensure that the specified address is never assigned by the DHCP server. When static addressed hosts are present in the network, it is helpful to exclude the IP addresses of the host from the DHCP IP address pool. This will avoid IP address overlap.

Usage Examples

The following example excludes an IP address of 172.22.5.100 and the range 172.22.5.200 through 172.22.5.250:

(config)# **ip dhcp-server excluded-address 172.22.5.100**
(config)# **ip dhcp-server excluded-address 172.22.5.200 172.22.5.250**

ip dhcp-server ping packets *<#packets>*

Use the **ip dhcp-server ping packets** command to specify the number of ping packets the DHCP server will transmit before assigning an IP address to a requesting DHCP client. Transmitting ping packets verifies that no other hosts on the network are currently configured with the specified IP address. Use the **no** form of this command to prevent the DHCP server from using ping packets as part of the IP address assignment process.

Syntax Description

<#packets>	Specifies the number of DHCP ping packets sent on the network before assigning the IP address to a requesting DHCP client

Default Values

<#packets>	2 packets

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 2.1	Command was introduced

Functional Notes

Before assigning an IP address to a requesting client, the ADTRAN OS DHCP server transmits a ping packet on the network to verify there are no other network hosts already configured with the specified address. If the DHCP server receives no reply, the IP address is assigned to the requesting client and added to the DHCP database as an assigned address. Configuring the **ip dhcp-server ping packets** command with a value of **0** prevents the DHCP server from using ping packets as part of the IP address assignment process.

Usage Examples

The following example configures the DHCP server to transmit 4 ping packets before assigning an address:

(config)# **ip dhcp-server ping packets 4**

ip dhcp-server ping timeout <*milliseconds*>

Use the **ip dhcp-server ping timeout** command to specify the interval (in milliseconds) the DHCP server will wait for a response to a transmitted DHCP ping packet. The DHCP server transmits ping packets before assigning an IP address to a requesting DHCP client. Transmitting ping packets verifies that no other hosts on the network are currently configured with the specified IP address. Use the **no** form of this command to return to the default timeout interval.

Syntax Description

<*milliseconds*>	Specifies the number of milliseconds the DHCP server will wait for a response to a transmitted DHCP ping packet.

Default Values

<*milliseconds*>	500 milliseconds

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 2.1	Command was introduced

Functional Notes

Before assigning an IP address to a requesting client, the ADTRAN OS DHCP server transmits a ping packet on the network to verify there are no other network hosts already configured with the specified address. If the DHCP server receives no reply, the IP address is assigned to the requesting client and added to the DHCP database as an assigned address.

Usage Examples

The following example configures the DHCP server to wait 900 milliseconds for a response to a transmitted DHCP ping packet before considering the ping a failure:

(config)# **ip dhcp-server ping timeout 900**

ip dhcp-server pool *<name>*

Use the **ip dhcp-server pool** command to create a DHCP address pool and enter the DHCP pool command set. Use the **no** form of this command to remove a configured DHCP address pool. See the section *DHCP Pool Command Set* on page 240 for more information.

Syntax Description

<name>	Alphanumeric string (up to 32 characters in length) used as an identifier for the configured DHCP server address pool (example SALES)

Default Values

By default, there are no configured DHCP address pools.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 2.1	Command was introduced

Functional Notes

Use the **ip dhcp-server pool** to create multiple DHCP server address pools for various segments of the network. Multiple address pools can be created to service different segments of the network with tailored configurations.

Usage Examples

The following example creates a DHCP server address pool (labeled SALES) and enters the DHCP server pool command set:

```
(config)# ip dhcp-server pool SALES
(config-dhcp)#
```


ip domain-lookup

Use the **ip domain-lookup** command to enable the IP DNS (domain naming system), allowing DNS-based host translation (name-to-address). Use the **no** form of this command to disable DNS.

Syntax Description

No subcommands

Default Values

By default, this command is enabled.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 3.1	Command was introduced

Functional Notes

Use the **ip domain-lookup** command to enable the DNS client in the router. This will allow the user to input web addresses instead of IP addresses for applications such as ping, Telnet, and traceroute.

Usage Examples

The following example enables DNS:

(config)# **ip domain-lookup**

ip domain-name *<name>*

Use the **ip domain-name** command to define a default IP domain name to be used by the ADTRAN OS to resolve host names. Use the **no** form of this command to disable this function.

Syntax Description

<name>	Default IP domain name used to resolve unqualified host names. Do not include the initial period that separates the unresolved name from the default domain name.

Default Values

By default, this command is disabled.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 3.1	Command was introduced

Functional Notes

Use the **ip domain-name** command to set a default name which will be used to complete any IP host name that is invalid (i.e., any name that is not recognized by the name-server). When this command is enabled, any IP host name that is not initially recognized will have the **ip domain-name** appended to it and the request will be resent.

Usage Examples

The following example defines **adtran** as the default domain name:

(config)# **ip domain-name adtran**

ip domain-proxy

Use the **ip domain-proxy** command to enable DNS proxy for the router. This enables the router to act as a proxy for other units on the network.

Syntax Description

No subcommands

Default Values

By default, this command is disabled.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 3.1	Command was introduced

Functional Notes

When this command is enabled, incoming DNS requests will be handled by the router. It will first search its host table for the query, and if it is not found there the request will be forwarded to the servers configured with the **ip name-server** command.

Usage Examples

The following example enables DNS proxy:

(config)# **ip domain-proxy**

ip firewall

Use the **ip firewall** command to enable ADTRAN OS security features including access control policies and lists, Network Address Translation (NAT), and the stateful inspection firewall. Use the **no** form of this command to disable the security functionality.

*Disabling the ADTRAN OS security features (using the **no ip firewall** command) does not affect security configuration. All configuration parameters will remain intact, but no security data processing will be attempted.*

*For information regarding the use of OSPF with **ip firewall** enabled, see the **Functional Note** for router ospf on page 227.*

*For information regarding the use of IKE negotiation for VPN with **ip firewall** enabled, see the **Functional Note** for the command crypto map <mapname> on page 301.*

Syntax Description

No subcommands

Default Values

By default, all ADTRAN OS security features are disabled.

Command Modes

(config)#	Global Configuration Mode

Command History

Release 2.1	Command was introduced

Functional Notes

This command enables firewall processing for all interfaces with a configured policy class. Firewall processing consists of the following functions:

1. Attack Protection: Detects and discards traffic that matches profiles of known networking exploits or attacks.
2. Session Initiation Control: Allows only sessions that match traffic patterns permitted by access-control policies to be initiated through the router.
3. Ongoing Session Monitoring and Processing: Each session that has been allowed through the router is monitored for any irregularities that match patterns of known attacks or exploits. This traffic will be dropped. Also, if NAT is configured, the firewall modifies all traffic associated with the session according to the translation rules defined in NAT access-policies. Finally, if sessions are inactive for a user-specified amount of time, the session will be closed by the firewall.
4. Application Specific Processing: Certain applications need special handling to work correctly in the presence of a firewall. ADTRAN OS uses ALGs (application-level gateways) for these applications.

The ADTRAN OS includes several security features to provide controlled access to your network. The following features are available when security is enabled (using the **ip firewall** command):

1. Stateful Inspection Firewall
The ADTRAN OS (and your unit) act as an application-level gateway and employ a stateful inspection firewall that protects an organization's network from common cyber attacks including TCP syn-flooding, IP spoofing, ICMP redirect, land attacks, ping-of-death, and IP reassembly problems. In addition, further security is added with use of Network Address Translation (NAT) and Port Address Translation (PAT) capability.

2. Access Policies (ACPs)
ADTRAN OS access control policies are used to allow, discard, or manipulate (using NAT) data for each physical interface. Each ACP consists of a selector (access list) and an action (allow, discard, NAT). When packets are received on an interface, the configured ACPs are applied to determine whether the data will be processed or discarded.

3. Access Lists (ACLs)
Access control lists are used as packet selectors by ACPs; by themselves they do nothing. ACLs are composed of an ordered list of entries. Each entry contains two parts: an action (permit or deny) and a packet pattern. A permit ACL is used to permit packets (meeting the specified pattern) to enter the router system. A deny ACL is used to block entry to the network for specified criteria. The ADTRAN OS provides two types of ACLs: standard and extended. Standard ACLs allow source IP address packet patterns only. Extended ACLs may specify patterns using most fields in the IP header and the TCP or UDP header.

Usage Examples

The following example enables the ADTRAN OS security features:

(config)# **ip firewall**

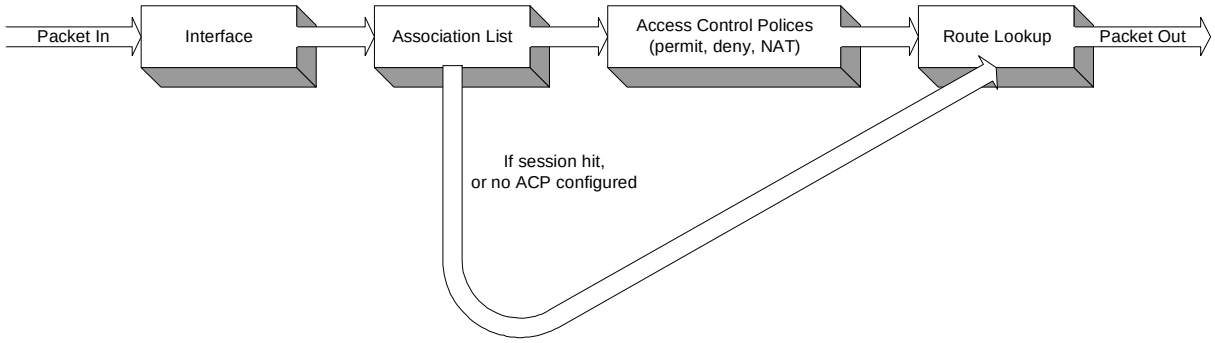
Technology Review

Concepts:

Access control using the ADTRAN OS firewall has two fundamental parts: Access Control Lists (ACLs) and Access Policy Classes (ACPs). ACLs are used as packet selectors by other ADTRAN OS systems; by themselves they do nothing. ACPs consist of a selector (ACL) and an action (allow, discard, NAT). ACPs integrate both allow and discard policies with NAT. ACPs have no effect until they are assigned to a network interface.

Both ACLs and ACPs are order dependent. When a packet is evaluated, the matching engine begins with the first entry in the list and progresses through the entries until it finds a match. The first entry that matches is executed.

Packet Flow:



Case 1: Packets from interfaces with a configured policy class to any other interface

ACPs are applied when packets are received on an interface. If an interface has not been assigned a policy class, by default it will allow all received traffic to pass through. If an interface has been assigned a policy class but the firewall has not been enabled with the **ip firewall** command, traffic will flow normally from this interface with no firewall processing.

Case 2: Packets that travel in and out a single interface with a configured policy class

These packets are processed through the ACPs as if they are destined for another interface (identical to Case 1).

Case 3: Packets from interfaces without a configured policy class to interfaces with one

These packets are routed normally and are not processed by the firewall. The **ip firewall** command has no effect on this traffic.

Case 4: Packets from interfaces without a configured policy class to other interfaces without a configured policy class

This traffic is routed normally. The **ip firewall** command has no effect on this traffic.

Technology Review (Continued)

Attack Protection:

When the **ip firewall** command is enabled and access-policies are created using the **ip policy-class** command and applied to interfaces with the **access-policy** command, firewall attack protection is enabled. The ADTRAN OS blocks traffic (matching patterns of known networking exploits) from traveling through the device. For some of these attacks, the user may manually disable checking/blocking while other attack checks are always on anytime the firewall is enabled.

The table (on the following pages) outlines the types of traffic discarded by the Firewall Attack Protection Engine. Many attacks use similar invalid traffic patterns; therefore attacks other than the examples listed below may also be blocked by the firewall. To determine if a specific attack is blocked by the ADTRAN OS firewall, please contact ADTRAN technical support.

Invalid Traffic Pattern	Manually Enabled?	ADTRAN OS Firewall Response	Common Attacks
Larger than allowed packets	No	Any packets that are longer than those defined by standards will be dropped.	Ping of Death
Fragmented IP packets that produce errors when attempting to reassemble	No	The firewall intercepts all fragments for an IP packet and attempts to reassemble them before forwarding to destination. If any problems or errors are found during reassembly, the fragments are dropped.	SynDrop, TearDrop, OpenTear, Nestea, Targa, Newtear, Bonk, Boink
Smurf Attack	No	The firewall will drop any ping responses that are not part of an active session.	Smurf Attack
IP Spoofing	No	The firewall will drop any packets with a source IP address that appears to be spoofed. The IP route table is used to determine if a path to the source address is known (out of the interface from which the packet was received). For example, if a packet with a source IP address of 10.10.10.1 is received on interface fr 1.16 and no route to 10.10.10.1 (through interface fr 1.16) exists in the route table, the packet is dropped.	IP Spoofing
ICMP Control Message Floods and Attacks	No	The following types of ICMP packets are allowed through the firewall: echo, echo-reply, TTL expired, dest. Unreachable, and quench. These ICMP messages are only allowed if they appear to be in response to a valid session. All others are discarded.	Twinge

Invalid Traffic Pattern	Manually Enabled?	ADTRAN OS Firewall Response	Common Attacks
Attacks that send TCP URG packets	Yes	Any TCP packets that have the URG flag set are discarded by the firewall.	Winnuke, TCP XMAS Scan
Falsified IP Header Attacks	No	The firewall verifies that the packet's actual length matches the length indicated in the IP header. If it does not, the packet is dropped.	Jolt/Jolt2
Echo	No	All UDP echo packets are discarded by the firewall.	Char Gen
Land Attack	No	Any packets with the same source and destination IP addresses are discarded.	Land Attack
Broadcast Source IP	No	Packets with a broadcast source IP address are discarded.	
Invalid TCP Initiation Requests	No	TCP SYN packets that have ack, urg rst, or fin flags set are discarded.	
Invalid TCP Segment Number	No	The sequence numbers for every active TCP session are maintained in the firewall session database. If the firewall received a segment with an unexpected (or invalid) sequence number, the packet is dropped.	
IP Source Route Option	No	All IP packets containing the IP source route option are dropped.	

Application Specific Processing:

The following applications and protocols require special processing to operate concurrently with NAT/firewall functionality. The ADTRAN OS firewall includes ALGs for handling these applications and protocols:

AOL Instant Messenger
VPN ALGS: ESP and IKE
FTP
H.323: H.245 Q.931 ASN1 PER decoding and Encoding
ICQ
IRC
Microsoft Games
Net2Phone
PPTP
Quake
Real-Time Streaming Protocol
SMTP
HTTP
CUseeme
SIP
L2TP
PcAnywhere
SQL
Microsoft Gaming Zone

To determine if a specific application requires special processing, contact ADTRAN at *www.adtran.com*.

ip firewall attack-log threshold <*value*>

Use the **ip firewall attack-log threshold** command to specify the number of attack mounting attempts the ADTRAN OS will identify before generating a log message. Use the **no** form of this command to return to the default threshold.

> NOTE *The ADTRAN OS security features must be enabled (using the **ip firewall** command) for the stateful inspection firewall to be activated.*

Syntax Description

<*value*>	Specifies the number of attack mounting attempts the ADTRAN OS will identify before generating a log message (valid range: 0 to 4294967295).

Default Values

<*value*>	100

Command Modes

(config)#	Global Configuration Mode

Command History

Release 2.1	Command was introduced

Usage Examples

The following example specifies a threshold of 25 attacks before generating a log message:

(config)# **ip firewall attack-log threshold 25**

ip firewall check syn-flood

Use the **ip firewall check syn-flood** command to enable the ADTRAN OS stateful inspection firewall to filter out phony TCP service requests and allow only legitimate requests to pass through. Use the **no** form of this command to disable this feature.

*The ADTRAN OS security features must be enabled (using the **ip firewall** command) for the stateful inspection firewall to be activated.*

Syntax Description

No subcommands

Default Values

*All ADTRAN OS security features are disabled by default until the **ip firewall** command is issued at the the Global Configuration prompt. In addition, the SYN-flood check is disabled until the **ip firewall check syn-flood** command is issued.*

Command Modes

(config)#	Global Configuration Mode

Command History

Release 2.1	Command was introduced

Functional Notes

SYN Flooding is a well-known denial of service attack on TCP-based services. TCP requires a three-way handshake before actual communications begin between two hosts. A server must allocate resources to process new connection requests that are received. A potential intruder is capable of transmitting large amounts of service requests (in a very short period of time), causing servers to allocate all resources to process the phony incoming requests. Using the **ip firewall check syn-flood** command configures the ADTRAN OS stateful inspection firewall to filter out phony service requests and allow only legitimate requests to pass through.

Usage Examples

The following example enables the ADTRAN OS syn-flood check:

(config)# **ip firewall check syn-flood**

ip firewall check winnuke

Use the **ip firewall check winnuke** command to enable the ADTRAN OS stateful inspection firewall to discard all Out of Band (OOB) data (to protect against WinNuke attacks). Use the **no** form of this command to disable this feature.

*The ADTRAN OS security features must be enabled (using the **ip firewall** command) for the stateful inspection firewall to be activated.*

Syntax Description

No subcommands

Default Values

*All ADTRAN OS security features are disabled by default until the **ip firewall** command is issued at the the Global Configuration prompt. Issuing the **ip firewall** command enables the WinNuke check.*

Command Modes

(config)#	Global Configuration Mode

Command History

Release 2.1	Command was introduced

Functional Notes

WinNuke attack is a well-known denial of service attack on hosts running Windows® operating systems. An intruder sends Out of Band (OOB) data over an established connection to a Windows user. Windows cannot properly handle the OOB data and the host reacts unpredictably. Normal shut-down of the hosts will generally return all functionality. Using the **ip firewall check winnuke** command configures the ADTRAN OS stateful inspection firewall to filter all OOB data to prevent network problems.

Usage Examples

The following example enables the firewall to filter all OOB data:

(config)# **ip firewall check winnuke**

ip firewall policy-log threshold <*value*>

Use the **ip firewall policy-log threshold** command to specify the number of connections required by an access control policy before the ADTRAN OS will generate a log message. Use the **no** form of this command to return to the default threshold.

*The ADTRAN OS security features must be enabled (using the **ip firewall** command) for the stateful inspection firewall to be activated.*

Syntax Description

<*value*>	Specifies the number of access policy connections the ADTRAN OS will identify before generating a log message (valid range: 0 to 4294967295).

Default Values

<*value*>	100

Command Modes

(config)#	Global Configuration Mode

Command History

Release 2.1	Command was introduced

Usage Examples

The following example specifies a threshold of 15 connections before generating a log message:

(config)# **ip firewall policy-log threshold 15**

ip forward-protocol udp <*port number*>

Use the **ip forward-protocol udp** command to specify the protocols and ports the ADTRAN OS allows when forwarding broadcast packets. Use the **no** form of this command to disable a specified protocol or port from being forwarded.

> NOTE *The **ip helper** command must be used in conjunction with the **ip forward-protocol** command to configure the ADTRAN OS to forward UDP broadcast packets. See ip helper-address <address> on page 312 for more information.*

Syntax Description

<*port number*>	Specifies the UDP traffic type (using source port)

The following is the list of UDP port numbers that may be identified using the text name:

biff (Port 512)	ntp (Port 123)
bootpc (Port 68)	pim-auto-rp (496)
bootps(Port 67)	rip (Port 520)
discard (Port 9)	snmp (Port 161)
dnsix (Port 195)	snmptrap (Port 162)
domain (Port 53)	sunrpc (Port 111)
echo (Port 7)	syslog (Port 514)
isakmp (Port 500)	tacacs (Port 49)
mobileip (Port 434)	talk (Port 517)
nameserver (Port 42)	tftp (Port 69)
netbios-dgm (Port 138)	time (Port 37)
netbios-ns (Port 137)	who (Port 513)
netbios-ss (Port 139)	xdmcp (Port 177)

Alternately, the <*port number*> may be specified using the following syntax: <0-65535>. Specifies the port number used by UDP to pass information to upper layers. All ports below 1024 are considered well-known ports and are controlled by the Internet Assigned Numbers Authority (IANA). All ports above 1024 are dynamically assigned ports that include registered ports for vendor-specific applications.

Default Values

By default, the ADTRAN OS forwards broadcast packets for all protocols and ports.

Command Modes

(config)#	Global Configuration Mode

Command History

Release 2.1	Command was introduced

Functional Notes

Use this command to configure the ADTRAN OS to forward UDP packets across the WAN link to allow remote devices to connect to a UDP service on the other side of the WAN link.

Usage Examples

The following example forwards all Domain Name Server broadcast traffic to the DNS server with IP address 192.33.5.99:

(config)# **ip forward-protocol udp domain**
(config)# **interface eth 0/1**
(config-eth 0/1)# **ip helper-address 192.33.5.99**

ip ftp access-class <*policyname*> in

Use the **ip ftp access-class in** command to assign an access policy to all self-bound File Transfer Protocol (FTP) sessions.

Syntax Description

<*policyname*>	Specifies the configured access policy (ACP) to apply to inbound FTP traffic

Default Values

By default, all ftp access is allowed.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 2.1	Command was introduced

Usage Examples

The following example applies the configured ACP (labeled Inbound_FTP) to inbound FTP traffic:

(config)# **ip ftp access-class Inbound_FTP in**

ip ftp agent

Use the **ip ftp agent** command to enable the file transfer protocol (FTP) agent.

Syntax Description

No subcommands

Default Values

By default, the FTP agent is enabled.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 4.1	Command was introduced

Usage Examples

The following example enables the IP FTP agent:

(config)# **ip ftp agent**

ip host *<name> <address1>*

Use the **ip host** command to define an IP host name. This allows you to statically map host names and addresses in the host cache. Use the **no** form of this command to remove defined maps.

Syntax Description

<name>	Name of the host.
<address1>	IP address associated with this IP host.

Default Values

By default, the host table is empty.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 3.1	Command was introduced

Functional Notes

The name may be any combination of numbers and letters as long as it is not a valid IP address or does not exceed 256 characters.

Usage Examples

The following example defines two static mappings:

(config)# **ip host mac 10.2.0.2**
(config)# **ip host dal 172.38.7.12**

ip http [server | access-class <*listname*> in]

Use the **ip http** command to enable web access to the router.

Syntax Description

server	Enable the http server connection.
access-class	Enable http for all incoming connections associated with a specific access list.
<*listname*>	Access list name.
in	Apply to all incoming connections.

Default Values

By default, this command is disabled.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 3.1	Command was introduced

Usage Examples

The following example enables web access to the router:

(config)# **ip http server**

ip n-form [agent | access-class <*listname*> in]

Use the **ip n-form** command to enable and customize N-Form access to the router

Syntax Description

agent	Enable the N-Form agent.
access-class	Enable N-Form access for all incoming connections associated with a specific access list.
<*listname*>	Access list name.
in	Apply to all incoming connections.

Default Values

*By default, **n-form agent** is disabled.*

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 3.1	Command was introduced

Usage Examples

The following example enables N-Form access to the router:

(config)# **ip n-form agent**

ip name-server *<server-address1> [server-address2....server-address6]*

Use the **ip name-server** command to designate one or more name servers to use for name-to-address resolution. Use the **no** form of this command to remove any addresses previously specified.

Syntax Description

<server-address1-6>	Enter up to six name-server addresses.

Default Values

By default, no name servers are specified.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 3.1	Command was introduced

Usage Examples

The following example specifies host 172.34.1.111 as the primary name server and host 172.34.1.2 as the secondary server:

(config)# **ip name-server 172.341.1.111 172.34.1.2**

This command will be reflected in the configuration file as follows:

ip name-server 172.34.1.111 172.34.1.2

ip policy-class <*policyname*> max-sessions

Use the **ip policy-class** command to create an access control policy and enter the access control policy command set. Use the **no** form of this command to delete an access policy and all the entries contained in it.

*Configured access policies will only be active if the **ip firewall** command has been entered at the Global Configuration Mode prompt to enable the ADTRAN OS security features. All configuration parameters are valid, but no security data processing will be attempted unless the security features are enabled.*

Before applying an access control policy to an interface, verify your Telnet connection will not be affected by the policy. If a policy is applied to the interface you are connecting through and it does not allow Telnet traffic, your connection will be lost.

Syntax Description

<*policyname*>	Alphanumeric descriptor for identifying the configured access policy (all access policy descriptors are case-sensitive).
max-sessions **Optional*	Configure a maximum number of allowed policy sessions. This number must be within the appropriate range limits. The limits are either 1-4000 or 1-30000 (depending on the type of AOS device you are using).

Default Values

By default, all ADTRAN OS security features are disabled and there are no configured access lists.

Command Modes

(config)#	Global Configuration Mode

Command History

Release 2.1	Command was introduced

Functional Notes

ADTRAN OS access control policies are used to allow, discard, or manipulate (using NAT) data for each physical interface. Each ACP consists of a selector (access list) and an action (allow, discard, NAT). When packets are received on an interface, the configured ACPs are applied to determine whether the data will be processed or discarded.

The following commands are contained in the **policy-class** command set:

allow list <*access list names*>
All packets passed by the access list(s) entered will be allowed to enter the router system.

discard list <*access list names*>
All packets passed by the access list(s) entered will be dropped from the router system.

allow list <*access list names*> **policy** <*access policy name*>
All packets passed by the access list(s) entered and destined for the interface using the access policy listed will be permitted to enter the router system. This allows for configurations to permit packets to a single interface and not the entire system.

discard list <*access list names*> **policy** <*access policy name*>
All packets passed by the access list(s) entered and destined for the interface using the access policy listed will be blocked from the router system. This allows for configurations to deny packets on a specified interface.

nat source list <*access list names*> **address** <*IP address*> **overload**
All packets passed by the access list(s) entered will be modified to replace the source IP address with the entered IP address. The **overload** keyword allows multiple source IP addresses to be replaced with the single IP address entered. This hides private IP addresses from outside the local network.

nat source list <*access list names*> **interface** <*interface*> **overload**
All packets passed by the access list(s) entered will be modified to replace the source IP address with the primary IP address of the listed interface. The **overload** keyword allows multiple source IP addresses to be replaced with the single IP address of the specified interface. This hides private IP addresses from outside the local network.

nat destination list <*access list names*> **address** <*IP address*>
All packets passed by the access list(s) entered will be modified to replace the destination IP address with the entered IP address. The **overload** keyword is not an option when performing NAT on the destination IP address; each private address must have a unique public address. This hides private IP addresses from outside the local network.

Usage Examples

See the **Technology Review** (which follows) for command syntax examples.

Technology Review

Creating access policies and lists to regulate traffic through the routed network is a four-step process:

Step 1:

Enable the security features of the ADTRAN OS using the **ip firewall** command.

Step 2:

Create an access list to permit or deny specified traffic. Standard access lists provide pattern matching for source IP addresses only. (Use extended access lists for more flexible pattern matching.) IP addresses can be expressed in one of three ways:

1. Using the keyword **any** to match any IP address. For example, entering **deny any** will effectively shut down the interface that uses the access list because all traffic will match the **any** keyword.
2. Using the **host** <A.B.C.D> to specify a single host address. For example, entering **permit 196.173.22.253** will allow all traffic from the host with an IP address of 196.173.22.253.
3. Using the <A.B.C.D> <wildcard> format to match all IP addresses in a “range”. Wildcard masks work in reverse logic from subnet mask. Specifying a one in the wildcard mask equates to a “don’t care”. For example, entering **discard 192.168.0.0 0.0.0.255** will discard all traffic from the 192.168.0.0/24 network.

Step 3:

Create an access policy that uses a configured access list. ADTRAN OS access policies are used to allow, discard, or manipulate (using NAT) data for each physical interface. Each ACP consists of a selector (access list) and an action (allow, discard, NAT). When packets are received on an interface, the configured ACPs are applied to determine whether the data will be processed or discarded. Possible actions performed by the access policy are as follows:

allow list <*access list names*>

discard list <*access list names*>

allow list <*access list names*> **policy** <*access policy name*>

discard list <*access list names*> **policy** <*access policy name*>

nat source list <*access list names*> **address** <*IP address*> **overload**

nat source list <*access list names*> **interface** <*interface*> **overload**

nat destination list <*access list names*> **address** <*IP address*>

Step 4:

Apply the created access policy to an interface. To assign an access policy to an interface, enter the interface configuration mode for the desired interface and enter **access policy** <*policy name*>. The following example assigns access policy **MatchAll** to the ethernet 0/1 interface:

(config)# **interface ethernet 0/1**

(config-eth 0/1)# **access-policy MatchAll**

ip policy-timeout *<protocol> <port> <seconds>*

Use multiple **ip policy-timeout** commands to customize timeout intervals for protocols (TCP UDP ICMP) or specific services (by listing the particular port number). Use the **no** form of this command to return to the default timeout values.

Syntax Description

<protocol>	Specifies the data protocol such as ICMP, TCP, or UDP.
<port> ***Optional*	Service port to apply the timeout value to; valid only for specifying TCP and UDP services.

The following is the list of UDP port numbers that may be identified using the text name (in **bold**):

all_ports
biff (Port 512)
bootpc (Port 68)
bootps(Port 67)
discard (Port 9)
dnsix (Port 195)
domain (Port 53)
echo (Port 7)
isakmp (Port 500)
mobile-ip (Port 434)
nameserver (Port 42)
netbios-dgm (Port 138)
netbios-ns (Port 137)
netbios-ss (Port 139)
ntp (Port 123)
pim-auto-rp (496)
rip (Port 520)
snmp (Port 161)
snmptrap (Port 162)
sunrpc (Port 111)
syslog (Port 514)
tacacs (Port 49)
talk (Port 517)
tftp (Port 69)
time (Port 37)
who (Port 513)
xdmcp (Port 177)

The following is the list of TCP port numbers that may be identified using the text name (in **bold**):

all_ports
bgp (Port 179)
chargen (Port 19)
cmd (Port 514)
daytime (Port 13)
discard (Port 9)
domain (Port 53)
echo (Port 7)
exec (Port 512)
finger (Port 79)
ftp (Port 21)
login (Port 513)
lpd (Port 515)
nntp (Port 119)
pim-auto-rp (Port 496)
pop2 (Port 109)
pop3 (Port 110)
smtp (Port 25)
sunrpc (Port 111)
syslog (Port 514)
tacacs (Port 49)
talk (Port 517)

Syntax Description (Continued)

<*port*> ***Optional**	ftp-data (Port 20)	**tftp** (Port 69)
	gopher (Port 70)	**telnet** (Port 23)
	hostname (Port 101)	**time** (Port 37)
	ident (Port 113)	**uucp** (Port 540)
	irc (Port 194)	**whois** (Port 43)
	klogin (Port 543)	**www** (Port 80)
	kshell (Port 544)	
<*seconds*>	Wait interval (in seconds) before an active session is closed (valid range: 0 to 4294967295 seconds).	

Default Values

<*seconds*>	The following default policy timeout intervals apply: **tcp** (600 seconds; 10 minutes) **udp** (60 seconds; 1 minute) **icmp** (60 seconds; 1 minute)

Command Modes

(config)#	Global Configuration Mode

Command History

Release 2.1	Command was introduced

Usage Examples

The following example creates customized policy timeouts for the following:
internet traffic (TCP Port 80) timeout 24 hours (86400 seconds)
telnet (TCP Port 23) timeout 20 minutes (1200 seconds)
FTP (21) timeout 5 minutes (300 seconds)
All other TCP services timeout 8 minutes (480 seconds)

(config)# **ip policy-timeout tcp www 86400**
(config)# **ip policy-timeout tcp telnet 1200**
(config)# **ip policy-timeout tcp ftp 300**
(config)# **ip policy-timeout tcp all_ports 480**

ip route *<ip address> <subnet mask> <interface or ip address>*

Use the **ip route** command to add a static route to the route table. This command can be used to add a default route by entering **ip route 0.0.0.0 0.0.0.0** and specifying the interface or IP address. Use the **no** form of this command to remove a configured static route.

Syntax Description

<ip address>	Specifies the network address (in dotted decimal notation) to add to the route table.
<subnet mask>	Specifies the subnet mask (in dotted decimal notation) associated with the listed network IP address.
<interface or ip address>	Specifies the gateway peer IP address (in dotted decimal notation) or a configured interface in the unit. Valid interfaces include: virtual frame relay sub-interfaces (fr 1.16) and virtual PPP interfaces (ppp 1).

Default Values

By default, there are no configured routes in route table.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example adds a static route to the **10.220.0.0/16** network through the next-hop router **192.22.45.254** and a default route to **175.44.2.10**:

(config)# **ip route 10.220.0.0 255.255.0.0 192.22.45.254**
(config)# **ip route 0.0.0.0 0.0.0.0 175.44.2.10**

ip routing

Use the **ip routing** command to enable the ADTRAN OS IP routing functionality. Use the **no** form of this command to disable IP routing.

Syntax Description

No subcommands

Default Values

By default, IP routing is enabled.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example enables the ADTRAN OS IP routing functionality:

(config)# **ip routing**

ip snmp agent

Use the **ip snmp agent** command to enable the Simple Network Management Protocol (SNMP) agent.

Syntax Description

No subcommands

Default Values

By default, the SNMP agent is disabled.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

Allows a MIB browser to access standard MIBs within the product. This also allows the product to send traps to a trap management station.

Usage Examples

The following example enables the IP SNMP agent:

(config)# **ip snmp agent**

ip subnet-zero

The **ip subnet-zero** command is the default operation and cannot be disabled. This command signifies the router's ability to route to subnet-zero subnets.

Syntax Description

No subcommands.

Default Values

By default, this command is enabled.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 3.1	Command was introduced

Usage Examples

The following example **subnet-zero** is enabled:

(config)# **ip subnet-zero**

line [**console** | **telnet**] *<line-number> <ending number>*

Use the **line** command to enter the line configuration command set for the specified console or telnet session. This command can be used to add a route to the default gateway by entering **ip route 0.0.0.0 0.0.0.0** and specifying the interface or IP address. See the sections *Line (Console) Interface Config Command Set* on page 516 and *Line (Telnet) Interface Config Command Set* on page 526 for information on the subcommands found in these command sets.

Syntax Description

console	Specifies the DB-9 (female) **CONSOLE** port located on the rear panel of the unit. See the sections *Line (Console) Interface Config Command Set* on page 516 for information on the subcommands found in this command set.
telnet	Specifies a Telnet session(s) to configure for remote access . See the section *Line (Telnet) Interface Config Command Set* on page 526 for information on the subcommands found in this command set.
<line-number>	Specifies the starting Telnet or console session to configure for remote access (valid range for console: 0; valid range for Telnet: 0 to 4). If configuring a single Telnet session, enter the Telnet session number and leave the *<ending number>* field blank.
<ending number> ***Optional**	Specifies the last Telnet session to configure for remote access (valid range: 0 to 4). To configure all available Telnet sessions, enter **line telnet 0 4**.

Default Values

By default, the ADTRAN OS line console parameters are configured as follows:

Data Rate: 9600
Data bits: 8
Stop bits:1
Parity Bits:0
No flow control

By default, there are no configured Telnet sessions.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example begins the configuration for the **CONSOLE** port located on the rear of the unit:

(config)# **line console 0**
(config-con0)#

The following example begins the configuration for all available Telnet sessions:

(config)# **line telnet 0 4**
(config-telnet0-4)#

logging email address-list <*email address*> ; <*email address*>

Use the **logging email** command to specify one or more email addresses that will receive notification when an event matching the criteria configured using the **logging email priority-level** command is logged by the ADTRAN OS. See *logging email priority-level <priority>* on page 221 for more information. Use the **no** form of this command to remove a listed address.

Syntax Description

<*email address*>	Specifies the complete email address to use when sending logged messages (This field allows up to 256 characters.) Enter as many email addresses as desired, placing a semi-colon (;) between addresses.

Default Values

By default, there are no configured logging email addresses.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example specifies three email addresses to use when sending logged messages:

(config)# **logging email address-list admin@adtran.com;ntwk@adtran.com;support@adtran.com**

logging email on

Use the **logging email on** command to enable the ADTRAN OS email event notification feature. Use the **logging email address-list** command to specify email address(es) that will receive notification when an event matching the criteria configured using the **logging email priority-level** command is logged by the ADTRAN OS. See *logging email address-list <email address> ; <email address>* on page 219 and *logging email priority-level <priority>* on page 221 for more information. Use the **no** form of this command to disable the email notification feature

Syntax Description

No subcommands

Default Values

By default, email event notification is disabled.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

The domain name is appended to the sender name when sending event notifications. See the command *ip domain-name <name>* on page 188 for related information.

Usage Examples

The following example enables the ADTRAN OS email event notification feature:

(config)# **logging email on**

logging email priority-level *<priority>*

Use the **logging email priority-level** command to set the threshold for events sent to the addresses specified using the **logging email address-list** command. All events with the specified priority or higher will be sent to all addresses in the list. The logging email on command must be enabled. See *logging email address-list <email address> ; <email address>* on page 219 and *logging email on* on page 220 for related information. Use the **no** form of this command to return to the default priority.

Syntax Description

<priority>	Sets the minimum priority threshold for sending messages to email addresses specified using the **logging email address-list** command. The following priorities are available (ranking from lowest to highest): **4 - Information** When priority 4 is selected, all events (priorities 0 through 4) are logged. **3 - Notice** When priority 3 is selected, events with priority 3, 2, 1, or 0 are logged. **2 - Warning** When priority 2 is selected, events with priority 2, 1, or 0 are logged. **1 - Error** When priority 1 is selected, events with priority 1 or 0 are logged. **0 - Fatal** When priority 0 is selected, only events with priority 0 are logged.

Default Values

<priority>	2 - Warning

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example sends all messages with **Warning** level or greater to the email addresses listed using the **logging email address-list** command:

(config)# **logging email priority-level 2**

logging email receiver-ip *<ip address>*

Use the **logging email receiver-ip** command to specify the IP address of the email server to use when sending notification that an event matched the criteria configured using the **logging email priority-level** command. See *logging email priority-level <priority>* on page 221 for related information. Use the **no** form of this command to remove a configured address.

Syntax Description

<ip address>	Specifies the IP address (in dotted decimal notation) of the mail server to use when sending logged messages.

Default Values

By default, there are no configured email server addresses.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example specifies an email server (with address 172.5.67.99) to use when sending logged messages:

(config)# **logging email receiver-ip 172.5.67.99**

logging facility *<facility type>*

Use the **logging facility** command to specify a syslog facility type for the syslog server. Error messages meeting specified criteria are sent to the syslog server. For this service to be active, you must enable log forwarding. See *logging forwarding on* on page 224 for related information. The different facility types are described under **Functional Notes** below. Use the **no** form of this command to return this command to its default setting.

Syntax Description

<facility type>	Enter the syslog facility type (see **Functional Notes** below).

Default Values

The default value is local7

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 3.1	Command was introduced

Functional Notes

The following is a list of all the valid facility types:

auth	Authorization system
cron	Cron facility
daemon	System daemon
kern	Kernel
local0 - local7	Reserved for locally defined messages
lpr	Line printer system
mail	Mail system
news	USENET news
sys9 - sys14	System use
syslog	System log
user	User process
uucp	UNIX-to-UNIX copy system

Usage Examples

The following example configures the syslog facility to the cron facility type:

(config)#**logging facility cron**

logging forwarding on

Use the **logging forwarding on** command to enable the ADTRAN OS syslog event feature. Use the **logging forwarding priority-level** command to specify the event matching the criteria used by the ADTRAN OS to determine whether a message should be forwarded to the syslog server. See *logging forwarding priority-level <priority>* on page 225 for related information. Use the **no** form of this command to disable the syslog event feature.

Syntax Description

No subcommands

Default Values

By default, syslog event notification is disabled.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example enables the ADTRAN OS syslog event feature:

(config)# **logging forwarding on**

logging forwarding priority-level <*priority*>

Use the **logging forwarding priority-level** command to set the threshold for events sent to the configured syslog server specified using the **logging forwarding receiver-ip** command. All events with the specified priority or higher will be sent to all configured syslog servers. See *logging forwarding receiver-ip <ip address>* on page 226 for related information. Use the **no** form of this command to return to the default priority.

Syntax Description

<*priority*>	Sets the minimum priority threshold for sending messages to the syslog server specified using the **logging forwarding receiver-ip** command. The following priorities are available (ranking from lowest to highest): **4 - Information** When priority 4 is selected, all events (priorities 0 through 4) are logged. **3 - Notice** When priority 3 is selected, events with priority 3, 2, 1, or 0 are logged. **2 - Warning** When priority 2 is selected, events with priority 2, 1, or 0 are logged. **1 - Error** When priority 1 is selected, events with priority 1 or 0 are logged. **0 - Fatal** When priority 0 is selected, only events with priority 0 are logged.

Default Values

<*priority*>	2 - Warning

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example sends all messages with **Warning** level or greater to the syslog server listed using the **logging forwarding receiver-ip** command.

(config)# **logging forwarding priority-level 2**

logging forwarding receiver-ip *<ip address>*

Use this command to specify the IP address of the syslog server to use when logging events that match the criteria configured using the **logging forwarding priority-level** command. Enter multiple **logging forwarding receiver-ip** commands to develop a list of syslog servers to use. See *logging forwarding priority-level <priority>* on page 225 for related information. Use the **no** form of this command to remove a configured address.

Syntax Description

<ip address>	Specifies the IP address (in dotted decimal notation) of the syslog server to use when logging messages.

Default Values

By default, there are no configured syslog server addresses.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example specifies a syslog server (with address 172.5.67.99) to use when logging messages:

(config)# **logging forwarding receiver-ip 172.5.67.99**

router ospf

Use the **router ospf** command to activate OSPF in the router and to enter the OSPF Configuration Mode. See the section *Router (OSPF) Configuration Command Set* on page 541 for more information. Use the **no** form of this command to disable OSPF routing.

Syntax Description

No subcommands.

Default Values

By default, OSPF is disabled.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 3.1	Command was introduced

Functional Notes

The AOS can be configured to use OSPF with the firewall enabled (using the **ip firewall** command). To do this, configure the OSPF networks as usual, specifying which networks the system will listen for and broadcast OSPF packets to. See *ip firewall* on page 190 for more information.

To apply stateful inspection to packets coming into the system, create a policy-class that describes the type of action desired and then associate that policy-class to the particular interface (see *ip policy-class <policyname> max-sessions* on page 208). The firewall is intelligent and will only allow OSPF packets that were received on an OSPF configured interface. No modification to the policy-class is required to allow OSPF packets into the system.

Usage Examples

The following example uses the **router ospf** command to enter the OSPF Configuration Mode:

```
(config)# router ospf
(config-ospf)#
```

router rip

Use the **router rip** command to enter the RIP Configuration Mode. See the section *Router (RIP) Configuration Command Set* on page 532 for more information.

Syntax Description

No subcommands.

Default Values

No default values necessary for this command.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example uses the **router rip** command to enter the RIP Configuration Mode:

```
(config)# router rip
(config-rip)#
```


Technology Review

The RIP protocol is based on the Bellham-Ford (distance-vector) algorithm. This algorithm provides that a network will converge to the correct set of shortest routes in a finite amount of time, provided that:

- Gateways continuously update their estimates of routes.
- Updates are not overly delayed and are made on a regular basis.
- The radius of the network is not excessive.
- No further topology changes take place.

RIP is described in RFC 1058 (Version 1) and updated in RFCs 1721, 1722, and 1723 for Version 2. Version 2 includes components that ease compatibility in networks operating with RIP V1.

All advertisements occur on regular intervals (every 30 seconds). Normally, a route that is not updated for 180 seconds is considered dead. If no other update occurs in the next 60 seconds for a new and better route, the route is flushed after 240 seconds. Consider a connected route (one on a local interface). If the interface fails, an update is immediately triggered for that route only (advertised with a metric of 16).

Now consider a route that was learned and does not receive an update for 180 seconds. The route is marked for deletion, and even if it was learned on an interface, a poisoned (metric =16) route should be sent by itself immediately and during the next two update cycles with the remaining normal split horizon update routes. Following actual deletion, the poison reverse update ceases. If an update for a learned route is not received for 180 seconds, the route is marked for deletion. At that point, a 120-second garbage collection (GC) timer is started. During the GC timer, expiration updates are sent with the metric for the timed out route set to 16.

If an attached interface goes down, the associated route is immediately (within the same random five-second interval) triggered. The next regular update excludes the failed interface. This is the so-called first hand knowledge rule. If a gateway has first hand knowledge of a route failure (connected interfaces) or reestablishment, the same action is taken. A triggered update occurs, advertising the route as failed (metric = 16) or up (normal metric) followed by the normal scheduled update.

The assumption here is that if a gateway missed the triggered update, it will eventually learn from another gateway in the standard convergence process. This conserves bandwidth.

RIP-Related Definitions:

- Route - A description of the path and its cost to a network.
- Gateway - A device that implements all or part of RIP - a router.
- Hop - Metric that provides the integer distance (number of intervening gateways) to a destination network gateway.
- Advertisement - A broadcast or multicast packet to port 520 that indicates the route for a given destination network.
- Update - An advertisement sent on a regular 30-second interval including all routes exclusive of those learned on an interface.

snmp-server chassis-id *<id string>*

Use the **snmp-server chassis-id** command to specify an identifier for the Simple Network Management Protocol (SNMP) server. Use the **no** form of this command to return to the default value.

Syntax Description

<id string>	Alphanumeric string (up to 32 characters in length) used to identify the product.

Default Values

<id string>	Chassis ID

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example configures a chassis ID of **A432692**:

(config)# **snmp-server chassis-id A432692**

snmp-server community <*community*> [**ro** | **rw**] <*listname*>]

Use the **snmp-server community** command to specify a community string to control access to the Simple Network Management Protocol (SNMP) information. Use the **no** form of this command to remove a specified community.

Syntax Description

<*community*>	Specifies the community string (a password to grant SNMP access).
ro ***Optional**	Keyword to grant read-only access, allowing retrieval of MIB objects.
rw ***Optional**	Keyword to grant read-write access, allowing retrieval and modification of MIB objects.
<*listname*> ***Optional**	Access-control list name used to limit access. See *ip access-list extended <listname>* on page 170 and *ip access-list standard <listname>* on page 176 for more information on creating access-control lists.

Default Values

By default, there are no configured SNMP communities.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example defines a community named **MyCommunity** and assigns read-write access:

(config)# **snmp-server community MyCommunity rw**

snmp-server contact <*string*>

Use the **snmp-server contact** command to specify the SNMP sysContact string. Use the **no** form of this command to remove a configured contact.

Syntax Description

"<string>"	Alphanumeric string encased in quotes (up to 32 characters in length) used to populate the sysContact string.

Default Values

<*string*>	Customer Service

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example specifies **Network Administrator x4000** for the sysContact string:

(config)# **snmp-server contact "Network Administrator x4000"**

snmp-server enable traps *<trap type>*

Use the **snmp-server enable traps** command to enable all Simple Network Management Protocol (SNMP) traps available on your system or specified using the *<trap type>* option. Use multiple **snmp-server enable traps** to enable multiple trap types. Use the **no** form of this command to disable traps (or the specified traps).

Syntax Description

<trap type> ***Optional*	Specifies the type of notification trap to enable. Leaving this option blank enables ALL system traps.
snmp	Enables a subset of traps specified in RFC 1157 The following traps are supported: coldStart warmStart linkUp linkDown authenticationFailure

Default Values

By default, there are no enabled traps.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example enables the SNMP traps:

(config)# **snmp-server enable traps snmp**

snmp-server host <*address*> **traps** <*community*> <*trap type*>

Use the **snmp-server host traps** command to specify traps sent to an identified host. Use multiple **snmp-server host traps** commands to specify all desired hosts. Use the **no** form of this command to return to the default value.

Syntax Description

<*address*>	Specifies the IP address of the SNMP host that receives the traps.
<*community*>	Specifies the community string (used as a password) for authorized agents to obtain access to SNMP information.
<*trap type*> ***Optional***	Specifies the type of notification trap to enable. Leaving this option blank enables ALL system traps.
snmp	Enables a subset of traps specified in RFC 1157. The following traps are supported: coldStart warmStart linkUp linkDown authenticationFailure

Default Values

By default, there are no hosts or traps enabled.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example sends all SNMP traps to the host at address **190.3.44.69** and community string **My Community**:

(config)# **snmp-server host 190.3.44.69 traps My Community snmp**

snmp-server host *<address>* **traps version** *<version> <community> <trap type>*

Use the **snmp-server host traps version** command to specify traps sent to an identified host. Use multiple **snmp-server host traps version** commands to specify all desired hosts. Use the **no** form of this command to return to the default value.

Syntax Description

<address>	Specifies the IP address of the SNMP host that receives the traps.
<version>	Specifies the SNMP version as one of the following: **1** - SNMPv1 **2C** - SNMPv2C
<community>	Specifies the community string (used as a password) for authorized agents to obtain access to SNMP information.
<trap type> ***Optional*	Specifies the type of notification trap to enable. Leaving this option blank enables ALL system traps.
snmp	Enables a subset of traps specified in RFC 1157. The following traps are supported: coldStart warmStart linkUp linkDown authenticationFailure

Default Values

By default, there are no hosts or traps enabled.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example sends all SNMP traps to the host at address **190.3.44.69** and community string **My Community** using SNMPv2C:

(config)# **snmp-server host 190.3.44.69 traps version 2c My Community snmp**

snmp-server location <*string*>

Use the **snmp-server location** command to specify the Simple Network Management Protocol (SNMP) system location string. Use the **no** form of this command to return to the default value.

Syntax Description

"<string>"	Alphanumeric string encased in quotation marks (up to 32 characters in length) used to populate the system location string.

Default Values

<*string*>	ADTRAN

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example specifies a location of **5th Floor Network Room**:

(config)# **snmp-server location "5th Floor Network Room"**

snmp-server trap-source *<interface>*

Use the **snmp-server trap-source** command to tell the AOS where to expect SNMP traps to originate from (interface type). Use the **no** form of this command to remove specified interfaces.

Syntax Description

<interface>	Specifies the physical interface that should originate SNMP traps. Enter **snmp-server trap-source ?** for a complete list of valid interfaces.

Default Values

By default, there are no trap-source interfaces defined.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example specifies that the Ethernet interface (**ethernet 0/1**) should be the source for all SNMP traps:

(config)# **snmp-server trap-source ethernet 0/1**

sntp server *<address or hostname>* **version** *<1-3>*

Use the **sntp server** command to set the hostname of the SNTP server as well as the version of SNTP to use. The Simple Network Time Protocol (SNTP) is an abbreviated version of the Network Time Protocol (NTP). SNTP is used to set the time of the AOS product over a network. The SNTP server usually serves the time to many devices within a network.

Syntax Description

<address or hostname>	Specifies the IP address or hostname of the SNTP server.
version	Specifies which NTP version is used (1-3).

Default Values

By default, version is set to 1.

Command Modes

(config)#	Global Configuration Mode

Command History

Release 3.1	Command was introduced

Usage Examples

The following example sets the SNTP server to **time.nist.gov** using SNTP version 1 (the default version):

(config)# **sntp server time.nist.gov**

The following example sets the SNTP server as **time.nist.gov**. All requests for time use version 2 of the SNTP:

(config)#**sntp server time.nist.gov version 2**

username *<username>* **password** *<password>*

Use this command to configure the username and password to use for all protocols requiring a username-based authentication system including FTP server authentication, line (login local-user list), and HTTP access.

Syntax Description

<username>	Alphanumerical string up to 30 characters in length (the username is case-sensitive)
<password>	Alphanumerical string up to 30 characters in length (the username is case-sensitive)

Default Values

By default, there is no established username and password.

Command Modes

(config)#	Global Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

All users defined using the **username/password** command are valid for access to the unit using the **login local-userlist** command.

Usage Examples

The following example creates a username of **ADTRAN** with password **ADTRAN**:

(config)# **username ADTRAN password ADTRAN**

DHCP POOL COMMAND SET

To activate the DHCP Pool command set, enter the **ip dhcp-server pool** command at the Global Configuration Mode prompt. For example:

```
Router> enable
Router# configure terminal
Router(config)# ip dhcp-server pool MyPool
Router(config-dhcp)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

do on page 558

end on page 559

exit on page 560

All other commands for this command set are described in this section in alphabetical order.

client-identifier <identifier> on page 241

client-name <name> on page 243

default-router <address> <secondary> on page 244

dns-server <address> <secondary> on page 245

domain-name <domain> on page 246

hardware-address <hardware-address> <type> on page 247

host <address> [<subnet mask> or <prefix length>] on page 249

lease <days> <hours> <minutes> on page 250

netbios-name-server <address> <secondary> on page 251

netbios-node-type <type> on page 252

network <address> [<subnet mask> or <prefix length>] on page 253

client-identifier <*identifier*>

Use the **client-identifier** command to specify a unique identifier (in dotted hexadecimal notation) for a Dynamic Host Configuration Protocol (DHCP) client. Use the **no** form of this command to remove a configured client-identifier.

Syntax Description

<*identifier*>	Specify a custom client-identifier using a text string (that is converted to a hexadecimal equivalent) or 7 to 28 hexadecimal numbers (with colon delimiters). OR Specify the hexadecimal MAC address including a hexadecimal number added to the front of the MAC address to identify the media type. For example, specifying the **client-identifier** for a MAC address of d217.0491.1150 defines the client identifier as **01:d2:17:04:91:11:50** (where 01 defines the media type as Ethernet). For example, a custom client identifier of **0f:ff:ff:ff:ff:51:04:99:a1** may be entered using the <*identifier*> option.

Default Values

client-id ***Optional**	By default, the client identifier is populated using the following formula: TYPE: INTERFACE SPECIFIC INFO : MAC ADDRESS Where TYPE specifies the media type in the form of one hexadecimal byte (refer to *hardware-address <hardware-address> <type>* on page 247 for a detailed listing of media types) and the MAC ADDRESS is the Media Access Control (MAC) address assigned to the first Ethernet interface in the unit in the form of six hexadecimal bytes. (For units with a single Ethernet interface, the MAC ADDRESS assigned to ethernet 0/1 is used in this field). INTERFACE SPECIFIC INFO is only used for frame relay interfaces and can be determined using the following: FR_PORT# : Q.922 ADDRESS Where the FR_PORT# specifies the label assigned to the virtual frame relay interface using four hexadecimal bytes. For example, a virtual frame relay interface labeled 1 would have a FR_PORT# of 00:00:00:01.

Default Values (Continued)

The Q.922 ADDRESS field is populated using the following:

8	7	6	5	4	3	2	1
DLCI (high order)						C/R	EA
DLCI (lower)				FECN	BECN	DE	EA

Where the FECN, BECN, C/R, DE, and high order EA bits are assumed to be 0, and the lower order extended address (EA) bit is set to 1.

The following list provides a few example DLCIs and associated Q.922 addresses:

DLCI (decimal) / Q.922 address (hex)

16 / 0x0401
50 / 0x0C21
60 / 0x0CC1
70 / 0x1061
80 / 0x1401

Command Modes

(config-dhcp)#	DHCP Pool Command Set

Command History

Release 2.1	Command was introduced

Functional Notes

DHCP clients use client-identifiers in place of hardware addresses. To create the client-identifier, begin with the two-digit numerical code representing the media type and append the client's MAC address. For example, a Microsoft client with an Ethernet (01) MAC address d2:17:04:91:11:50 uses a client-identifier of 01:d2:17:04:91:11:50.

Usage Examples

The following example specifies the client-identifier for a Microsoft client with an Ethernet MAC address of d217.0491.1150:

(config)# **ip dhcp-server pool Microsoft_Clients**
(config-dhcp)# **client-identifier 01:d2:17:04:91:11:50**

client-name *<name>*

Use the **client-name** command to specify the name of a Dynamic Host Configuration Protocol (DHCP) client. Use the **no** form of this command to remove the configured client name.

Syntax Description

<name>	Alphanumeric string (up to 32 characters in length) used to identify the DHCP client (example is **client1**).

The specified client name should not contain the domain name.

Default Values

By default, there are no specified client names.

Command Modes

(config-dhcp)#	DHCP Pool Command Set

Command History

Release 2.1	Command was introduced

Usage Examples

The following example specifies a client name of **myclient**:

(config)# **ip dhcp-server pool MyPool**
(config-dhcp)# **client-name myclient**

default-router *<address> <secondary>*

Use the **default-router** command to specify the default primary and secondary routers to use for the Dynamic Host Configuration Protocol (DHCP) client. Use the **no** form of this command to remove the configured router.

Syntax Description

<address>	Specifies the address (in dotted decimal notation) of the preferred router on the client's subnet (example: 192.22.4.254).
<secondary> ***Optional***	Specifies the address (in dotted decimal notation) of the second preferred router on the client's subnet (example: 192.22.4.253).

Default Values

By default, there are no specified default routers.

Command Modes

(config-dhcp)#	DHCP Pool Command Set

Command History

Release 2.1	Command was introduced

Functional Notes

When specifying a router to use as the primary/secondary preferred router, verify that the listed router is on the same subnet as the DHCP client. The ADTRAN OS allows a designation for two routers, listed in order of precedence.

Usage Examples

The following example configures a default router with address **192.22.4.253** and a secondary router with address **192.22.4.254**:

(config)# **ip dhcp-server pool MyPool**
(config-dhcp)# **default-router 192.22.4.253 192.22.4.254**

dns-server *<address> <secondary>*

Use the **dns-server** command to specify the default primary and secondary Domain Name System (DNS) servers to use for the Dynamic Host Configuration Protocol (DHCP) client. Use the **no** form of this command to remove the configured DNS server.

Syntax Description

<address>	Specifies the address (in dotted decimal notation) of the preferred DNS server on the network (example: 192.72.4.254).
<secondary> ***Optional***	Specifies the address (in dotted decimal notation) of the second preferred DNS server on the network (example: 192.100.4.253).

Default Values

By default, there are no specified default DNS servers.

Command Modes

(config-dhcp)#	DHCP Pool Command Set

Command History

Release 2.1	Command was introduced

Usage Examples

The following example specifies a default DNS server with address **192.72.3.254** and a secondary DNS server with address **192.100.4.253**:

(config)# **ip dhcp-server pool MyPool**
(config-dhcp)# **dns-server 192.72.3.254 192.100.4.253**

domain-name <*domain*>

Use the **domain-name** command to specify the domain name for the Dynamic Host Configuration Protocol (DHCP) client. Use the **no** form of this command to remove the configured domain name.

Syntax Description

<*name*>	Alphanumeric string (up to 32 characters in length) used to identify the DHCP client (e.g., adtran.com).

Default Values

By default, there are no specified domain-names.

Command Modes

(config-dhcp)#	DHCP Pool Command Set

Command History

Release 2.1	Command was introduced

Usage Examples

The following example specifies a domain name of **adtran.com**:

(config)# **ip dhcp-server pool MyPool**
(config-dhcp)# **domain-name adtran.com**

hardware-address <*hardware-address*> <*type*>

Use the **hardware-address** command to specify the name of a Dynamic Host Configuration Protocol (DHCP) client. Use the **no** form of this command to remove the configured client name.

Syntax Description

<*hardware-address*>	Specifies the hardware address (in hexadecimal notation with colon delimiters) of the preferred router on the client's subnet (example d2:17:04:91:11:50).
<*type*> ***Optional**	Specifies the hardware protocol of the DHCP client. The hardware type field can be entered as follows:
ethernet	Specifies standard Ethernet networks.
ieee802	Specifies IEEE 802 standard networks.
<1-21>	Enter one of the hardware types listed in RFC 1700. The valid hardware types are as follows: 1 10 Mb Ethernet 2 Experimental 3 Mb Ethernet 3 Amateur Radio AX.25 4 Proteon ProNET Token Ring 5 Chaos 6 IEEE 802 Networks 7 ARCNET 8 Hyperchannel 9 Lanstar 10 Autonet Short Address 11 LocalTalk 12 LocalNet (IBM PCNet or SYTEK LocalNet) 13 Ultra link 14 SMDS 15 Frame Relay 16 Asynchronous Transmission Mode (ATM) 17 HDLC 18 Fibre Channel 19 Asynchronous Transmission Mode (ATM) 20 Serial Line 21 Asynchronous Transmission Mode (ATM)

Default Values

<*type*>	1 - 10 Mb Ethernet

Command Modes

(config-dhcp)#	DHCP Pool Command Set

Command History

Release 2.1	Command was introduced

Usage Examples

The following example specifies an Ethernet client with a MAC address of **ae:11:54:60:99:10**:

(config)# **ip dhcp-server pool MyPool**
(config-dhcp)# **hardware-address ae:11:54:60:99:10 Ethernet**

host <*address*> [<*subnet mask*> or <*prefix length*>]

Use the **host** command to specify the IP address and subnet mask for a manual binding to a Dynamic Host Configuration Protocol (DHCP) client. Use the **no** form of this command to remove the configured client address.

Syntax Description

<*address*>	Specifies the IP address (in dotted decimal notation) for a manual binding to a DHCP client.
<*subnet mask*> ***Optional**	Specifies the network mask (subnet) for a manual binding to a DHCP client. If the subnet mask is left unspecified, the DHCP server examines its address pools to obtain an appropriate mask. If no valid mask is found in the address pools, the DHCP server uses the Class A, B, or C natural mask.
<*prefix length*> ***Optional**	Alternately, the prefix length may be used to specify the number of bits that comprise the network address. The prefix length must be preceded by a forward slash (/). For example, to specify an IP address with a subnet mask of 255.255.0.0, enter **/16** after the address.

Default Values

By default, there are no specified host addresses.

Command Modes

(config-dhcp)#	DHCP Pool Command Set

Command History

Release 2.1	Command was introduced

Usage Examples

The following examples show two different ways to specify a client with IP address **12.200.5.99** and a 21-bit subnet mask:

(config)# **ip dhcp-server pool MyPool**
(config-dhcp)# **host 12.200.5.99 255.255.248.0**

or

(config)# **ip dhcp-server pool MyPool**
(config-dhcp)# **host 12.200.5.99 /21**

lease *<days> <hours> <minutes>*

Use the **lease** command to specify the duration of the lease for an IP address assigned to a Dynamic Host Configuration Protocol (DHCP) client. Use the **no** form of this command to return to the default lease value.

Syntax Description

<days>	Specifies the duration of the IP address lease in days.
<hours> ***Optional***	Specifies the number of hours in a lease. You may only enter a value in the hours field if the days field is specified.
<minutes> ***Optional***	Specifies the number of minutes in a lease. You may only enter a value in the minutes field if the days and hours fields are specified.

Default Values

By default, an IP address lease is one day.

Command Modes

(config-dhcp)#	DHCP Pool Command Set

Command History

Release 2.1	Command was introduced

Usage Examples

The following example specifies a lease of **2 days**:

(config)# **ip dhcp-server pool MyPool**
(config-dhcp)# **lease 2**

The following example specifies a lease of **1 hour**:

(config)# **ip dhcp-server pool MyPool**
(config-dhcp)# **lease 0 1**

The following example specifies a lease of **30 minutes**:

(config)# **ip dhcp-server pool MyPool**
(config-dhcp)# **lease 0 0 30**

netbios-name-server *<address> <secondary>*

Use the **netbios-name-server** command to specify the primary and secondary NetBIOS Windows Internet Naming Service (WINS) name servers available for use by the Dynamic Host Configuration Protocol (DHCP) clients. Use the **no** form of this command to remove a configured NetBIOS name server.

Syntax Description

<address>	Specifies the address (in dotted decimal notation) of the preferred NetBIOS WINS name server on the network (example: 192.72.4.254).
<secondary> ***Optional***	Specifies the address (in dotted decimal notation) of the second preferred NetBIOS WINS name server on the network (example: 192.100.4.253).

Default Values

By default, there are no configured NetBIOS WINS name servers.

Command Modes

(config-dhcp)#	DHCP Pool Command Set

Command History

Release 2.1	Command was introduced

Usage Examples

The following example specifies a primary NetBIOS WINS name server with an IP address of **172.45.6.99** and a secondary with an IP address of **172.45.8.15**:

(config)# **ip dhcp-server pool MyPool**
(config-dhcp)# **netbios-name-server 172.45.6.99 172.45.8.15**

netbios-node-type <*type*>

Use the **netbios-node-type** command to specify the type of NetBIOS node used with Dynamic Host Configuration Protocol (DHCP) clients. Use the **no** form of this command to remove a configured NetBIOS node type.

Syntax Description

<*type*>	Specifies the NetBIOS node type used with DHCP clients. Valid node types are as follows: **b-node** (1) - Broadcast node **p-node** (2) - Peer-to-Peer node **m-node** (4) - Mixed node **h-node** (8) - Hybrid node (Recommended) Alternately, the node type can be specified using the numerical value listed next to the nodes above (valid range: 1 to 8).

Default Values

<*type*>	**h-node** (8) - Hybrid node

Command Modes

(config-dhcp)#	DHCP Pool Command Set

Command History

Release 2.1	Command was introduced

Usage Examples

The following example specifies a client's NetBIOS node type as **h-node**:

```
(config)# ip dhcp-server pool MyPool
(config-dhcp)# netbios-node-type h-node
```

Alternately, the following also specifies the client's NetBIOS node type as **h-node**:

```
(config-dhcp)# netbios-node-type 8
```


network <*address*> [<*subnet mask*> or <*prefix length*>]

Use the **network** command to specify the subnet number and mask for an ADTRAN OS Dynamic Host Configuration Protocol (DHCP) server address pool. Use the **no** form of this command to remove a configured subnet.

Syntax Description

<*ip address*>	Specifies the IP address (in dotted decimal notation) of the DHCP address pool.
<*subnet mask*> ***Optional**	Specifies the network mask (subnet) for the address pool. If the subnet mask is left unspecified, the DHCP server uses the Class A, B, or C natural mask.
<*prefix length*> ***Optional**	Alternately, the prefix length may be used to specify the number of bits that comprise the network address. The prefix length must be preceded by a forward slash (/). For example, to specify an IP address with a subnet mask of 255.255.0.0, enter **/16** after the address.

Default Values

By default, there are no configured DHCP address pools.

Command Modes

Any Configuration Mode

Command History

Release 2.1	Command was introduced

Usage Examples

The following examples show two different ways to configure an address pool subnet of **192.34.0.0** with a 16-bit subnet mask:

(config)# **ip dhcp-server pool MyPool**
(config-dhcp)# **network 192.34.0.0 255.255.0.0**

or

(config)# **ip dhcp-server pool MyPool**
(config-dhcp)# **network 192.34.0.0 /16**

IKE POLICY COMMAND SET

To activate the IKE Policy command set, enter the **crypto ike policy** command at the Global Configuration Mode prompt. For example:

```
Router> enable
Router# configure terminal
Router(config)# crypto ike policy 1
Router(config-ike)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

do on page 558

end on page 559

exit on page 560

All other commands for this command set are described in this section in alphabetical order.

attribute <policynumber> on page 255

client configuration pool <poolname> on page 256

initiate [main | aggressive] on page 260

local-id [address | fqdn | user-fqdn] <ipaddress or domain name> on page 261

peer [<ip address> | any] on page 263

respond [main | aggressive | anymode] on page 265

*For VPN configuration example scripts, refer to the technical support note **Configuring VPN** located on the **NetVanta 3000 Series System Manual** CD provided with your unit.*

attribute *<policynumber>*

Use the **attribute** command to define attributes for the associated IKE policy. Multiple attributes can be created for a single IKE policy. Once you enter this command, you are in the IKE Policy Attribute command set. Refer to *IKE Policy Attributes Command Set* on page 270 for more information.

Syntax Description

<policynumber>	Assign a number (range: 1-65535) to the attribute policy. The number is the attribute's priority number and specifies the order in which the resulting VPN proposals get sent to the far-end. This command takes you to the **(config-ike-attribute)#** prompt. From here, you can configure the settings for the attribute as outlined in the section *IKE Policy Attributes Command Set* on page 270.

Default Values

By default, no attribute is defined.

Command Modes

(config-ike)#	IKE Policy Configuration Mode

Command History

Release 4.1	Command was introduced

Functional Notes

Multiple attributes on an IKE policy are ordered by number (with the lowest number representing the highest priority).

Usage Examples

The following example defines a policy attribute (**10**) and takes you into the IKE Policy Attributes command set:

```
(config-ike)# attribute 10
(config-ike-attribute)#
```

client configuration pool <*poolname*>

Use the **client configuration pool** command to configure the AOS to perform as mode-config server (edge device) when an IKE policy is negotiated.

Syntax Description

<*poolname*>	The pool from which to obtain parameters to assign to the client.

Default Values

By default, if this command is not present in the IKE policy, the ADTRAN device allocates mode-config IP addresses, DNS server addresses, and NetBIOS name server addresses, and mode-config is not performed.

Command Modes

(config-ike)#	IKE Policy Configuration Mode

Command History

Release 4.1	Command was introduced

Functional Notes

This command ties an existing client configuration pool to an IKE policy.

Usage Examples

The following example ties the **ConfigPool1** configuration pool to this IKE policy:

(config-ike)# **client configuration pool ConfigPool1**

Technology Review

The following example configures an ADTRAN OS product for VPN using IKE aggressive mode with pre-shared keys and mode config support (i.e., IPv4 address, primary and secondary DNS, and NBNS addresses). The ADTRAN OS product can be set to initiate IKE negotiation in main mode or aggressive mode. The product can be set to respond to IKE negotiation in main mode, aggressive mode, or any mode. In this example, the device is configured to initiate in aggressive mode and to respond to any mode.

This example assumes that the ADTRAN OS product has been configured with a WAN IP Address of 192.168.1.1 on interface **ethernet 0/1** and a LAN IP Address of 10.10.10.254 on interface **ethernet 0/2**. The Peer Private IP Subnet is 10.10.20.0.

For more detailed information on VPN configuration, refer to the technical support note *Configuring VPN* located on the ***NetVanta 3000 Series System Manual*** CD provided with your unit.

Step 1:
Enter the Global configuration mode (i.e., config terminal mode).

```
>enable
#configure terminal
```

Step 2:
Enable VPN support using the **ip crypto** command. This command allows crypto maps to be applied to interfaces, and enables the IKE server to listen for IKE negotiation sessions on UDP port 500.

```
(config)#ip crypto
```

Step 3:
Set the local ID. During IKE negotiation, local-ids are exchanged between the local device and the peer device. In the ADTRAN OS, the default setting for all local-ids is configured by the **crypto ike local-id** command. The default setting is for all local-ids to be the IPv4 address of the interface over which the IKE negotiation is occurring. In the future, a unique system-wide Hostname or Fully Qualified Domain Name could be used for all IKE negotiation.

```
(config)#crypto ike local-id address
```

Step 4:
Create IKE policy. In order to use IKE negotiation, an IKE policy must be created. Within the system, a list of IKE policies is maintained. Each IKE policy is given a priority number in the system. That priority number defines the position of that IKE policy within the system list. When IKE negotiation is needed, the system searches through the list, starting with the policy with priority of 1, looking for a match to the peer IP address.

An individual IKE policy can override the system local-id setting by having the **local-id** command specified in the IKE policy definition. This command in the IKE policy is used to specify the type of local-id and the local-id data. The type can be of IPv4 address, Fully Qualified Domain Name, or User-Specified Fully Qualified Domain Name.

An IKE policy may specify one or more peer IP addresses that will be allowed to connect to this system. To specify multiple unique peer IP addresses, the **peer A.B.C.D** command is used multiple times within a single IKE policy. To specify that all possible peers can use a default IKE policy, the **peer any** command is given instead of the **peer A.B.C.D** command inside of the IKE policy. The policy with the **peer any** command specified will match to any peer IP address (and therefore should be given the highest numerical priority number). This will make the policy the last one to be compared against during IKE negotiation.

Technology Review (Continued)

```
(config)#crypto ike policy 10
(config-ike)#no local-id
(config-ike)#peer 192.168.1.2
(config-ike)#initiate aggressive
(config-ike)#respond anymode
(config-ike)#client configuration pool vpn_users
(config-ike)#attribute 10
(config-ike-attribute)#encryption 3des
(config-ike-attribute)#hash sha
(config-ike-attribute)#authentication pre-share
(config-ike-attribute)#group 1
(config-ike-attribute)#lifetime 900
```

Step 5:
Define the remote-id settings. The **crypto ike remote-id** command is used to specify the remote-id for a peer connecting to the system. This command is also used to specify the preshared-key associated with the specific remote-id. The **crypto ike remote-id** command is used to define the remote-id for a peer connecting to the system, specify the preshared-key associated with the specific remote-id, and (optionally) determine that the peer matching this remote-id should not use mode config (by using the **no-mode-config** keyword). See *crypto ike remote-id* on page 154 for more information.

```
(config)#crypto ike remote-id address 192.168.1.2 preshared-key
mysecret123
```

Step 6:
Define the transform-set. A transform-set defines the encryption and/or authentication algorithms to be used to secure the data transmitted over the VPN tunnel. Multiple transform-sets may be defined in a system. Once a transform-set is defined, many different crypto maps within the system can reference it. In this example, a transform-set named **highly_secure** has been created. This transform-set defines ESP with Authentication implemented using 3DES encryption and SHA1 authentication.

```
(config)#crypto ipsec transform-set highly_secure esp-3des esp-sha-hmac
(cfg-crypto-trans)#mode tunnel
```

Step 7:
Define an ip-access list. An Extended Access Control List is used to specify which traffic needs to be sent securely over the VPN tunnel. The entries in the list are defined with respect to the local system. The source IP address will be the source of the traffic to be encrypted. The destination IP address will be the receiver of the data on the other side of the VPN tunnel.

```
(config)#ip access-list extended corporate_traffic
(config-ext-nacl)#permit ip 10.10.10.0 0.0.0.255 10.10.20.0 0.0.0.255 log
  deny ip any any
```


Technology Review (Continued)

Step 8:
Create crypto map. A Crypto Map is used to define a set of encryption schemes to be used for a given interface. A crypto map entry has a unique index within the crypto map set. The crypto map entry will specify whether IKE is used to generate encryption keys or if manually specified keys will be used. The crypto map entry will also specify who will be terminating the VPN tunnel, as well as which transform-set or sets will be used to encrypt and/or authenticate the traffic on that VPN tunnel. It also specifies the lifetime of all created IPSec Security Associations.

```
(config)#crypto map corporate_vpn 1 ipsec-ike
(config-crypto-map)#match address corporate_traffic
(config-crypto-map)#set peer 192.168.1.2
(config-crypto-map)#set transform-set highly_secure
(config-crypto-map)#set security-association lifetime kilobytes 8000
(config-crypto-map)#set security-association lifetime seconds 86400
(config-crypto-map)#no set pfs
```

Step 9:
Configure public interface. This process includes configuring the IP address for the interface and applying the appropriate crypto map to the interface. Crypto maps are applied to the interface on which encrypted traffic will be transmitted.

```
(config)#interface ethernet 0/1
(config-eth 0/1)#ip address 192.168.1.1 255.255.255.0
(config-eth 0/1)#crypto map corporate_vpn
(config-eth 0/1)#no shutdown
```

Step 10:
Configure private interface and add a static route to allow all traffic destined for the VPN tunnel to be routed to the appropriate gateway.

```
(config)#interface ethernet 0/2
(config-eth 0/2)#ip address 10.10.10.254 255.255.255.0
(config-eth 0/2)#no shutdown
(config-eth 0/2)#exit
(config)#ip route 10.10.20.0 255.255.255.0 192.168.1.2
```

initiate [main | aggressive]

Use the **initiate** command to allow the IKE policy to initiate negotiation (in main mode or aggressive mode) with peers. Use the **no** form of this command to allow the policy to respond only.

Syntax Description

main	Specify to initiate using main mode. Main mode requires that each end of the VPN tunnel has a static WAN IP address. Main mode is more secure than aggressive mode because more of the main mode negotiations are encrypted.
aggressive	Specify to initiate using aggressive mode. Aggressive mode can be used when one end of the VPN tunnel has a dynamically assigned address. The side with the dynamic address has to be the initiator of the traffic and tunnel. The side with the static address has to be the responder.

Default Values

By default, initiate in main mode is enabled.

Command Modes

(config-ike)#	IKE Policy Configuration Mode

Command History

Release 4.1	Command was introduced

Functional Notes

By using the **initiate** and **respond** commands, you can configure the IKE policy to initiate and respond, initiate only, or respond only. It is an error if you have both **initiate** and **respond** disabled.

Usage Examples

The following example enables the AOS device to initiate IKE negotiation in main mode:

(config-ike)# **initiate main**

local-id [**address | fqdn | user-fqdn**] <*ipaddress or domain name*>

Use the **local-id** command to set the local ID for the IKE policy. This setting overrides the system local ID setting (set in the Global command set using the **crypto ike local-id address** command).

Syntax Description

address <*ipaddress*>	Specifies a remote ID of IPv4 type.
fqdn <*domain name*>	Specifies a fully qualified domain name (e.g., adtran.com) as the remote ID.
user-fqdn <*domain name*>	Specifies a user fully qualified domain name (e.g., user1@adtran.com) as the remote ID.

Default Values

By default, local-id is not defined.

Command Modes

(config-ike)#	IKE Policy Configuration Mode

Command History

Release 4.1	Command was introduced

Functional Notes

The local-id for a particular IKE policy can be set in two ways. The first (default) method is to use the global system command:
(config)#**crypto ike local-id address**

This command, which by default is executed on start-up, makes the local-id of an IKE policy equal to the IPv4 address of the interface on which an IKE negotiation is occurring. This is particularly useful for products that could have multiple public interfaces.

The second method is to use the IKE policy command:
(config-ike)#**local-id [address | fqdn | user-fqdn]** <*ipaddress or fqdn*>

This policy-specific command allows you to manually set the local-id for an IKE policy on a per-policy basis. You can use both methods simultaneously in the product. Several IKE policies can be created, some of which use the default system setting of the IPv4 address of the public interface. Others can be set to override this system setting and manually configure a local-id specific to those policies. When a new IKE policy is created, they default to **no local-id**. This allows the system local-id setting to be applied to the policy.

Usage Examples

The following example sets the local ID of this IKE policy to the IPv4 address 192.168.1.1:

(config-ike)# **local-id address 192.168.1.1**

peer [*<ip address>* **| any**]

Use the **peer** command to enter the IP address of the peer device. Repeat this command for multiple peers. Use the **any** keyword if you want to set up a policy that will initiate or respond to any peer.

Syntax Description

<ip address>	Enter a peer IP address.
any	Allow any peer to connect to this IKE policy.

Default Values

There are no default settings for this command.

Command Modes

(config-ike)#	IKE Policy Configuration Mode

Command History

Release 4.1	Command was introduced

Functional Notes

An IKE policy is incomplete unless one of the peer commands is specified. Only one IKE policy can be configured with **peer any**.

Usage Examples

The following example sets multiple peers on an IKE policy for an initiate and respond policy using pre-shared secret, des, md5, and Diffie-Hellman group 1:

(config)#**crypto ike policy 100**
(config-ike)#**peer 192.168.1.1**
(config-ike)#**peer 192.168.1.2**
(config-ike)#**peer 192.168.1.3**
(config-ike)#**respond anymode**
(config-ike)#**initiate main**

The following example sets up a policy allowing any peer to initiate using pre-shared secret, des, md5, and Diffie-Hellman group 1.

(config)#**crypto ike policy 100**
(config-ike)#**peer any**
(config-ike)#**respond anymode**
(config-ike)#**initiate main**

Technology Review

IKE policies must have a peer address associated with them to allow certain peers to negotiate with the ADTRAN product. This is a problem when you have "roaming" users (those who obtain their IP address using DHCP or some other dynamic means). To allow for "roaming" users, the IKE policy can be set up with **peer any** to allow any peer to negotiate with the ADTRAN product. There can only be one **peer any** policy in the running configuration.

respond [main | aggressive | anymode]

Use the **respond** command to allow the IKE policy to respond to negotiations by a peer. Use the **no** form of this command to allow the policy to only initiate negotiations.

Syntax Description

main	Specify to respond to only main mode.
aggressive	Specify to respond to only aggressive mode.
anymode	Specify to respond to any mode.

Default Values

By default, respond to any mode is enabled.

Command Modes

(config-ike)#	IKE Policy Configuration Mode

Command History

Release 4.1	Command was introduced

Functional Notes

By using the **initiate** and **respond** commands, you can configure the IKE policy to initiate and respond, initiate only, or respond only. It is an error if you have both **initiate** and **respond** disabled.

Usage Examples

The following example configures the router to initiate and respond to IKE negotiations:

(config-ike)# **respond anymode**
(config-ike)# **initiate main**

IKE CLIENT COMMAND SET

To activate the IKE Client command set, enter the **crypto ike client** command at the Global Configuration Mode prompt. For example:

```
Router> enable
Router# configure terminal
Router(config)# crypto ike client configuration pool ConfigPool1
Router(config-ike-client-pool)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

do on page 558

end on page 559

exit on page 560

All other commands for this command set are described in this section in alphabetical order.

dns-server <address1> <address2> on page 267

ip-range <start ip> <end ip> on page 268

netbios-name-server <address1> <address2> on page 269

*For VPN configuration example scripts, refer to the technical support note **Configuring VPN** located on the **NetVanta 3000 Series System Manual** CD provided with your unit.*

dns-server *<address1> <address2>*

Use the **dns-server** command to specify the DNS server address(es) to assign to a client.

Syntax Description

<address1>	The first DNS server address to assign.
<address2> **Optional*	The second DNS server address to assign.

Default Values

By default, no DNS server address is defined.

Command Modes

(config-ike-client-pool)#	IKE Client Configuration Mode

Command History

Release 4.1	Command was introduced

Usage Examples

The following example defines two DNS server addresses for this configuration pool:

(config-ike-client-pool)# **dns-server 172.1.17.1 172.1.17.3**

ip-range *<start ip> <end ip>*

Use the **ip-range** command to specify the range of addresses from which the router draws when assigning an IP address to a client.

Syntax Description

<start ip>	The first IP address in the range for this pool.
<end ip>	The last IP address in the range for this pool.

Default Values

By default, no IP address range is defined.

Command Modes

(config-ike-client-pool)#	IKE Client Configuration Mode

Command History

Release 4.1	Command was introduced

Usage Examples

The following example defines an IP address range for this configuration pool:

(config-ike-client-pool)# **ip-range 172.1.1.1 172.1.1.25**

netbios-name-server *<address1> <address2>*

Use the **netbios-name-server** command to specify the NetBIOS Windows Internet Naming Service (WINS) name servers to assign to a client.

Syntax Description

<address1>	The first WINs server address to assign.
<address2>	The second WINs server address to assign.

Default Values

By default, no WINs server address is defined.

Command Modes

(config-ike-client-pool)#	IKE Client Configuration Mode

Command History

Release 4.1	Command was introduced

Usage Examples

The following example defines two WINs server addresses for this configuration pool:

(config-ike-client-pool)# **netbios-name-server 172.1.17.1 172.1.17.25**

IKE POLICY ATTRIBUTES COMMAND SET

To activate the IKE Policy Attributes command set, enter the **attribute** command at the IKE Policy prompt. For example:

```
Router> enable
Router# configure terminal
Router(config)# crypto ike policy 1
Router(config-ike)# attribute 10
Router(config-ike-attribute)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

do on page 558

end on page 559

exit on page 560

All other commands for this command set are described in this section in alphabetical order.

authentication pre-share on page 271

encryption [aes-xxx-cbc | des | 3des] on page 272

group [1 | 2] on page 273

hash [md5| sha] on page 274

lifetime <seconds> on page 275

*For VPN configuration example scripts, refer to the technical support note **Configuring VPN** located on the **NetVanta 3000 Series System Manual** CD provided with your unit.*

authentication pre-share

Use the **authentication pre-share** command to configure this IKE policy to use pre-shared secrets during IKE negotiation. Other authentication choices will be available in future ADTRAN OS updates.

Syntax Description

pre-share	Specify the use of pre-shared secrets during IKE negotiation to validate the peer.

Default Values

By default, this command is enabled.

Command Modes

(config-ike-attribute)#	IKE Policy Attribute Configuration Mode

Command History

Release 4.1	Command was introduced

Functional Notes

Both sides must share the same pre-shared secret in order for the negotiation to be successful.

Usage Examples

The following example enables pre-shared secrets for this IKE policy:

(config-ike-attribute)# **authentication pre-share**

encryption [aes-xxx-cbc | des | 3des]

Use the **encryption** command to specify which encryption algorithm this IKE policy will use to transmit data over the IKE-generated SA.

Syntax Description

aes-128-cbc	Choose the aes-128-cbc encryption algorithm.
aes-192-cbc	Choose the aes-192-cbc encryption algorithm.
aes-256-cbc	Choose the aes-256-cbc encryption algorithm.
des	Choose the des encryption algorithm.
3des	Choose the 3des encryption algorithm.

Default Values

By default, encryption is set to des.

Command Modes

(config-ike-attribute)#	IKE Policy Attribute Configuration Mode

Command History

Release 4.1	Command was introduced

Usage Examples

The following example selects 3des as the encryption algorithm for this IKE policy:

(config-ike-attribute)# **encryption 3des**

group [1 | 2]

Use the **group** command to specify the Diffie-Hellman group (1 or 2) to be used by this IKE policy to generate the keys (which are then used to create the IPSec SA).

Syntax Description

1	768-bit mod P
2	1024-bit mod P

Default Values

By default, group is set to 1.

Command Modes

(config-ike-attribute)#	IKE Policy Attribute Configuration Mode

Command History

Release 4.1	Command was introduced

Functional Notes

The local IKE policy and the peer IKE policy must have matching group settings in order for negotiation to be successful.

Usage Examples

The following example sets this IKE policy to use Diffie-Hellman group 2:

(config-ike-attribute)# **group 2**

hash [md5| sha]

Use the **hash** command to specify the hash algorithm to be used to authenticate the data transmitted over the IKE SA.

Syntax Description

md5	Choose the md5 hash algorithm.
sha	Choose the sha hash algorithm.

Default Values

By default, hash is set to sha.

Command Modes

(config-ike-attribute)#	IKE Policy Attribute Configuration Mode

Command History

Release 4.1	Command was introduced

Usage Examples

The following example specifies **md5** as the hash algorithm:

(config-ike-attribute)# **hash md5**

lifetime *<seconds>*

Use the **lifetime** command to specify how long an IKE SA is valid before expiring.

Syntax Description

<seconds>	Specify how many seconds an IKE SA will last before expiring.

Default Values

By default, lifetime is set to 28,800 seconds.

Command Modes

(config-ike-attribute)#	IKE Policy Attribute Configuration Mode

Command History

Release 4.1	Command was introduced

Usage Examples

The following example sets a lifetime of two hours:

(config-ike-attribute)# **lifetime 7200**

CRYPTO MAP IKE COMMAND SET

To activate the Crypto Map IKE command set, enter a valid version of the **crypto map ipsec-ike** command at the Global Configuration Mode prompt. For example:

```
Router> enable
Router# configure terminal
Router(config)# crypto map Map-Name 10 ipsec-ike
Router(config-crypto-map)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

do on page 558

end on page 559

exit on page 560

All other commands for this command set are described in this section in alphabetical order.

match address <*listname*> on page 277

set peer <*address*> on page 279

set pfs [group1 | *group2]* on page 280

set security-association lifetime [kilobytes | *seconds]* <*value*> on page 281

set transform-set <*setname1* - *setname6*> on page 282

For VPN configuration example scripts, refer to the technical support note ***Configuring VPN*** *located on the* ***NetVanta 3000 Series System Manual*** *CD provided with your unit.*

match address *<listname>*

Use the **match address** command to assign an IP access-list to a crypto map definition. The access-list designates the IP packets to be encrypted by this crypto map. See *ip access-list extended <listname>* on page 170 for more information on creating access-lists.

Syntax Description

<listname>	Enter the name of the access-list you wish to assign to this crypto map.

Default Values

By default, no IP access-lists are defined.

Command Modes

(config-crypto-map)#	Crypto Map Configuration Mode (IKE or Manual)

Command History

Release 4.1	Command was introduced

Functional Notes

Crypto map entries do not directly contain the selectors used to determine which data to secure. Instead, the crypto map entry refers to an access control list. An access control list (ACL) is assigned to the crypto map using the **match address** command (see *crypto map* on page 157). If no ACL is configured for a crypto map, then the entry is incomplete and will have no effect on the system.

The entries of the ACL used in a crypto map should be created with respect to traffic sent by the ADTRAN product. The source information must be the local ADTRAN product and the destination must be the peer.

Only extended access-lists can be used in crypto maps.

Usage Examples

The following example shows setting up an ACL (called **NewList**) and then assigning the new list to a crypto map (called **NewMap**):

(config)#**ip access-list extended NewList**

Configuring New Extended ACL "NewList"

(config-ext-nacl)#**exit**
(config)#**crypto map NewMap 10 ipsec-ike**
(config-crypto-map)#**match address NewList**

Technology Review

A crypto map entry is a single policy that describes how certain traffic is to be secured. There are two types of crypto map entries: ipsec-manual and ipsec-ike. Each entry is given an index, which is used to sort the ordered list.

When a non-secured packet arrives on an interface, the crypto map set associated with that interface is processed in order. If a crypto map entry matches the non-secured traffic, the traffic is discarded.

When a packet is to be transmitted on an interface, the crypto map set associated with that interface is processed in order. The first crypto map entry that matches the packet will be used to secure the packet. If a suitable SA exists, that is used for transmission. Otherwise, IKE is used to establish an SA with the peer. If no SA exists, and the crypto map entry is “respond only”, the packet is discarded.

When a secured packet arrives on an interface, its SPI is used to look up an SA. If an SA does not exist, or if the packet fails any of the security checks (bad authentication, traffic does not match SA selectors, etc.), it is discarded. If all checks pass, the packet is forwarded normally.

set peer *<address>*

Use the **set peer** command to set the IP address of the peer device. This must be set for multiple remote peers.

Syntax Description

<address>	Enter the IP address of the peer device. If this is not configured, it implies responder only to any peer.

Default Values

There are no default settings for this command.

Command Modes

(config-crypto-map)#	Crypto Map Configuration Mode (IKE or Manual)

Command History

Release 4.1	Command was introduced

Functional Notes

If no peer IP addresses are configured, the entry will only be used to respond to IPSec requests; it cannot initiate the requests (since it doesn't know which IP address to send the packet to). If a single peer IP address is configured, the crypto map entry can be used to both initiate and respond to SAs.

The peer IP address is the public IP address of the device which will terminate the IPSec tunnel. If the peer IP address is not static, the ADTRAN product cannot initiate the VPN tunnel. By setting no peer IP address, the ADTRAN product can respond to an IPSec tunnel request in this case.

Usage Examples

The following example sets the peer IP address of 10.100.23.64:

(config-crypto-map)# **set peer 10.100.23.64**

set pfs [group1 | group2]

Use the **set pfs** command to choose the type of perfect forward secrecy (if any) that will be required during IPSec negotiation of security associations for this crypto map. Use the **no** form of this command to require no PFS.

Syntax Description

group1	IPSec is required to use Diffie-Hellman Group 1 (768-bit modulus) exchange during IPSec SA key generation.
group2	IPSec is required to use Diffie-Hellman Group 2 (1024-bit modulus) exchange during IPSec SA key generation.

Default Values

By default, no PFS will be used during IPSec SA key generation.

Command Modes

(config-crypto-map)#	Crypto Map IKE Configuration Mode

Command History

Release 4.1	Command was introduced

Functional Notes

If left at the default setting, no perfect forward secrecy (PFS) will be used during IPSec SA key generation. If PFS is specified, then the specified Diffie-Hellman Group exchange will be used for the initial and all subsequent key generation, thus providing no data linkage between prior keys and future keys.

Usage Examples

The following example specifies use of the Diffie-Hellman Group 1 exchange during IPSec SA key generation:

(config-crypto-map)# **set pfs group 1**

set security-association lifetime [kilobytes | seconds] <*value*>

Use the **set security-association lifetime** command to define the lifetime (in kilobytes and/or seconds) of the IPSec SAs created by this crypto map.

Syntax Description

kilobytes <*value*>	SA lifetime limit in kilobytes.
seconds <*value*>	SA lifetime limit in seconds.

Default Values

By default, security-association lifetime is set to 28,800 seconds and there is no default for the kilobytes lifetime.

Command Modes

(config-crypto-map)#	Crypto Map IKE Configuration Mode

Command History

Release 4.1	Command was introduced

Functional Notes

Values can be entered for this command in both kilobytes and seconds. Whichever limit is reached first will end the security association.

Usage Examples

The following example sets the SA lifetime to 300 kilobytes and 2 hours:

(config-crypto-map)# **set security-association lifetime kilobytes 300**
(config-crypto-map)# **set security-association lifetime seconds 7200**

set transform-set *<setname1 - setname6>*

Use the **set transform-set** command to assign up to six transform-sets to a crypto map. See *crypto ipsec transform-set <setname> <parameters>* on page 155 for information on defining transform-sets.

Syntax Description

<setname>	Assign up to six transform-sets to this crypto map by listing the set names, separated by a space.

Default Values

By default, there is no transform-set assigned to the crypto map.

Command Modes

(config-crypto-map)#	Crypto Map Configuration Mode (IKE or Manual)

Command History

Release 4.1	Command was introduced

Functional Notes

Crypto map entries do not directly contain the transform configuration for securing data. Instead, the crypto map is associated with transform sets which contain specific security algorithms (see *crypto ipsec transform-set <setname> <parameters>* on page 155).

If no transform-set is configured for a crypto map, then the entry is incomplete and will have no effect on the system.

Usage Examples

The following example first creates a transform-set (**Set1**) consisting of two security algorithms (up to three may be defined), and then assigns the transform-set to a crypto map (**Map1**):

```
(config)#crypto ipsec transform-set Set1 esp-3des esp-sha-hmac
(cfg-crypto-trans)#exit

(config)#crypto map Map1 1 ipsec-ike
(config-crypto-map)#set transform-set Set1
```


CRYPTO MAP MANUAL COMMAND SET

To activate the Crypto Map Manual command set, enter a valid version of the **crypto map ipsec-manual** command at the Global Configuration Mode prompt. For example:

```
Router> enable
Router# configure terminal
Router(config)# crypto map Map-Name 10 ipsec-manual
Router(config-crypto-map)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

do on page 558

end on page 559

exit on page 560

All other commands for this command set are described in this section in alphabetical order.

match address <*listname*> on page 284

set peer <*address*> on page 286

set session-key [inbound | outbound] on page 287

set transform-set <*setname*> on page 290

For VPN configuration example scripts, refer to the technical support note ***Configuring VPN*** *located on the* ***NetVanta 3000 Series System Manual*** *CD provided with your unit.*

match address *<listname>*

Use the **match address** command to assign an IP access-list to a crypto map definition. The access-list designates the IP packets to be encrypted by this crypto map. See *ip access-list extended <listname>* on page 170 for more information on creating access-lists.

Syntax Description

<listname>	Enter the name of the access-list you wish to assign to this crypto map.

Default Values

By default, no IP access-lists are defined.

Command Modes

(config-crypto-map)#	Crypto Map Configuration Mode (IKE or Manual)

Command History

Release 4.1	Command was introduced

Functional Notes

Crypto map entries do not directly contain the selectors used to determine which data to secure. Instead, the crypto map entry refers to an access control list. An access control list (ACL) is assigned to the crypto map using the **match address** command (see *crypto map* on page 157). If no ACL is configured for a crypto map, then the entry is incomplete and will have no effect on the system.

The entries of the ACL used in a crypto map should be created with respect to traffic sent by the ADTRAN product. The source information must be the local ADTRAN product, and the destination must be the peer.

Only extended access-lists can be used in crypto maps.

Usage Examples

The following example shows setting up an access-list (called **NewList**) and then assigning the new list to a crypto map (called **NewMap**):

(config)#**ip access-list extended NewList**

Configuring New Extended ACL "NewList"

(config-ext-nacl)#**exit**

(config)#**crypto map NewMap 10 ipsec-manual**
(config-crypto-map)#**match address NewList**

Technology Review

A crypto map entry is a single policy that describes how certain traffic is to be secured. There are two types of crypto map entries: ipsec-manual and ipsec-ike. Each entry is given an index, which is used to sort the ordered list.

When a non-secured packet arrives on an interface, the crypto map set associated with that interface is processed in order. If a crypto map entry matches the non-secured traffic, the traffic is discarded.

When a packet is to be transmitted on an interface, the crypto map set associated with that interface is processed in order. The first crypto map entry that matches the packet will be used to secure the packet. If a suitable SA exists, that is used for transmission. Otherwise, IKE is used to establish an SA with the peer. If no SA exists, and the crypto map entry is “respond only”, the packet is discarded.

When a secured packet arrives on an interface, its SPI is used to look up an SA. If an SA does not exist, or if the packet fails any of the security checks (bad authentication, traffic does not match SA selectors, etc.), it is discarded. If all checks pass, the packet is forwarded normally.

set peer *<address>*

Use the **set peer** command to set the IP address of the peer device.

Syntax Description

<address>	Enter the IP address of the peer device.

Default Values

There are no default settings for this command.

Command Modes

(config-crypto-map)#	Crypto Map Configuration Mode (IKE or Manual)

Command History

Release 4.1	Command was introduced

Functional Notes

If no peer IP address is configured, the manual crypto map is not valid and not complete. A peer IP address is required for manual crypto maps. To change the peer IP address, the **no set peer** command must be issued first; then the new peer IP address can be configured.

Usage Examples

The following example sets the peer IP address of 10.100.23.64:

(config-crypto-map)# **set peer 10.100.23.64**

set session-key [inbound | outbound]

Use the **set session-key** command to define the encryption and authentication keys for this crypto map.

Variations of this command include the following:
set session-key inbound ah <*SPI*> <*keyvalue*>
set session-key inbound esp <*SPI*> authenticator <*keyvalue*>
set session-key inbound esp <*SPI*> cipher <*keyvalue*>
set session-key inbound esp <*SPI*> cipher <*keyvalue*> authenticator <*keyvalue*>
set session-key outbound ah <*SPI*> <*keyvalue*>
set session-key outbound esp <*SPI*> authenticator <*keyvalue*>
set session-key outbound esp <*SPI*> cipher <*keyvalue*>
set session-key outbound esp <*SPI*> cipher <*keyvalue*> authenticator <*keyvalue*>

Syntax Description

inbound	Use this keyword to define encryption keys for inbound traffic.
outbound	Use this keyword to define encryption keys for outbound traffic.
ah <*SPI*>	Authentication header protocol.
esp <*SPI*>	Encapsulating security payload protocol.
cipher <*keyvalue*>	Specify encryption/decryption key.
authenticator <*keyvalue*>	Specify authentication key.

Default Values

There are no default settings for this command.

Command Modes

(config-crypto-map)#	Crypto Map Manual Configuration Mode

Command History

Release 4.1	Command was introduced

Functional Notes

The inbound local SPI (security parameter index) must equal the outbound remote SPI. The outbound local SPI must equal the inbound remote SPI. The key values are the hexadecimal representations of the keys. They are not true ASCII strings. Therefore, a key of 3031323334353637 represents “01234567”.

See the following table for key length requirements.

Algorithm	Minimum key length required
des	64-bits in length; 8 hexadecimal bytes
3des	192-bits in length; 24 hexadecimal bytes

Functional Notes (Continued)

AES-128-CBC	128-bits in length; 16 hexadecimal bytes
AES-192-CBC	192-bits in length; 24 hexadecimal bytes
AES-256-CBC	256-bits in length; 32 hexadecimal bytes
md5	128-bits in length; 16 hexadecimal bytes
sha1	160-bits in length; 20 hexadecimal bytes

Technology Review

The following example configures an ADTRAN OS product for VPN using IPSec manual keys. This example assumes that the ADTRAN OS product has been configured with a WAN IP Address of 192.168.1.1 on interface **ethernet 0/1** and a LAN IP Address of 10.10.10.254 on interface **ethernet 0/2**. The Peer Private IP Subnet is 10.10.20.0.

For more detailed information on VPN configuration, refer to the technical support note *Configuring VPN* located on the ***NetVanta 3000 Series System Manual*** CD provided with your unit.

Step 1:
Enter the Global configuration mode (i.e., config terminal mode).

```
>enable
#configure terminal
```

Step 2:
Enable VPN support using the **ip crypto** command. This command allows crypto maps to be applied to interfaces, and enables the IKE server to listen for IKE negotiation sessions on UDP port 500.

```
(config)#ip crypto
```

Step 3:
Define the transform-set. A transform-set defines the encryption and/or authentication algorithms to be used to secure the data transmitted over the VPN tunnel. Multiple transform-sets may be defined in a system. Once a transform-set is defined, many different crypto maps within the system can reference it. In this example, a transform-set named **highly_secure** has been created. This transform-set defines ESP with Authentication implemented using 3DES encryption and SHA1 authentication.

```
(config)#crypto ipsec transform-set highly_secure esp-3des esp-sha-hmac
(cfg-crypto-trans)#mode tunnel
```

Step 4:
Define an ip-access list. An Extended Access Control List is used to specify which traffic needs to be sent securely over the VPN tunnel. The entries in the list are defined with respect to the local system. The source IP address will be the source of the traffic to be encrypted. The destination IP address will be the receiver of the data on the other side of the VPN tunnel.

```
(config)#ip access-list extended corporate_traffic
(config-ext-nacl)#permit ip 10.10.10.0 0.0.0.255 10.10.20.0 0.0.0.255 log
  deny ip any any
```


Technology Review (Continued)

Step 5:
Create crypto map and define manual keys. A Crypto Map is used to define a set of encryption schemes to be used for a given interface. A crypto map entry has a unique index within the crypto map set. The crypto map entry will specify whether IKE is used to generate encryption keys or if manually specified keys will be used. The crypto map entry will also specify who will be terminating the VPN tunnel, as well as which transform-set or sets will be used to encrypt and/or authenticate the traffic on that VPN tunnel. It also specifies the lifetime of all created IPSec Security Associations.

The keys for the algorithms defined in the transform-set associated with the crypto map will be defined by using the **set session-key** command. A separate key is needed for both inbound and outbound traffic. The key format consists of a string of hexadecimal values without the leading **0x** for each character. For example, a cipher key of **this is my cipher key** would be entered as:
74686973206973206D79206369706865722206B6579.

A unique Security Parameter Index (SPI) is needed for both inbound and outbound traffic. The local system's inbound SPI and keys will be the peer's outbound SPI and keys. The local system's outbound SPI and keys will be the peer's inbound SPI and keys. In this example the following keys and SPIs are used:

- Inbound cipher SPI: 300Inbound cipher key: "2te$#g89jnr(j!@4rvnfhg5e"
- Outbound cipher SPI: 400Outbound cipher key: "8564hgjelrign*&(gnb#1$d3"
- Inbound authenticator key:"r5%^ughembkdhj34$x.<"
- Outbound authenticator key:"io78*7gner#4(mgnsd!3"

```
(config)#crypto map corporate_vpn 1 ipsec-ike
(config-crypto-map)#match address corporate_traffic
(config-crypto-map)#set peer 192.168.1.2
(config-crypto-map)#set transform-set highly_secure
(config-crypto-map)#set session-key inbound esp 300 cipher
32746524236738396A6E72286A21403472766E6668673565 authenticator
7235255E756768656D626B64686A333424782E3C
(config-crypto-map)#set session-key outbound esp 400 cipher
3835363468676A656C7269676E2A2628676E622331246433 authenticator
696F37382A37676E65722334286D676E73642133
```

Step 6:
Configure public interface. This process includes configuring the IP address for the interface and applying the appropriate crypto map to the interface. Crypto maps are applied to the interface on which encrypted traffic will be transmitted.

```
(config)#interface ethernet 0/1
(config-eth 0/1)#ip address 192.168.1.1 255.255.255.0
(config-eth 0/1)#crypto map corporate_vpn
(config-eth 0/1)#no shutdown
```

Step 7:
Configure private interface and add a static route to allow all traffic destined for the VPN tunnel to be routed to the appropriate gateway.

```
(config)#interface ethernet 0/2
(config-eth 0/2)#ip address 10.10.10.254 255.255.255.0
(config-eth 0/2)#no shutdown
(config-eth 0/2)#exit
(config)#ip route 10.10.20.0 255.255.255.0 192.168.1.2
```

set transform-set <*setname*>

Use the **set transform-set** command to assign a transform-set to a crypto map. See *crypto ipsec transform-set <setname> <parameters>* on page 155 for information on defining transform-sets.

Syntax Description

<*setname*>	Assign a transform-set to this crypto map by entering the set name.

Default Values

By default, no transform-set is assigned to the crypto map.

Command Modes

(config-crypto-map)#	Crypto Map Configuration Mode (IKE or Manual)

Command History

Release 4.1	Command was introduced

Functional Notes

Crypto map entries do not directly contain the transform configuration for securing data. Instead, the crypto map is associated with transform sets which contain specific security algorithms (see *crypto ipsec transform-set <setname> <parameters>* on page 155).

If no transform-set is configured for a crypto map, then the entry is incomplete and will have no effect on the system. For manual key crypto maps, only one transform set can be specified.

Usage Examples

The following example first creates a transform-set (**Set1**) consisting of two security algorithms (up to three may be defined), and then assigns the transform-set to a crypto map (**Map1**):

```
(config)#crypto ipsec transform-set Set1 esp-3des esp-sha-hmac
(cfg-crypto-trans)#exit

(config)#crypto map Map1 1 ipsec-manual
(config-crypto-map)#set transform-set Set1
```

ETHERNET INTERFACE CONFIGURATION COMMAND SET

To activate the Ethernet Interface Configuration command set, enter the **interface ethernet** command at the Global Configuration Mode prompt. For example:

```
Router> enable
Router# configure terminal
Router(config)# interface ethernet 0/1
Router(config-eth 0/1)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

All other commands for this command set are described in this section in alphabetical order.

access-policy *<policyname>*

Use the **access-policy** command to assign a specified access policy to an interface. Use the **no** form of this command to remove an access policy association.

Syntax Description

<policyname>	Alphanumeric descriptor for identifying the configured access policy (all access policy descriptors are case-sensitive)

Default Values

By default, there are no configured access policies associated with an interface.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), and virtual frame relay sub-interfaces (fr 1.20)

Command History

Release 2.1	Command was introduced

Functional Notes

To assign an access policy to an interface, enter the interface configuration mode for the desired interface and enter **access policy** <*policy name*>.

Usage Examples

The following example associates the access policy **UnTrusted** (to allow inbound traffic to the Web server) to the ethernet 0/1 interface:

Enable the ADTRAN OS security features:
(config)# **ip firewall**

Create the access list (this is the packet selector):
(config)# **ip access-list extended InWeb**
(config-ext-nacl)# **permit tcp any host 63.12.5.253 eq 80**

Create the access policy that contains the access list **InWeb**:
(config)# **ip policy-class UnTrusted**
(config-policy-class)# **allow list InWeb**

Usage Examples (Continued)

Associate the access policy with the ethernet 0/1 interface:
(config)# **interface ethernet 0/1**
(config-eth 0/1) **access-policy UnTrusted**

Technology Review

Creating access policies and lists to regulate traffic through the routed network is a four-step process:

Step 1:

Enable the security features of the ADTRAN OS using the **ip firewall** command.

Step 2:

Create an access list to permit or deny specified traffic. Standard access lists provide pattern matching for source IP addresses only. (Use extended access lists for more flexible pattern matching.) IP addresses can be expressed in one of three ways:

1. Using the keyword **any** to match any IP address. For example, entering **deny any** will effectively shut down the interface that uses the access list because all traffic will match the **any** keyword.
2. Using the **host** <A.B.C.D> to specify a single host address. For example, entering **permit host 196.173.22.253** will allow all traffic from the host with an IP address of 196.173.22.253.
3. Using the <A.B.C.D> <wildcard> format to match all IP addresses in a “range”. Wildcard masks work in reverse logic from subnet mask. Specifying a one in the wildcard mask equates to a “don’t care”. For example, entering **deny 192.168.0.0 0.0.0.255** will deny all traffic from the 192.168.0.0/24 network.

Step 3:

Create an IP policy class that uses a configured access list. ADTRAN OS access policies are used to permit, deny, or manipulate (using NAT) data for each physical interface. Each ACP consists of a selector (access list) and an action (allow, discard, NAT). When packets are received on an interface, the configured ACPs are applied to determine whether the data will be processed or discarded. Possible actions performed by the access policy are as follows:

allow list <*access list names*>
All packets passed by the access list(s) entered will be allowed to enter the router system.

discard list <*access list names*>
All packets passed by the access list(s) entered will be dropped from the router system.

allow list <*access list names*> **dest-policy** <*access policy name*>
All packets passed by the access list(s) entered and destined for the interface using the access policy listed will be permitted to enter the router system. This allows for configurations to permit packets to a single interface and not the entire system.

allow list <*access list names*> **dest-policy** <*access policy name*>
All packets passed by the access list(s) entered and destined for the interface using the access policy listed will be blocked from the router system. This allows for configurations to deny packets on a specified interface.

Technology Review (Continued)

nat source list <*access list names*> **address** <*IP address*> **overload**
All packets passed by the access list(s) entered will be modified to replace the source IP address with the entered IP address. The **overload** keyword allows multiple source IP addresses to be replaced with the single IP address entered. This hides private IP addresses from outside the local network.

nat source list <*access list names*> **interface** <*interface*> **overload**
All packets passed by the access list(s) entered will be modified to replace the source IP address with the primary IP address of the listed interface. The **overload** keyword allows multiple source IP addresses to be replaced with the single IP address of the specified interface. This hides private IP addresses from outside the local network.

nat destination list <*access list names*> **address** <*IP address*>
All packets passed by the access list(s) entered will be modified to replace the destination IP address with the entered IP address. The **overload** keyword is not an option when performing NAT on the destination IP address; each private address must have a unique public address. This hides private IP addresses from outside the local network.

Step 4:

Apply the created access policy to an interface. To assign an access policy to an interface, enter the interface configuration mode for the desired interface and enter **access policy** <*policy name*>. The following example assigns access policy **MatchAll** to the ethernet 0/1 interface:

(config)# **interface ethernet 0/1**

(config-eth 0/1)# **access-policy MatchAll**

arp arpa

Use the **arp arpa** command to enable address resolution protocol on the Ethernet interface.

Syntax Description

arpa	Keyword used to set standard address resolution protocol for this interface.

Default Values

The default for this command is arpa.

Command Modes

(config-eth 0/1)#	Ethernet Interface Configuration Mode

Command History

Release 3.1	Command was introduced

Usage Examples

The following example enables standard ARP for the Ethernet interface:

(config)# **interface eth 0/1**
(config-eth 0/1)# **arp arpa**

bandwidth <*value*>

Use the **bandwidth** command to provide the bandwidth value of an interface to the higher-level protocols. This value is used in cost calculations. Use the **no** form of this command to restore the default values.

Syntax Description

<*value*>	Enter bandwidth in kbps.

Default Values

*To view default values use the **show interfaces** command.*

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), Frame Relay Virtual Sub-interfaces (fr 1.20), virtual PPP (ppp 1), and loopback interfaces

Command History

Release 3.1	Command was introduced

Functional Notes

The **bandwidth** command is an informational value that is communicated to the higher-level protocols to be used in cost calculations. This is a routing parameter only and does not affect the physical interface.

Usage Examples

The following example sets bandwidth of the ethernet 0/1 interface to 10 Mbps:

(config)# **interface eth 0/1**
(config-eth 0/1)# **bandwidth 10000**

bridge-group *<group#>*

Use the **bridge-group** command to assign an interface to the specified bridge group. This command is supported on all Ethernet interfaces, PPP virtual interfaces, and frame relay virtual sub-interfaces. Use the **no** form of this command to remove the interface from the bridge group.

Syntax Description

<group#>	Bridge group number (1 to 255) specified using the **bridge-group** command

Default Values

By default, there are no configured bridge groups.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), and virtual frame relay sub-interfaces (fr 1.20)

Command History

Release 1.1	Command was introduced

Functional Notes

A bridged network can provide excellent traffic management to reduce collisions and limit the amount of bandwidth wasted with unnecessary transmissions when routing is not necessary. Any two interfaces can be bridged (Ethernet to T1 bridge, Ethernet to frame relay sub-interface).

Usage Examples

The following example assigns the Ethernet interface to bridge-group 17:

(config)# **interface eth 0/1**
(config-eth 0/1)# **bridge-group 17**

bridge-group <*group#*> **path-cost** <*value*>

Use the **bridge-group path-cost** command to assign a cost to a bridge group that is used when computing the spanning-tree root path. To return to the default path-cost value, use the **no** form of this command.

Syntax Description

<*group#*>	Bridge group number (1 to 255) specified using the **bridge-group** command
<*value*>	Number assigned to the bridge interface to be used as the path cost in spanning calculations (valid range: 0 to 65535)

Default Values

<*value*>	*1000/(link speed in Mbps)*

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), and virtual frame relay sub-interfaces (fr 1.20)

Command History

Release 1.1	Command was introduced

Functional Notes

The specified value is inversely proportional to the likelihood the bridge interface will be chosen as the root path. Set the path-cost value lower to increase the chance the interface will be the root. To obtain the most accurate spanning-tree calculations, develop a system for determining path costs for links and apply it to all bridged interfaces.

Usage Examples

The following example assigns a path cost of 100 for bridge group 17 on ethernet 0/1:

(config)# **interface eth 0/1**
(config-eth 0/1)# **bridge-group 17 path-cost 100**

Technology Review

Spanning-tree protocol provides a way to prevent loopback or parallel paths in bridged networks. Using the priority values and path costs assigned to each bridging interface, the spanning-tree protocol determines the root path and identifies whether to block or allow other paths.

bridge-group <*group#*> **priority** <*value*>

Use the **bridge-group priority** command to select the priority level of a port associated with a bridge. To return to the default bridge-group priority value, use the **no** version of this command.

Syntax Description

<*group#*>	Bridge group number (1 to 255) specified using the **bridge-group** command
<*value*>	Priority value for the bridge group; the lower the value, the higher the priority (valid range: 0 to 255)

Default Values

<*value*>	128

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), and virtual frame relay sub-interfaces (fr 1.20)

Command History

Release 1.1	Command was introduced

Functional Notes

The only time that this priority level is used is when two interfaces with a path to the root have equal cost. At that point, the level set in this command will determine which port the bridge will use. Set the priority value lower to increase the chance the interface will be used.

Usage Examples

The following example sets the maximum priority on the Ethernet interface in bridge group 17:

(config)# **interface eth 0/1**
(config-eth 0/1)# **bridge-group 17 priority 0**

bridge-group <*group#*> spanning-disabled

Use the **bridge-group spanning-disabled** command to transparently bridge two interfaces on a network (that have no parallel paths) without the overhead of spanning-tree protocol calculations. To enable the spanning-tree protocol on an interface, use the **no** form of this command.

Syntax Description

<*group#*>	Bridge group number (1 to 255) specified using the **bridge-group** command

Default Values

By default, spanning-tree protocol is enabled on all created bridge groups.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), and virtual frame relay sub-interfaces (fr 1.20)

Command History

Release 1.1	Command was introduced

Functional Notes

When no parallel (redundant) paths exist within a bridged network, disabling the spanning tree protocol reduces traffic on the bridged interface. This traffic reduction can be helpful when bridging over a WAN link.

Before disabling the spanning-tree protocol on a bridged interface, verify that no redundant loops exist.

Usage Examples

The following example disables the spanning-tree protocol for bridge group 17 on ethernet 0/1:

(config)# **interface eth 0/1**
(config-eth 0/1)# **bridge-group 17 spanning-disabled**

Technology Review

Spanning-tree protocol provides a way to prevent loopback or parallel paths in bridged networks. Using the priority values and path costs assigned to each bridging interface, the spanning-tree protocol determines the root path and identifies whether to block or allow other paths.

crypto map *<mapname>*

Use the **crypto map** command to associate crypto maps with the interface.

When you apply a map to an interface, you are applying all crypto maps with the given map name. This allows you to apply multiple crypto maps if you have created maps which share the same name but have different map index numbers.

For VPN configuration example scripts, refer to the technical support note ***Configuring VPN*** *located on the* ***NetVanta 3000 Series System Manual*** *CD provided with your unit.*

Syntax Description

<mapname>	Enter the crypto map name that you wish to assign to the interface.

Default Values

By default, no crypto maps are assigned to an interface.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), virtual frame relay sub-interfaces (fr 1.20), and loopback interfaces

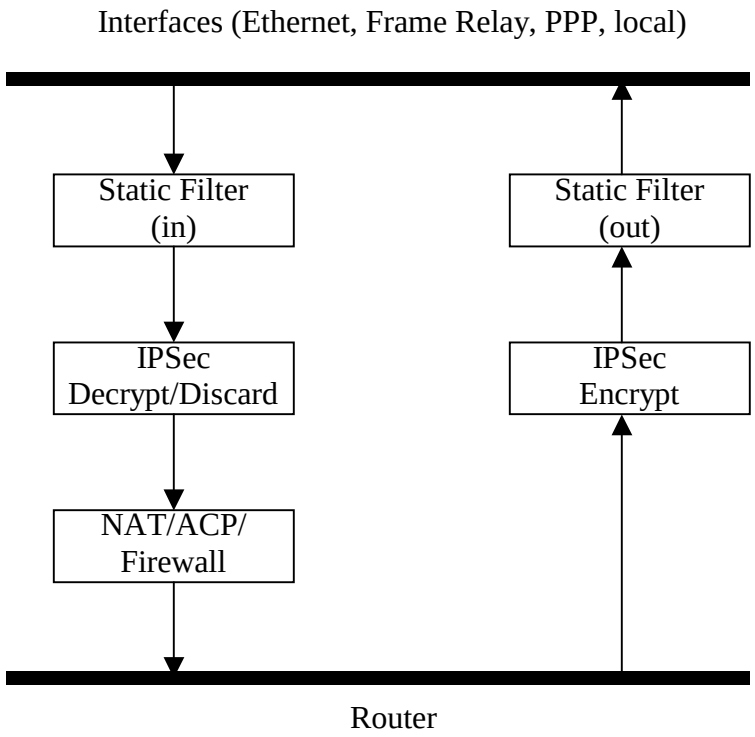
Command History

Release 4.1	Command was introduced

Functional Notes

When configuring a system to use both the stateful inspection firewall and IKE negotiation for VPN, keep the following notes in mind.

When defining the policy-class and associated access-control lists (ACLs) that describe the behavior of the firewall, do not forget to include the traffic coming into the system over a VPN tunnel terminated by the system. The firewall should be set up with respect to the un-encrypted traffic that is destined to be sent or received over the VPN tunnel. The following diagram represents typical AOS data-flow logic.



As shown in the diagram above, data coming into the product is first processed by the static filter associated with the interface on which the data is received. This access-group is a true static filter and is available for use regardless of whether the firewall is enabled or disabled. Next (if the data is encrypted) it is sent to the IPSec engine for decryption. The decrypted data is then processed by the stateful inspection firewall. Therefore, given a terminating VPN tunnel, only un-encrypted data is processed by the firewall.

The ACLs for a crypto map on an interface work in reverse logic to the ACLs for a policy-class on an interface. When specifying the ACLs for a crypto map, the source information is the private local-side, un-encrypted source of the data. The destination information will be the far-end, un-encrypted destination of the data. However, ACLs for a policy-class work in reverse. The source information for the ACL in a policy-class is the far-end. The destination information is the local-side.

Usage Examples

The following example applies all crypto maps with the name **MyMap** to the Ethernet interface:

(config-eth 0/1)# **crypto map MyMap**

full-duplex

Use the **full-duplex** command to configure the Ethernet interface for full-duplex operation. This allows the interface to send and receive simultaneously. Use the **no** form of this to return to the default **half-duplex** operation.

Syntax Description

No subcommands

Default Values

By default, all Ethernet interfaces are configured for half-duplex operation.

Command Modes

(config-eth 0/1)#	Ethernet Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

Full-duplex Ethernet is a variety of Ethernet technology currently being standardized by the IEEE. Because there is no official standard, vendors are free to implement their independent versions of full-duplex operation. Therefore, it is not safe to assume that one vendor's equipment will work with another.
Devices at each end of a full-duplex link have the ability to send and receive data simultaneously over the link. Theoretically, this simultaneous action can provide twice the bandwidth of normal (half-duplex) Ethernet. To deploy full-duplex Ethernet, each end of the link must only connect to a single device (a workstation or a switched hub port). With only two devices on a full-duplex link, there is no need to use the medium access control mechanism (to share the signal channel with multiple stations) and listen for other transmissions or collisions before sending data.

The 10BaseT, 100BaseTX, and 100BaseFX signalling systems support full-duplex operation (because they have transmit and receive signal paths that can be simultaneously active).

*If the **speed** is manually set to **10** or **100,** the duplex must be manually configured as **full-duplex** or **half-duplex**. See speed <rate> on page 325 for more information.*

Usage Examples

The following example configures the Ethernet interface for **full-duplex** operation:

(config)# **interface ethernet 0/1**
(config-eth 0/1)#**full-duplex**

half-duplex

Use the **half-duplex** command to configure the Ethernet interface for half-duplex operation. This setting allows the Ethernet interface to either send or receive at any given moment, but not simultaneously. Use the **no** form of this command to disable half-duplex operation.

Syntax Description

No subcommands

Default Values

By default, all Ethernet interfaces are configured for half-duplex operation.

Command Modes

(config-eth 0/1)#	Ethernet Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

Half-duplex Ethernet is the traditional form of Ethernet that employs the Carrier Sense Multiple Access/Collision Detect (CSMA/CD) protocol to allow two or more hosts to share a common transmission medium while providing mechanisms to avoid collisions. A host on a half-duplex link must "listen" on the link and only transmit when there is an idle period. Packets transmitted on the link are broadcast (so it will be "heard" by all hosts on the network). In the event of a collision (two hosts transmitting at once), a message is sent to inform all hosts of the collision and a backoff algorithm is implemented. The backoff algorithm requires the station to remain silent for a random period of time before attempting another transmission. This sequence is repeated until a successful data transmission occurs.

*If the **speed** is manually set to **10** or **100,** the duplex must be manually configured as **full-duplex** or **half-duplex**. See speed <rate> on page 325 for more information.*

Usage Examples

The following example configures the Ethernet interface for **half-duplex** operation:

```
(config)# interface ethernet 0/1
(config-eth 0/1)# half-duplex
```


ip access-group <*listname*> [**in | out**]

Use the **ip access-group** command to create an access list to be used for packets transmitted to/from the asynchronous host. Use the **no** form of this command to disable this type of control.

Syntax Description

listname	Assigned IP access list name.
in	Enables access control on packets *transmitted from* the host.
out	Enables access control on packets *sent to* the host.

Default Values

By default, these commands are disabled.

Command Modes

(config-interface)#	Interface Configuration Mode required.

Command History

Release 3.1	Command was introduced

Functional Notes

When this command is enabled, the IP destination address of each packet must be validated before being passed through. If the packet is not acceptable per these settings, it is dropped.

Usage Examples

The following example sets up the router to only allow Telnet traffic into the Ethernet interface:

(config)# **ip access-list extended TelnetOnly**
(config-ext-nacl)#**permit tcp any any eq telnet**
(config-ext-nacl)#**interface eth 0/1**
(config-eth 0/1)#**ip access-group TelnetOnly in**

ip address dhcp

Use the **ip address dhcp** command to use Dynamic Host Configuration Protocol (DHCP) to obtain an address on the Ethernet interface. Use the **no** form of this command to remove a configured IP address (using DHCP) and disable DHCP operation on the interface.

ip address dhcp {**client-id** [<*interface*> | <*identifier*>] **hostname** *"<string>"* }

Syntax Description

client-id ***Optional**	Specifies the client identifier used when obtaining an IP address from a DHCP server.
<*interface*>	Specifying an interface defines the client identifier as the hexadecimal MAC address of the specified interface (including a hexadecimal number added to the front of the MAC address to identify the media type). For example, specifying the **client-id ethernet 0/1** (where the Ethernet interface has a MAC address of d217.0491.1150) defines the client identifier as **01:d2:17:04:91:11:50** (where 01 defines the media type as Ethernet).
<*identifier*>	Specify a custom client-identifier using a text string (that is converted to a hexadecimal equivalent) or 7 to 28 hexadecimal numbers (with colon delimiters). For example, a custom client identifier of **0f:ff:ff:ff:ff:51:04:99:a1** may be entered using the <*identifier*> option.
host-name ***Optional**	Specifies a text string (to override the global router name) to use as the name in the DHCP option 12 field.
"<string>"	String (encased in quotation marks) of up to 35 characters to use as the name of the host for DHCP operation.

Default Values

client-id ***Optional**	By default, the client identifier is populated using the following formula: TYPE: INTERFACE SPECIFIC INFO : MAC ADDRESS Where TYPE specifies the media type in the form of one hexadecimal byte (refer to the **hardware-address** command for a detailed listing of media types) and the MAC ADDRESS is the Media Access Control (MAC) address assigned to the first Ethernet interface in the unit in the form of six hexadecimal bytes. (For units with a single Ethernet interface, the MAC ADDRESS assigned to ethernet 0/1 is used in this field). INTERFACE SPECIFIC INFO is only used for frame relay interfaces and can be determined using the following: FR_PORT# : Q.922 ADDRESS Where the FR_PORT# specifies the label assigned to the virtual frame relay

Default Values (Continued)

interface using four hexadecimal bytes. For example, a virtual frame relay interface labeled 1 would have a FR_PORT# of 00:00:00:01.

The Q.922 ADDRESS field is populated using the following:

8	7	6	5	4	3	2	1
DLCI (high order)						C/R	EA
DLCI (lower)				FECN	BECN	DE	EA

Where the FECN, BECN, C/R, DE, and high order EA bits are assumed to be 0 and the lower order extended address (EA) bit is set to 1.

The following list provides a few example DLCIs and associated Q.922 address:
DLCI (decimal) / Q.922 address (hex):

16 / 0x0401
50 / 0x0C21
60 / 0x0CC1
70 / 0x1061
80 / 0x1401

host-name
Optional

"<string>" By default, the hostname is the name configured using the Global Configuration **hostname** command.

Command Modes

(config-eth 0/1)# Ethernet Interface Configuration Mode required

Command History

Release 2.1 Command was introduced

Functional Notes

Dynamic Host Configuration Protocol (DHCP) allows interfaces to acquire a dynamically assigned IP address from a configured DHCP server on the network. Many Internet Service Providers (ISPs) require the use of DHCP when connecting to their services. Using DHCP reduces the number of dedicated IP addresses the ISP must obtain. Consult your ISP to determine the proper values for the **client-id** and **hostname** fields.

Usage Examples

The following example enables DHCP operation on Ethernet interface 0/1:

(config)# **interface eth 0/1**
(config-eth 0/1)# **ip address dhcp**

ip address <*address*> <*mask*> secondary

Use the **ip address** command to define an IP address on the specified interface. Use the optional **secondary** keyword to define a secondary IP address. Use the **no** form of this command to remove a configured IP address.

Syntax Description

<*address*>	Defines the IP address for the interface in dotted decimal notation (for example: 192.22.73.101).
<*mask*>	Specifies the subnet mask that corresponds to the listed IP address.
secondary **Optional*	Optional keyword used to configure a secondary IP address for the specified interface.

Default Values

By default, there are no assigned IP addresses.

Command Modes

(config-interface)#	Interface Configuration Mode required (applies to all physical interfaces as well as virtual interfaces)

Command History

Release 1.1	Command was introduced
Release 2.1	Added **ip address dhcp** for DHCP client support

Functional Notes

Use secondary IP addresses to allow dual subnets on a single interface (when you need more IP addresses than the primary subnet can provide). When using secondary IP addresses, avoid routing loops by verifying that all devices on the network segment are configured with secondary IP addresses on the secondary subnet.

Usage Examples

The following example configures a secondary IP address of **192.22.72.101/30**:

(config)# **interface ethernet 0/1**
(config-eth 0/1)# **ip address 192.22.72.101 255.255.255.252 secondary**

ip dhcp release

Use the **ip dhcp release** command to transmit a message to the DHCP server requesting termination of the IP address lease on that interface.

WARNING	*If you are currently connected to the unit using a Telnet session through the Ethernet interface, using the **ip dhcp release** command will terminate your Telnet session and render your Telnet capability inoperable until a new IP address is assigned by the DHCP server.*

Syntax Description

No subcommands

Default Values

No defaults necessary for this command.

Command Modes

(config-eth 0/1)#	Ethernet Interface Configuration Mode required

Command History

Release 2.1	Command was introduced

Functional Notes

Dynamic Host Configuration Protocol (DHCP) allows interfaces to acquire a dynamically-assigned IP address from a configured DHCP server on the network. Many Internet Service Providers (ISPs) require the use of DHCP when connecting to their services. Using DHCP reduces the number of dedicated IP addresses the ISP must obtain. Consult your ISP to determine the proper values for the **client-id** and **hostname** fields.

Usage Examples

The following example releases the IP address assigned (by DHCP) on the Ethernet interface (eth 0/1):

(config)# **int eth 0/1**
(config-eth 0/1)# **ip dhcp release**

ip dhcp renew

Use the **ip dhcp renew** command to transmit a message to the DHCP server requesting renewal of the IP address lease on that interface.

Default Values

No defaults necessary for this command.

Command Modes

(config-eth 0/1)#	Ethernet Interface Configuration Mode required

Command History

Release 2.1	Command was introduced

Functional Notes

Dynamic Host Configuration Protocol (DHCP) allows interfaces to acquire a dynamically assigned IP address from a configured DHCP server on the network. Many Internet Service Providers (ISPs) require the use of DHCP when connecting to their services. Using DHCP reduces the number of dedicated IP addresses the ISP must obtain. Consult your ISP to determine the proper values for the **client-id** and **hostname** fields.

Usage Examples

The following example renews the IP address assigned (by DHCP) on the Ethernet interface (eth 0/1):

(config)# **int eth 0/1**
(config-eth 0/1)# **ip dhcp renew**

ip helper-address *<address>*

Use the **ip helper-address** command to configure the ADTRAN OS to forward User Datagram Protocol (UDP) broadcast packets received on the interface. Use the **no** form of this command to disable forwarding packets.

> NOTE *The **ip helper** command must be used in conjunction with the **ip forward-protocol** command to configure the ADTRAN OS to forward UDP broadcast packets. See ip forward-protocol udp <port number> on page 200 for more information.*

Syntax Description

<address>	Specifies the destination IP address (in dotted decimal notation) for the forwarded UDP packets

Default Values

By default, broadcast UDP packets are not forwarded.

Command Modes

(config-interface)#	Interface Configuration Mode required (applies to all physical interfaces as well as virtual interfaces)

Command History

Release 1.1	Command was introduced

Functional Notes

When used in conjunction with the **ip forward-protocol** command, the **ip helper-address** feature allows you to customize which broadcast packets are forwarded.

To implement the helper address feature, assign a helper-address(es) (specifying the device that needs to receive the broadcast traffic) to the interface closest to the host that transmits the broadcast packets. When broadcast packets (of the specified type forwarded using the **ip forward-protocol** command) are received on the interface, they will be forwarded to the device that needs the information.

Only packets meeting the following criteria are considered eligible by the **ip helper-address** feature:
1. The packet IP protocol is User Datagram Protocol (UDP).
2. Any UDP port specified using the **ip forward-protocol** command.
3. The media access control (MAC) address of the frame is an all-ones broadcast address (ffff.ffff.ffff).
4. The destination IP address is broadcast defined by all ones (255.255.255.255) or a subnet broadcast (for example, 192.33.4.251 for the 192.33.4.248/30 subnet).

Usage Examples

The following example forwards all DNS broadcast traffic to the DNS server with IP address 192.33.5.99:

(config)# **ip forward-protocol udp domain**
(config)# **interface eth 0/1**
(config-eth 0/1)# **ip helper-address 192.33.5.99**

ip ospf

Use the **ip ospf** command to customize OSPF settings (if needed).

Syntax Description

authentication-key <*password*>	Assign a simple-text authentication password to be used by other routers using the OSPF simple password authentication.
cost <*value*>	Specify the OSPF cost of sending a packet on the interface. This value overrides any computed cost value. Range: 1-65535.
dead-interval <*seconds*>	Set the maximum interval allowed between hello packets. If the maximum is exceeded, neighboring devices will determine that the device is down. Range: 0-32767.
hello-interval <*seconds*>	Specify the interval between hello packets sent on the interface. Range: 0-32767.
message-digest-key <*keyid*> **md5** <*key*>	Configure OSPF Message Digest 5 (MD5) authentication (16-byte max) keys.
priority <*value*>	Set the OSPF priority. The value set in this field helps determine the designated router for this network. Range: 0-255.
retransmit-interval <*seconds*>	Specify the time between link-state advertisements (LSAs). Range: 0-32767.
transmit-delay <*seconds*>	Set the estimated time required to send an LSA on the interface. Range: 0-32767.

Default Values

retransmit-interval <*seconds*>	5 seconds
transmit-delay <*seconds*>	1 second
hello-interval <*seconds*>	10 seconds: Ethernet, point-to-point, frame relay, and ppp
dead-interval <*seconds*>	40 seconds

Command Modes

(config-interface)#	Valid interfaces include: Ethernet (eth 0/1), virtual frame relay (fr 1), and virtual PPP (ppp 1).

Command History

Release 3.1	Command was introduced

ip ospf authentication [message-digest | null]

Use the **ip ospf authentication** command to authenticate an interface that is performing OSPF authentication.

Syntax Description

message-digest **Optional*	Select message-digest authentication type.
null **Optional*	Select for no authentication to be used.

Default Values

By default, this is set to null (meaning no authentication is used).

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), virtual frame relay sub-interfaces (fr 1.20), and loopback interfaces

Command History

Release 3.1	Command was introduced

Usage Examples

The following example specifies that no authentication will be used on the Ethernet interface:

(config-eth 0/1)# **ip ospf authentication null**

ip ospf network [broadcast | point-to-point]

Use the **ip ospf network** command to specify the type of network on this interface.

Syntax Description

broadcast	Set the network type for broadcast.
point-to-point	Set the network type for point-to-point.

Default Values

By default, Ethernet defaults to broadcast. PPP and frame relay default to point-to-point.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), virtual frame relay sub-interfaces (fr 1.20), and loopback interfaces

Command History

Release 3.1	Command was introduced

Functional Notes

A point-to-point network will not elect designated routers.

Usage Examples

The following example designates a broadcast network type:

(config-eth 0/1)# **ip ospf network broadcast**

ip proxy-arp

Use the **ip proxy-arp** to enable proxy Address Resolution Protocol (ARP) on the interface. Use the **no** form of this command to disable this feature.

Syntax Description

<address>	Defines the IP address for the interface in dotted decimal notation (for example: 192.22.73.101)
<subnet mask>	Specifies the subnet mask that corresponds to the listed IP address

Default Values

By default, proxy arp is enabled.

Command Modes

(config-interface)#	Interface Configuration Mode required (applies to all physical interfaces as well as virtual interfaces)

Command History

Release 1.1	Command was introduced

Functional Notes

In general, the principle of proxy-arp allows a router to insert its IP address in the source IP address field of a packet (if the packet is from a host on one of its subnetworks). This allows hosts to reach devices on other subnetworks without implementing routing or specifying a default gateway.

If proxy-arp is enabled, the ADTRAN OS will respond to all proxy-arp requests with its specified MAC address and forward packets accordingly.

Enabling proxy-arp on an interface may introduce unnecessary ARP traffic on the network.

Usage Examples

The following enables proxy-arp on the Ethernet interface:

(config)#**interface ethernet 0/1**
(config-eth 0/1)# **ip proxy-arp**

ip rip receive version *<version>*

Use the **ip rip receive version** command to configure the RIP version the unit accepts in all RIP packets received on the interface.

Syntax Description

<version>	Specifies the RIP version
1	Only accept received RIP version 1 packets on the interface
2	Only accept received RIP version 2 packets on the interface

Default Values

*By default, all interfaces implement RIP version 1 (the default value for the **version** command).*

Command Modes

(config-interface)#	Interface Configuration Mode required (applies to all physical interfaces as well as virtual interfaces)

Command History

Release 1.1	Command was introduced

Functional Notes

Use the **ip rip receive version** to specify a RIP version that overrides the **version** (in the Router RIP command set) configuration.

The ADTRAN OS only accepts one version (either 1 or 2) on a given interface.

Usage Examples

The following example configures the Ethernet interface to accept only RIP version 2 packets:

(config)# **interface eth 0/1**
(config-eth 0/1)# **ip rip receive version 2**

ip rip send version *<version>*

Use the **ip rip send version** command to configure the RIP version the unit sends in all RIP packets transmitted on the interface.

Syntax Description

<version>	Specifies the RIP version
1	Only transmits RIP version 1 packets on the interface
2	Only transmits RIP version 2 packets on the interface

Default Values

*By default, all interfaces transmit RIP version 1 (the default value for the **version** command).*

Command Modes

(config-interface)#	Interface Configuration Mode required (applies to all physical interfaces as well as virtual interfaces)

Command History

Release 1.1	Command was introduced

Functional Notes

Use the **ip rip send version** to specify a RIP version that overrides the **version** (in the Router RIP command set) configuration.

The ADTRAN OS only transmits one version (either 1 or 2) on a given interface.

Usage Examples

The following example configures the Ethernet interface to transmit only RIP version 2 packets:

(config)# **interface eth 0/1**
(config-eth 0/1)# **ip rip send version 2**

ip route-cache

Use the **ip route-cache** command to enable fast-cache switching on the interface. Use the **no** form of this command to disable fast-cache switching and return to process switching mode.

*Using Network Address Translation (NAT) or the ADTRAN OS firewall capabilities on an interface requires process switching mode (using the **no ip route-cache** command).*

Syntax Description

No subcommands.

Default Values

By default, fast-cache switching is enabled on all Ethernet and virtual frame relay sub-interfaces. IP route-cache is disabled for all virtual PPP interfaces.

Command Modes

(config-interface)#	Interface Configuration Mode required Valid interfaces include: Ethernet (eth 0/1), virtual frame relay sub-interfaces (fr 1.16), and virtual PPP interfaces (ppp 1).

Command History

Release 2.1	Command was introduced

Functional Notes

Fast switching allows an IP interface to provide optimum performance when processing IP traffic.

Usage Examples

The following example enables fast switching on the Ethernet interface:

```
(config)# interface ethernet 0/1
(config-eth 0/1)# ip route-cache
```


ip unnumbered *<interface>*

Use the **ip unnumbered** command to use the IP address assigned to the specified interface for all IP processing on the active interface. Use the **no** form of this command to remove the unnumbered configuration.

Syntax Description

<interface>	Specifies the interface (in the format **type slot/port**) that contains the IP address to be used as the source address for all packets transmitted on this interface Valid interfaces include: Ethernet (eth 0/1), virtual frame relay sub-interfaces (fr 1.16), virtual PPP interfaces (ppp 1), and loopback interfaces.

Default Values

*By default, all interfaces are configured to use a specified IP address (using the **ip address** command).*

Command Modes

(config-interface)#	Interface Configuration Mode required Valid interfaces include: Ethernet (eth 0/1), virtual frame relay sub-interfaces (fr 1.16), virtual PPP interfaces (ppp 1), and loopback interfaces

Command History

Release 1.1	Command was introduced

Functional Notes

If **ip unnumbered** is enabled on an interface, all IP traffic from the interface will use a source IP address taken from the specified interface. For example, specifying **ip unnumbered ppp 1** while in the Ethernet Interface Configuration Mode configures the Ethernet interface to use the IP address assigned to the PPP interface for all IP processing. In addition, the ADTRAN OS uses the specified interface information when sending route updates over the unnumbered interface.

Usage Examples

The following example configures the Ethernet interface (labeled **eth 0/1**) to use the IP address assigned to the PPP interface (**ppp 1**):

```
(config)# interface eth 0/ 1
(config-eth 0/1)# ip unnumbered ppp 1
```

mac-address *<address>*

Use the **mac-address** command to specify the Media Access Control (MAC) address of the unit. Only the last three values of the MAC address can be modified. The first three values are read-only and contain the ADTRAN reserved number (00:0A:C8). Use the **no** form of this command to return to the default MAC address programmed by ADTRAN.

Syntax Description

<address>	MAC address entered in a series of six dual-digit hexadecimal values separated by colons (for example 00:0A:C8:5F:00:D2)

Default Values

A unique default MAC address is programmed in each unit shipped by ADTRAN.

Command Modes

(config-eth 0/1)#	Ethernet Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example configures a MAC address of **00:0A:C8:5F:00:D2**:

(config)# **interface ethernet 0/1**
(config-eth 0/1)# **mac-address 00:0A:C8:5F:00:D2**

mtu <*size*>

Use the **mtu** command to configure the maximum transmit unit (MTU) size for the active interface. Use the **no** form of this command to return to the default value.

Syntax Description

<*size*>	Configures the window size for transmitted packets. The valid ranges for the various interfaces are listed below:	
	Ethernet (eth 0/1)	64 to 1500
	virtual frame relay sub-interfaces (fr 1.16)	64 to 1520
	virtual PPP interfaces (ppp 1)	64 to 1500
	loopback interfaces	64 to 1500

Default Values

<*size*>	The default values for the various interfaces are listed below:	
	Ethernet (eth 0/1)	1500
	virtual frame relay sub-interfaces (fr 1.16)	1500
	virtual PPP interfaces (ppp 1)	1500
	loopback interfaces	1500

Command Modes

(config-interface)#	Interface Configuration Mode required (applies only to IP interfaces) Valid interfaces include: Ethernet (eth 0/1), virtual frame relay sub-interfaces (fr 1.16), virtual PPP interfaces (ppp 1), and loopback interfaces.

Command History

Release 1.1	Command was introduced

Functional Notes

OSPF will not become adjacent on links where the MTU sizes do not match. If router A and router B are exchanging hello packets but their MTU sizes do not match, they will never reach adjacency. This is by design and required by the RFC.

Usage Examples

The following example specifies an MTU of 1200 on the Ethernet interface:

```
(config)# interface eth 0/1
(config-eth 0/1)# mtu 1200
```

snmp trap

Use the **snmp trap** command to enable all supported Simple Network Management Protocol (SNMP) traps on the interface.

Syntax Description

No subcommands

Default Values

By default, all interfaces (except virtual frame relay interfaces and sub-interfaces) have SNMP traps enabled.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), DDS (dds 1/1), serial (ser 1/1), virtual frame relay (fr 1), and SHDSL (shdsl 1/1) interfaces.

Command History

Release 1.1	Command was introduced

Usage Examples

The following example enables SNMP capability on the Ethernet interface:

(config)# **interface eth 0/1**
(config-eth 0/1)# **snmp trap**

speed *<rate>*

Use the **speed** command to configure the speed of an Ethernet interface. Use the **no** form of this command to return to the default value.

Syntax Description

<rate>	Defines the rate of the Ethernet port
10	10 Mb Ethernet
100	100 Mb Ethernet
auto ***Optional**	Automatically detects 10 or 100 Mb Ethernet and negotiates the duplex setting

*If the **speed** is manually set to **10** or **100,** the duplex must be manually configured as **full-duplex** or **half-duplex**.*

Default Values

<rate>	**auto**

Command Modes

(config-eth 0/1)#	Ethernet Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example configures the Ethernet port for 100 Mb operation:

(config)# **interface ethernet 0/1**
(config-eth 0/1)# **speed 100**

DDS INTERFACE CONFIGURATION COMMAND SET

To activate the DDS Interface Configuration command set, enter the **interface dds** command at the Global Configuration Mode prompt. For example:

```
Router> enable
Router# configure terminal
Router(config)# interface dds 1/1
Router(config-dds 1/1)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

alias <"text"> on page 556

description on page 557

do on page 558

end on page 559

exit on page 560

shutdown on page 561

All other commands for this command set are described in this section in alphabetical order.

clock rate <rate> on page 327

clock source <option> on page 328

data-coding scrambled on page 329

loopback [dte | line | remote] on page 330

remote-loopback on page 331

snmp trap on page 332

snmp trap link-status on page 333

clock rate *<rate>*

Use the **clock rate** command to configure the data rate used as the operating speed for the interface. This rate should match the rate required by the DDS service provider. Use the **no** form of this command to return to the default value.

Syntax Description

<rate>	Configures the operating speed used for the interface
auto	Automatically detects the clock rate and sets to match
bps56k	Sets the clock rate to 56 kbps
bps64k	Sets the clock rate to 64 kbps

Default Values

<rate>	**auto**

Command Modes

(config-dds 1/1)#	56K/64K (DDS) Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

When operating at 64 kbps (clear channel operation), the DTE data sequences may mimic network loop maintenance functions and erroneously cause other network elements to activate loopbacks. Use the **data-coding scrambled** command to prevent such occurrences. See *data-coding scrambled* on page 329 for related information.

Usage Examples

The following example configures the clock rate for 56 kbps operation:

(config)# **interface dds 1/1**
(config-dds 1/1)# **clock rate bps56k**

clock source <*option*>

Use the **clock source** command to configure the source timing used for the interface. The clock specified using the **clock source** command is also the system master clock. Use the **no** form of this command to return to the default value.

Syntax Description

<*option*>	Configures the timing source for the DDS interface.
line	Configures the unit to recover clocking from the circuit.
internal	Configures the unit to provide clocking using the internal oscillator.

Default Values

<*option*>	**line**

Command Modes

(config-dds 1/1)#	56K/64K (DDS) Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

When operating on a DDS network, the **clock source** should be **line**. On a point-to-point private network, one unit must be **line** and the other **internal**.

Usage Examples

The following example configures the unit to recover clocking from the circuit:

(config)# **interface dds 1/1**
(config-dds 1/1)# **clock source line**

data-coding scrambled

Use the **data-coding scrambled** command to enable the ADTRAN OS scrambler to combine user data with pattern data to ensure user data does not mirror standard DDS loop codes. The scrambler may only be used on 64 kbps circuits without frame relay signaling (clear channel).

Syntax Description

No subcommands

Default Values

By default, the scrambler is disabled.

Command Modes

(config-dds 1/1)#	56K/64K (DDS) Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

When operating at 64 kbps (clear channel operation), there is a possibility the DTE data sequences may mimic network loop maintenance functions and erroneously cause other network elements to activate loopbacks. Use the **data-coding scrambled** command to prevent such occurrences. Do not use this command if using frame relay or if using PPP to another device other than an AOS product also running scrambled.

Usage Examples

The following example enables the ADTRAN OS scrambler:

(config)# **interface dds 1/1**
(config-dds 1/1)# **data-coding scrambled**

loopback [dte | line | remote]

Use the **loopback** command to initiate a specified loopback on the interface. Use the **no** form of this command to deactivate the loop.

Syntax Description

dte	Initiates a loop to connect the transmit and receive path through the unit.
line	Initiates a loop of the DDS circuit towards the network by connecting the transmit path to the receive path.
remote	Transmits a DDS loop code over the circuit to the remote unit. In response, the remote unit should initiate a line loopback.

Default Values

No default values necessary for this command.

Command Modes

(config-dds 1/1)#	56K/64K (DDS) Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example activates a line loopback on the DDS interface:

(config)# **interface dds 1/1**
(config-dds 1/1)# **loopback line**

remote-loopback

Use the **remote-loopback** command to configure the interface to respond to loopbacks initiated by a remote unit (or the service provider). Use the **no** form of this command to disable this feature.

Syntax Description

No subcommands.

Default Values

By default, all interfaces respond to remote loopbacks.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: T1 (t1 1/1) and DDS (dds 1/1)

Command History

Release 1.1	Command was introduced

Usage Examples

The following example enables remote loopbacks on the DDS interface:

(config)# **interface dds 1/1**
(config-dds 1/1)# **remote-loopback**

snmp trap

Use the **snmp trap** command to enable all supported Simple Network Management Protocol (SNMP) traps on the interface.

Syntax Description

No subcommands

Default Values

By default, all interfaces (except virtual frame relay interfaces and sub-interfaces) have SNMP traps enabled.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), DDS (dds 1/1), serial (ser 1/1), virtual frame relay (fr 1), and SHDSL (shdsl 1/1) interfaces

Command History

Release 1.1	Command was introduced

Usage Examples

The following example enables SNMP capability on the DDS interface:

(config)# **interface dds 1/1**
(config-dds 1/1)# **snmp trap**

snmp trap link-status

Use the **snmp trap link-status** command to control the SNMP variable that enables (or disables) the interface to send SNMP traps when there is an interface status change (ifLinkUpDownTrapEnable of RFC 2863). Use the **no** form of this command to disable this trap.

Syntax Description

No subcommands

Default Values

By default, the ifLinkUpDownTrapEnable OID is set to enabled for all supported interfaces except virtual frame relay interfaces.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), T1 (t1 1/1), DSX-1 (t1 1/2), serial (ser 1/1), DDS (dds 1/1), virtual frame relay (fr 1), virtual PPP (ppp 1), and SHDSL (shdsl 1/1) interfaces.

Command History

Release 1.1	Command was introduced

Functional Notes

The **snmp trap link-status** command is used to control the RFC 2863 ifLinkUpDownTrapEnable OID (OID number 1.3.6.1.2.1.31.1.1.1.14.0).

Usage Examples

The following example disables the link-status trap on the DDS interface:

```
(config)# interface dds 1/1
(config-dds 1/1)# no snmp trap link-status
```

T1 INTERFACE CONFIGURATION COMMAND SET

To activate the T1 Interface Configuration command set, enter the **interface t1** command at the Global Configuration Mode prompt. For example:

```
Router> enable
Router# configure terminal
Router(config)# interface t1 1/1
Router(config-t1 1/1)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

All other commands for this command set are described in this section in alphabetical order.

clock source <*option*>

Use the **clock source** command to configure the source timing used for the interface. The clock specified using the **clock source** command is also the system master clock. Use the **no** form of this command to return to the default value.

Syntax Description

<*option*>	Configures the timing source for the T1 interface.
internal	Configures the unit to provide clocking using the internal oscillator.
line	Configures the unit to recover clocking from the primary circuit.
through	Configures the unit to recover clocking from the circuit connected to the DSX-1 interface.

Default Values

<*option*>	**line**

Command Modes

(config-t1 1/1)#	T1 or DSX-1 Interface Configuration Mode required.
(config-t1 1/2)#	

Command History

Release 1.1	Command was introduced.

Functional Notes

When operating on a circuit that is providing timing, setting the **clock source** to **line** can avoid errors such as Clock Slip Seconds (CSS).

Usage Examples

The following example configures the unit to recover clocking from the circuit:

(config)# **interface t1 1/1**
(config-t1 1/1)# **clock source line**

coding [ami | b8zs]

Use the **coding** command to configure the line coding for a T1 or DSX-1 physical interface. This setting must match the line coding supplied on the circuit by the provider.

Syntax Description

ami	Configures the line coding for alternate mark inversion.
b8zs	Configures the line coding for bipolar eight zero substitution.

Default Values

By default, all T1 interfaces are configured with B8ZS line coding.

Command Modes

(config-t1 1/1)#	T1 or DSX-1 Interface Configuration Mode required.
(config-t1 1/2)#	

Command History

Release 1.1	Command was introduced.

Functional Notes

The line coding configured in the unit must match the line coding of the T1 circuit. A mismatch will result in line errors (e.g., BPVs).

Usage Examples

The following example configures the T1 interface for AMI line coding:

(config)# **interface t1 1/1**
(config-t1 1/1)# **coding ami**

fdl [ansi | att | none]

Use the **fdl** command to configure the format for the facility data link channel on the T1 circuit. FDL channels are only available on point-to-point circuits. Use the **no** form of this command to return to the default value.

Syntax Description

ansi	Configures the FDL for ANSI T1.403 standard
att	Configures the FDL for ATT TR54016 standard
none	No FDL available on this circuit

Default Values

<format>	**ansi**

Command Modes

(config-t1 1/1)#	T1 Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

T1 circuits using ESF framing format (specified using the **framing** command) reserve 12 bits as a data link communication channel, referred to as the Facility Data Link (FDL), between the equipment on either end of the circuit. The FDL allows the transmission of trouble flags such as the Yellow Alarm signal. See *framing [d4 | esf]* on page 338 for related information.

Usage Examples

The following example disables the FDL channel for the T1 circuit:

```
(config)# interface t1 1/1
(config-t1 1/1)# fdl none
```


framing [d4 | esf]

Use the **framing** command to configure the framing format for the T1 or DSX-1 interface. This parameter should match the framing format supplied by your network provider. Use the **no** form of this command to return to the default value.

Syntax Description

d4	D4 superframe format (SF)
esf	Extended SF

Default Values

<*format*>	**esf**

Command Modes

(config-t1 1/1)#	T1 or DSX-1 Interface Configuration Mode required.
(config-t1 1/2)#	

Command History

Release 1.1	Command was introduced

Functional Notes

A frame is comprised of a single byte from each of the T1's timeslots; there are 24 timeslots on a single T1 circuit. Framing bits are used to separate the frames and indicate the order of information arriving at the receiving equipment. D4 and ESF are two methods of collecting and organizing frames over the circuit.

Usage Examples

The following example configures the T1 interface for D4 framing:

(config)# **interface t1 1/1**
(config-t1 1/1)# **framing d4**

lbo <*value*>

Use the **lbo** command to set the line build out (in dB) for the T1 interface. Use the **no** form of this command to return to the default value

Syntax Description

<*value*>	Configures the line build out for the T1 interface Valid options include: 0, -7.5, -15, and -22.5 dB

Default Values

<*value*>	0 dB

Command Modes

(config-t1 1/1)#	T1 Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

Line build out (LBO) is artificial attenuation of a T1 output signal to simulate a degraded signal. This is useful to avoid overdriving a receiver's circuits. The shorter the distance between T1 equipment (measured in cable length), the greater the attenuation value. For example, two units in close proximity should be configured for the maximum attention (-22.5 dB).

Usage Examples

The following example configures the T1 interface LBO for -22.5 dB:

(config)# **interface t1 1/1**
(config-t1 1/1)# **lbo -22.5**

loopback network [line | payload]

Use the **loopback network** command to initiate a loopback on the interface toward the network. Use the **no** form of this command to deactivate the loopback.

Syntax Description

line	Initiates a metallic loopback of the physical T1 network interface.
payload	Initiates a loopback of the T1 framer (CSU portion) of the T1 network interface.

Default Values

No default necessary for this command.

Command Modes

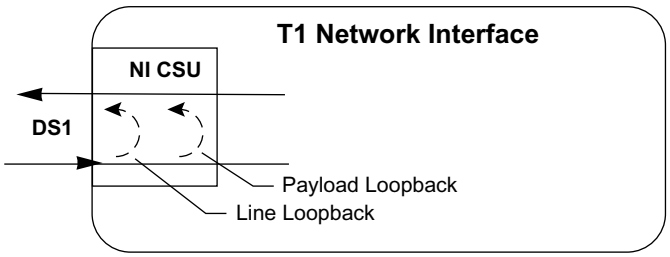
(config-t1 1/1)#	T1 or DSX-1 Interface Configuration Mode required.
(config-t1 1/2)#	

Command History

Release 1.1	Command was introduced

Functional Notes

The following diagram depicts the difference between a line and payload loopback.



Usage Examples

The following example initiates a payload loopback of the T1 interface:

(config)# **interface t1 1/1**
(config-t1 1/1)# **loopback network payload**

loopback remote line [fdl | inband]

Use the **loopback remote line** command to send a loopback code to the remote unit to initiate a line loopback. Use the **no** form of this command to send a loopdown code to the remote unit to deactivate the loopback.

Syntax Description

fdl	Uses the facility data link (FDL) to initiate a full 1.544 Mbps loopback of the signal received by the remote unit from the network.
inband	Uses the inband channel to initiate a full 1.544 Mbps physical loopback (metallic loopback) of the signal received from the network.

Default Values

No defaults necessary for this command.

Command Modes

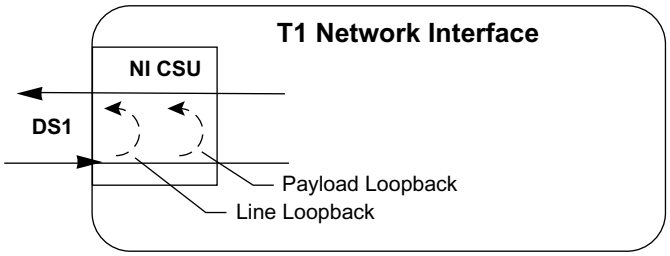
(config-t1 1/1)#	T1 Interface Configuration Mode required (does not apply to DSX-1 interfaces)

Command History

Release 1.1	Command was introduced

Functional Notes

The following diagram depicts the difference between a line and payload loopback.



Usage Examples

The following example initiates a remote line loopback using the FDL:

(config)# **interface t1 1/1**
(config-t1 1/1)# **loopback remote line fdl**

loopback remote payload

Use the **loopback remote payload** command to send a loopback code to the remote unit to initiate a payload loopback. A payload loopback is a 1.536 Mbps loopback of the payload data received from the network maintaining bit-sequence integrity for the information bits by synchronizing (regenerating) the timing. Use the **no** form of this command to send a loopdown code to the remote unit to deactivate the loopback.

Syntax Description

No subcommands

Default Values

No defaults necessary for this command.

Command Modes

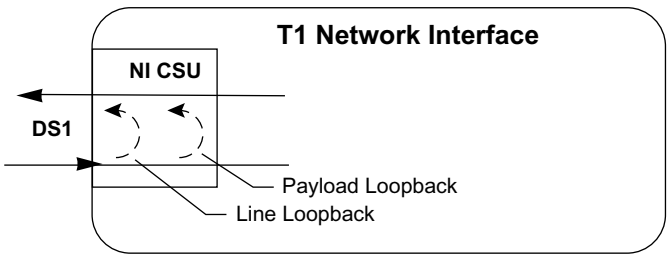
(config-t1 1/1)#	T1 or DSX-1 Interface Configuration Mode required.
(config-t1 1/2)#	

Command History

Release 1.1	Command was introduced

Functional Notes

The following diagram depicts the difference between a line and payload loopback.



Usage Examples

The following example initiates a remote payload loopback:

(config)# **interface t1 1/1**
(config-t1 1/1)# **loopback remote payload**

remote-loopback

Use the **remote-loopback** command to configure the interface to respond to loopbacks initiated by a remote unit (or the service provider). Use the **no** form of this command to disable this feature.

Syntax Description

No subcommands.

Default Values

By default, all interfaces respond to remote loopbacks.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: T1 (t1 1/1) and DDS (dds 1/1)

Command History

Release 1.1	Command was introduced

Usage Examples

The following example enables remote loopbacks on the T1 interface:

(config)# **interface t1 1/1**
(config-t1 1/1)# **remote-loopback**

show [p511]

Use the **show** command to display the current status of T1 tests, including information regarding loopbacks and test patterns.

Syntax Description

p511	511-bit repeating pattern of ones and zeros

Default Values

No defaults required for this command.

Command Modes

(config-t1 1/1)#	T1 Interface Configuration Mode required (does not apply to DSX-1 interfaces)

Command History

Release 2.1	Command was introduced

Usage Examples

The following example configures the T1 interface to display the P511 status:

```
(config)# interface t1 1/1
(config-t1 1/1)# show p511
P511 Errored Seconds: 12
```


snmp trap line-status

Use the **snmp trap line-status** command to control the Simple Network Management Protocol (SNMP) variable dsx1LineStatusChangeTrapEnable (RFC 2495) to enable (or disable) the interface to send SNMP traps when there is an interface status change. The dsx1LineStatusChangeTrapEnable variable is set to enabled by default.

Syntax Description

No subcommands

Default Values

By default, the dsx1LineStatusChangeTrapEnable OID is set to enabled for all interfaces except virtual frame relay interfaces.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: T1 (t1 1/1), DSX-1 (t1 1/2), serial (ser 1/1), DDS (dds 1/1), virtual frame relay interfaces (fr 1), and virtual PPP interfaces (ppp 1).

Command History

Release 1.1	Command was introduced

Functional Notes

The **snmp trap line-status** command is used to control the RFC 2495 dsx1LineStatusChangeTrapEnable OID (OID number 1.3.6.1.2.1.10.18.6.1.17.0).

Usage Examples

The following example disables the line-status trap on the T1 interface:

(config)# **interface t1 1/1**
(config-t1 1/1)# **no snmp trap line-status**

snmp trap link-status

Use the **snmp trap link-status** to control the SNMP variable ifLinkUpDownTrapEnable (RFC 2863) to enable (or disable) the interface to send SNMP traps when there is an interface status change. Use the **no** form of this command to disable this trap.

Syntax Description

No subcommands

Default Values

By default, the ifLinkUpDownTrapEnable OID is set to enabled for all interfaces except virtual frame relay interfaces.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), T1 (t1 1/1), DSX-1 (t1 1/2), serial (ser 1/1), DDS (dds 1/1), virtual frame relay (fr 1), virtual PPP (ppp 1), and SHDSL (shdsl 1/1) interfaces.

Command History

Release 1.1	Command was introduced

Functional Notes

The **snmp trap link-status** command is used to control the RFC 2863 ifLinkUpDownTrapEnable OID (OID number 1.3.6.1.2.1.31.1.1.1.14.0).

Usage Examples

The following example disables the link-status trap on the T1 interface:

(config)# **interface t1 1/1**
(config-t1 1/1)# **no snmp trap link-status**

tdm-group *<group number>* **timeslots** *<1-24>* **speed** [**56 | 64**]

Use the **tdm-group** command to create a group of contiguous DS0s on this interface to be used during the **cross-connect** process. See *cross-connect <#> <from interface> <slot/port> <tdm-group#> <to interface> <slot/port>* on page 147 for related information.

*Changing **tdm-group** settings could potentially result in service interruption.*

Syntax Description

<group number>	Numerical label to identify the created tdm-group (valid range: 1-255).
timeslots	Keyword to specify the DS0s to be used in this tdm-group.
<1-24>	Specifies the DS0s to be used in the tdm-group. This can be entered as a single number representing one of the 24 T1 channel timeslots or as a contiguous group of DS0s. (For example, 1-10 specifies the first 10 channels of the T1.)
speed ***Optional**	Keyword to specify the individual DS0 rate on the T1 interface. If the **speed** keyword is not used, the ADTRAN OS assumes a DS0 rate of 64 kbps.
56	Specifies a DS0 rate of 56 kbps.
64	Specifies a DS0 rate of 64 kbps.

Default Values

By default, there are no configured tdm-groups.

Command Modes

(config-t1 1/1)#	T1 Interface Configuration Mode required (does not apply to DSX-1 interfaces)

Command History

Release 1.1	Command was introduced

Usage Examples

The following example creates a tdm-group (labeled **5**) of 10 DS0s at 64 kbps each:

(config)# **interface t1 1/1**
(config-t1 1/1)# **tdm-group 5 timeslots 1-10 speed 64**

test-pattern [ones | zeros | clear | insert | p511]

Use the **test-pattern** command to activate the built-in pattern generator and begin sending the specified test pattern. This pattern generation can be used to verify a data path when used in conjunction with an active loopback. Use the **no** form of this command to cease pattern generation.

Syntax Description

ones	Generate continous ones.
zeros	Generate continous zeros.
clear	Clears the test pattern error count on the T1 interface.
insert	Inserts an error into the generated test pattern being transmitted on the T1 interface. The injected error result is displayed using the **show p511** command.
p511	511-bit repeating pattern of ones and zeros.

Default Values

No defaults necessary for this command.

Command Modes

(config-t1 1/1)#	T1 or DSX-1 Interface Configuration Mode required.
(config-t1 1/2)#	

Command History

Release 1.1	Command was introduced

Usage Examples

The following example activates the pattern generator for a stream of continuous ones:

(config)# **interface t1 1/1**
(config-t1 1/1)# **test-pattern ones**

DSX-1 INTERFACE CONFIGURATION COMMAND SET

To activate the DSX-1 Interface Configuration command set, enter the **interface t1** command (and specify the DSX-1 port) at the Global Configuration Mode prompt. For example:

```
Router> enable
Router# configure terminal
Router(config)# interface t1 1/2
Router(config-t1 1/2)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

alias <“text”> on page 556

description on page 557

do on page 558

end on page 559

exit on page 560

shutdown on page 561

All other commands for this command set are described in this section in alphabetical order.

coding [ami | b8zs] on page 350

framing [d4 | esf] on page 351

line-length <value> on page 352

loopback network [line | payload] on page 353

loopback remote line inband on page 354

remote-loopback on page 355

signaling-mode [message-oriented | none | robbed-bit] on page 356

snmp trap line-status on page 357

snmp trap link-status on page 358

test-pattern [ones | zeros] on page 359

coding [ami | b8zs]

Use the **coding** command to configure the line coding for a T1 or DSX-1 physical interface. This setting must match the line coding supplied on the circuit by the PBX.

Syntax Description

ami	Configures the line coding for alternate mark inversion.
b8zs	Configures the line coding for bipolar eight zero substitution.

Default Values

By default, all T1 interfaces are configured with B8ZS line coding.

Command Modes

(config-t1 1/1)#	T1 or DSX-1 Interface Configuration Mode required.
(config-t1 1/2)#	

Command History

Release 1.1	Command was introduced

Functional Notes

The line coding configured in the unit must match the line coding of the T1 circuit. A mismatch will result in line errors (e.g., BPVs).

Usage Examples

The following example configures the DSX-1 interface for AMI line coding:

(config)# **interface t1 1/2**
(config-t1 1/2)# **coding ami**

framing [d4 | esf]

Use the **framing** command to configure the framing format for the DSX-1 interface. This parameter should match the framing format set on the external device. Use the **no** form of this command to return to the default value.

Syntax Description

d4	D4 superframe format (SF)
esf	Extended superframe format

Default Values

<*format*>	**esf**

Command Modes

(config-t1 1/1)#	T1 or DSX-1 Interface Configuration Mode required.
(config-t1 1/2)#	

Command History

Release 1.1	Command was introduced

Functional Notes

A frame is comprised of a single byte from each of the T1's timeslots; there are 24 timeslots on a single T1 circuit. Framing bits are used to separate the frames and indicate the order of information arriving at the receiving equipment. D4 and ESF are two methods of collecting and organizing frames over the circuit.

Usage Examples

The following example configures the DSX-1 interface for D4 framing:

(config)# **interface t1 1/2**
(config-t1 1/2)# **framing d4**

line-length <*value*>

Use the **line-length** command to set the line build out (in feet or dB) for the DSX-1 interface. Use the **no** form of this command to return to the default value.

Syntax Description

<*value*>	Configures the line build out for the DSX-1 interface Valid options include: -7.5 dB or <0 to 655> feet

Default Values

<*value*>	0 feet

Command Modes

(config-t1 1/2)#	DSX-1 Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

The **line-length** value represents the physical distance between DSX equipment (measured in cable length). Based on this setting, the AOS device increases signal strength to compensate for the distance the signal must travel.

Valid distance ranges are listed below:

- 0-133 feet
- 134-265 feet
- 266-399 feet
- 400-533 feet
- 534-655 feet

Usage Examples

The following example configures the DSX-1 interface **line-length** for 300 feet:

(config)# **interface t1 1/2**
(config-t1 1/2)# **line-length 300**

loopback network [line | payload]

Use the **loopback network** command to initiate a loopback on the interface toward the network. Use the **no** form of this command to deactivate the loopback.

Syntax Description

line	Initiates a metallic loopback of the physical T1 network interface
payload	Initiates a loopback of the T1 framer (CSU portion) of the T1 network interface

Default Values

No default necessary for this command.

Command Modes

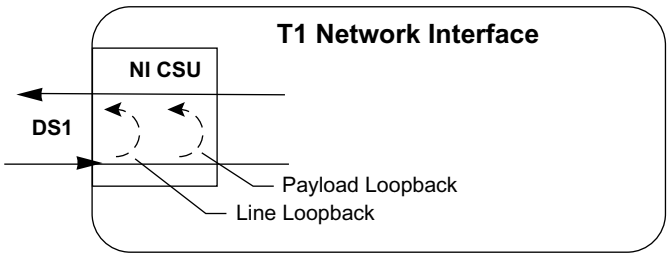
(config-t1 1/1)#	T1 or DSX-1 Interface Configuration Mode required.
(config-t1 1/2)#	

Command History

Release 1.1	Command was introduced

Functional Notes

The following diagram depicts the difference between a line and payload loopback.



Usage Examples

The following example initiates a payload loopback of the DSX-1 interface:

(config)# **interface t1 1/2**
(config-t1 1/2)# **loopback network payload**

loopback remote line inband

Use the **loopback remote line inband** command to send a loopback code to the remote unit to initiate a line loopback. Use the **no** form of this command to send a loopdown code to the remote unit to deactivate the loopback.

Syntax Description

inband	Uses the inband channel to initiate a full 1.544 Mbps physical loopback (metallic loopback) of the signal received from the network.

Default Values

No defaults necessary for this command.

Command Modes

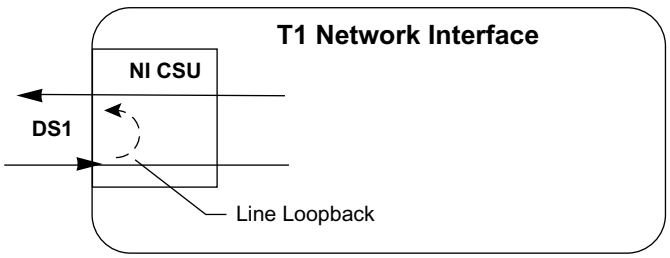
(config-t1 1/1)#	T1 or DSX-1 Interface Configuration Mode required.
(config-t1 1/2)#	

Command History

Release 1.1	Command was introduced

Functional Notes

A remote loopback can only be issued if a cross-connect does not exist on the interface and if the signaling mode is set to **none**. The following diagram depicts the difference between a line and payload loopback.



Usage Examples

The following example initiates a remote line loopback using the inband channel:

(config)# **interface t1 1/2**
(config-t1 1/2)# **loopback remote line inband**

remote-loopback

Use the **remote-loopback** command to configure the interface to respond to loopbacks initiated by a remote unit (or the service provider). Use the **no** form of this command to disable this feature.

Syntax Description

No subcommands.

Default Values

By default, all interfaces respond to remote loopbacks.

Command Modes

(config-t1 1/1)#	T1, DSX-1, or DDS Interface Configuration Mode required.
(config-t1 1/2)#	
(config-dds 1/1)#	

Command History

Release 1.1	Command was introduced

Usage Examples

The following example enables remote loopbacks on the DSX-1 interface:

(config)# **interface t1 1/2**
(config-t1 1/2)# **remote-loopback**

signaling-mode [message-oriented | none | robbed-bit]

Use the **signaling-mode** command to configure the signaling type (robbed-bit for voice or clear channel for data) for the DS0s mapped to the DSX-1 port.

Syntax Description

message-oriented	Clear channel signaling on Channel 24 only. Use this signaling type with QSIG installations.
none	Clear channel signaling on all 24 DS0s. Use this signaling type with data-only or PRI DSX-1 installations.
robbed-bit	Robbed bit signaling on all DS0s. Use this signaling type for voice-only DSX-1 applications.

Default Values

By default, the signaling mode is set to robbed-bit.

Command Modes

(config-t1 1/2)#	DSX-1 Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example configures the DSX-1 port for PRI compatibility:

(config)# **interface t1 1/2**
(config-t1 1/2)# **signaling-mode none**

snmp trap line-status

Use the **snmp trap line-status** command to control the Simple Network Management Protocol (SNMP) variable dsx1LineStatusChangeTrapEnable (RFC 2495) to enable (or disable) the interface to send SNMP traps when there is an interface status change. The dsx1LineStatusChangeTrapEnable variable is set to enabled by default.

Syntax Description

No subcommands

Default Values

By default, the dsx1LineStatusChangeTrapEnable OID is set to enabled for all interfaces except virtual frame relay interfaces.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: T1 (t1 1/1), DSX-1 (t1 1/2), serial (ser 1/1), DDS (dds 1/1), virtual frame relay interfaces (fr 1), and virtual PPP interfaces (ppp 1).

Command History

Release 1.1	Command was introduced

Functional Notes

The **snmp trap line-status** command is used to control the RFC 2495 dsx1LineStatusChangeTrapEnable OID (OID number 1.3.6.1.2.1.10.18.6.1.17.0).

Usage Examples

The following example disables the line-status trap on the DSX-1 interface:

(config)# **interface t1 1/2**
(config-t1 1/2)# **no snmp trap line-status**

snmp trap link-status

Use the **snmp trap link-status** to control the SNMP variable ifLinkUpDownTrapEnable (RFC 2863) to enable (or disable) the interface to send SNMP traps when there is an interface status change. Use the **no** form of this command to disable this trap.

Syntax Description

No subcommands

Default Values

By default, the ifLinkUpDownTrapEnable OID is set to enabled for all interfaces except virtual frame relay interfaces.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), T1 (t1 1/1), DSX-1 (t1 1/2), serial (ser 1/1), DDS (dds 1/1), virtual frame relay (fr 1), virtual PPP (ppp 1), and SHDSL (shdsl 1/1) interfaces.

Command History

Release 1.1	Command was introduced

Functional Notes

The **snmp trap link-status** command is used to control the RFC 2863 ifLinkUpDownTrapEnable OID (OID number 1.3.6.1.2.1.31.1.1.1.14.0).

Usage Examples

The following example disables the link-status trap on the DSX-1 interface:

(config)# **interface t1 1/2**
(config-t1 1/2)# **no snmp trap link-status**

test-pattern [ones | zeros]

Use the **test-pattern** command to activate the built-in pattern generator and begin sending the specified test pattern. This pattern generation can be used to verify a data path when used in conjunction with an active loopback. Use the **no** form of this command to cease pattern generation.

Syntax Description

ones	Generate continous ones
zeros	Generate continous zeros

Default Values

No defaults necessary for this command.

Command Modes

(config-t1 1/1)#	T1 or DSX-1 Interface Configuration Mode required.
(config-t1 1/2)#	

Command History

Release 1.1	Command was introduced

Usage Examples

The following example activates the pattern generator for a stream of continuous ones:

(config)# **interface t1 1/2**
(config-t1 1/2)# **test-pattern ones**

SERIAL INTERFACE CONFIGURATION COMMAND SET

To activate the Serial Interface Configuration command set, enter the **interface serial** command at the Global Configuration Mode prompt. For example:

```
Router> enable
Router# configure terminal
Router(config)# interface serial 1/1
Router(config-ser 1/1)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

alias <"text"> on page 556

description on page 557

do on page 558

end on page 559

exit on page 560

shutdown on page 561

All other commands for this command set are described in this section in alphabetical order.

et-clock-source <source> on page 361

ignore dcd on page 362

invert etclock on page 363

invert rxclock on page 364

invert txclock on page 365

serial-mode <mode> on page 366

snmp trap on page 367

snmp trap link-status on page 368

et-clock-source *<source>*

Use the **et-clock-source** command to configure the clock source used when creating the external transmit (reference clock). Use the **no** form of this command to return to the default value.

Syntax Description

<source>	Specifies the signal source to use when creating the External Transmit reference clock (et-clock).
rxclock	Use the clock recovered from the receive signal to generate et-clock.
txclock	Use the clock recovered from the transmit signal to generate et-clock.

Default Values

<source>	**txclock**

Command Modes

(config-ser 1/1)#	Serial Interface Configuration Mode

Command History

Release 1.1	Command was introduced

Functional Notes

External Transmit clock (et-clock) is an interface timing signal (provided by the DTE device) used to synchronize the transfer of transmit data.

Usage Examples

The following example configures the serial interface to recover the clock signal from the received signal and use it to generate et-clock:

(config)# **interface serial 1/1**
(config-ser 1/1)# **et-clock-source rxclock**

ignore dcd

Use the **ignore dcd** command to specify the behavior of the serial interface when the Data Carrier Detect (DCD) signal is lost. Use the **no** form of this command to return to the default value.

Syntax Description

No subcommands.

Default Values

By default, the serial interface does not ignore a change in status of the DCD signal.

Command Modes

(config-ser 1/1)#	Serial Interface Configuration Mode

Command History

Release 1.1	Command was introduced

Functional Notes

When configured to follow DCD (default condition), the serial interface will not attempt to establish a connection when DCD is not present. When configured to ignore DCD, the serial interface will continue to attempt to establish a connection even when DCD is not present.

Usage Examples

The following example configures the serial interface to ignore a loss of the DCD signal:

(config)# **interface serial 1/1**
(config-ser 1/1)# **ignore dcd**

invert etclock

Use the **invert etclock** command to configure the serial interface to invert the External Transmit (reference clock) in the data stream before transmitting. Use the **no** form of this command to return to the default value.

Syntax Description

No subcommands.

Default Values

By default, the serial interface does not invert etclock.

Command Modes

(config-ser 1/1)#	Serial Interface Configuration Mode

Command History

Release 1.1	Command was introduced

Functional Notes

If the serial interface cable is long, causing a phase shift in the data, the et clock can be inverted using the **invert etclock** command. This switches the phase of the clock, which compensates for a long cable.

Usage Examples

The following example configures the serial interface to invert etclock:

(config)# **interface serial 1/1**
(config-ser 1/1)# **invert etclock**

invert rxclock

Use the **invert rxclock** command to configure the serial interface to expect an inverted Receive Clock (found in the received data stream). Use the **no** form of this command to return to the default value.

Syntax Description

No subcommands.

Default Values

By default, the serial interface does not expect an inverted receive clock (rxclock).

Command Modes

(config-ser 1/1)#	Serial Interface Configuration Mode

Command History

Release 1.1	Command was introduced

Functional Notes

If the serial interface cable is long, causing a phase shift in the data, the transmit clock can be inverted. This switches the phase of the clock, which compensates for a long cable. If the transmit clock of the connected device is inverted, use the **invert rxclock** command to configure the receiving interface appropriately.

Usage Examples

The following example configures the serial interface to invert receive clock:

(config)# **interface serial 1/1**
(config-ser 1/1)# **invert rxclock**

invert txclock

Use the **invert txclock** command to configure the serial interface to invert the Transmit Clock (found in the transmitted data stream) before sending the signal. Use the **no** form of this command to return to the default value.

Syntax Description

No subcommands.

Default Values

By default, the serial interface does not invert txclock.

Command Modes

(config-ser 1/1)#	Serial Interface Configuration Mode

Command History

Release 1.1	Command was introduced

Functional Notes

If the serial interface cable is long, causing a phase shift in the data, the transmit clock can be inverted (using the **invert txclock** command). This switches the phase of the clock, which compensates for a long cable. If the transmit clock of the connected device is inverted, use the **invert rxclock** command to configure the receiving interface appropriately.

Usage Examples

The following example configures the serial interface to invert the transmit clock:

(config)# **interface serial 1/1**
(config-ser 1/1)# **invert txclock**

serial-mode <*mode*>

Use the **serial-mode** command to specify the electrical mode for the interface. Use the **no** form of this command to return to the default value.

Syntax Description

<*mode*>	Specifies the electrical specifications for the interface
V35	Configures the interface for use with the V.35 adapter cable (P/N 1200873L1)
X21	Configures the interface for use with the X.21 adapter cable (P/N 1200874L1)

Default Values

<*mode*>	**V35**

Command Modes

(config-ser 1/1)#	Serial Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

The pinouts for each of the available interfaces may be found in the Hardware Configuration Guide located on the ***NetVanta 3000 Series System Manual*** CD (provided in shipment).

Usage Examples

The following example configures the serial interface to work with the X.21 adapter cable:

(config)# **interface serial 1/1**
(config-ser 1/1)# **serial-mode X21**

snmp trap

Use the **snmp trap** command to enable all supported Simple Network Management Protocol (SNMP) traps on the interface.

Syntax Description

No subcommands

Default Values

By default, all interfaces (except virtual frame relay interfaces and sub-interfaces) have SNMP traps enabled.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), DDS (dds 1/1), serial (ser 1/1), virtual frame relay (fr 1), and SHDSL (shdsl 1/1) interfaces.

Command History

Release 1.1	Command was introduced

Usage Examples

The following example enables SNMP on the serial interface:

(config)# **interface serial 1/1**
(config-ser 1/1)# **snmp trap**

snmp trap link-status

Use the **snmp trap link-status** command to control the SNMP variable to enable (or disable) the interface to send SNMP traps when there is an interface status change (ifLinkUpDownTrapEnable per RFC 2863). Use the **no** form of this command to disable this trap.

Syntax Description

No subcommands

Default Values

By default, the ifLinkUpDownTrapEnable OID is set to enabled for all interfaces except virtual frame relay interfaces.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), T1 (t1 1/1), DSX-1 (t1 1/2), serial (ser 1/1), DDS (dds 1/1), virtual frame relay (fr 1), virtual PPP (ppp 1), and SHDSL (shdsl 1/1) interfaces.

Command History

Release 1.1	Command was introduced

Functional Notes

The **snmp trap link-status** command is used to control the RFC 2863 ifLinkUpDownTrapEnable OID (OID number 1.3.6.1.2.1.31.1.1.1.14.0).

Usage Examples

The following example disables the link-status trap on the serial interface:

(config)# **interface serial 1/1**
(config-ser 1/1)# **no snmp trap link-status**

SHDSL INTERFACE CONFIGURATION COMMAND SET

To activate the SHDSL Interface Configuration command set, enter the **interface shdsl** command at the Global Configuration Mode prompt. For example:

```
Router> enable
Router# configure terminal
Router(config)# interface shdsl 1/1
Router(config-shdsl 1/1)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

alias <"text"> on page 556

description on page 557

do on page 558

end on page 559

exit on page 560

shutdown on page 561

All other commands for this command set are described in this section in alphabetical order.

alarm-threshold [loop-attenuation | snr-margin] on page 370

boot alternate-image on page 371

equipment-type [co | cpe] on page 372

inband-detection on page 373

inband-protocol on page 374

linerate <value> on page 375

loopback network on page 376

loopback remote on page 377

outage-retrain on page 378

snmp trap on page 379

snmp trap link-status on page 380

test-pattern on page 381

alarm-threshold [loop-attenuation | snr-margin]

Use the **alarm-threshold** command to set thresholds for specific alarm conditions. Use the **no** form of this command to disable threshold settings.

Syntax Description

loop-attenuation <*value*>	Enter a value from 1-127 dB. If signal energy loss on the loop exceeds the configured value, the router issues an alarm.
snr-margin <*value*>	Signal-to-noise ratio margin. Enter a value from 1-15 dB. If the difference in amplitude between the baseband signal and the noise exceeds the configured value, the router issues an alarm.

Default Values

No defaults necessary for this command.

Command Modes

(config-shdsl 1/1)#	SHDSL Interface Configuration Mode required

Command History

Release 3.1	Command was introduced

Usage Examples

The following example sets the loop attenuation threshold at 45 dB:

(config)# **interface shdsl 1/1**
(config-shdsl 1/1)# **alarm-threshold loop-attenuation 45**

boot alternate-image

Use the **boot alternate-image** command to execute new code after a firmware upgrade.

Syntax Description

No subcommands.

Default Values

No defaults necessary for this command.

Command Modes

(config-shdsl 1/1)#	SHDSL Interface Configuration Mode required

Command History

Release 3.1	Command was introduced

Functional Notes

The current SHDSL NIM card (1200867L1) supports two code images commonly referred to as the "active" image and the "inactive" image. When a firmware upgrade is performed on the card (through the **copy <filename> interface shdsl x/y** command), the new firmware is placed in the "inactive" image space. This new code will not be executed until the **boot alternate-image** command is issued. When the user does this, the NIM will reboot (taking the current line down) with the new code. At this point, the old code becomes the "inactive" image and the new recently updated code becomes the "active" image.

Usage Examples

The following example causes the firmware upgrade to take effect:

(config)# **interface shdsl 1/1**
(config-shdsl 1/1)# **boot alternate-image**

equipment-type [co | cpe]

Use the **equipment-type** command to determine the operating mode for the SHDSL interface.

Syntax Description

co	Use this option only in a campus environment when operating two SHDSL network interface modules (NIMs) back-to-back. In this setup, configure the Master NIM to **co** and the Slave NIM to **cpe**.
cpe	Use this option when interfacing directly with your service provider or when acting as the Slave NIM in a campus environment.

Default Values

*The default for this command is **cpe**.*

Command Modes

(config-shdsl 1/1)#	SHDSL Interface Configuration Mode required

Command History

Release 3.1	Command was introduced

Usage Examples

The following example changes the operating mode of the SHDSL interface to CO:

(config)# **interface shdsl 1/1**
(config-shdsl 1/1)# **equipment-type co**

inband-detection

Use the **inband-detection** enable inband loopback pattern detection on the SHDSL interface. Use the **no** form of this command to disable **inband-detection**.

Syntax Description

No subcommands

Default Values

By default, this command is enabled.

Command Modes

(config-shdsl 1/1)#	SHDSL Interface Configuration Mode required

Command History

Release 4.1	Command was introduced

Usage Examples

The following example disables loopback pattern detection:

(config-shdsl 1/1)# **no inband-detection**

inband-protocol

Use the **inband-protocol** command to designate the inband loopback pattern to send/detect on the SHDSL interface. Use the **no** form of this command to reset the **inband-protocol** to its default.

Syntax Description

pn127	Selects PN127 as the inband loopback pattern to send/detect.
v54	Selects V.54 as the inband loopback pattern to send/detect.

Default Values

*By default, the **inband-protocol** is set to **v54**.*

Command Modes

(config-shdsl 1/1)#	SHDSL Interface Configuration Mode required

Command History

Release 4.1	Command was introduced

Functional Notes

Inband loopbacks are specific patterns that are sent in place of user data to trigger a loopback. Both PN127 and V.54 are industry-standard loopback patterns used to allow remote loopbacks.

Usage Examples

The following example sets the inband loopback pattern for PN127:

(config-shdsl 1/1)# **inband-protocol pn127**

linerate <*value*>

Use the **linerate** command to define the line rate for the SHDSL interface (the value includes 8 kbps of framing overhead). This command is functional only in CO operating mode (see the section *equipment-type [co | cpe] on page 372*). The first two selections listed (72 and 136 kbps) are not supported by the SHDSL NIM (1200867L1).

Syntax Description

<*value*>	Enter the line rate in kbps. Range: 200 to 2312 kbps in 64k increments.

Default Values

The default for this command is 2056 kbps.

Command Modes

(config-shdsl 1/1)#	SHDSL Interface Configuration Mode required

Command History

Release 3.1	Command was introduced

Usage Examples

The following example changes the line rate of the SHDSL interface to 264 kbps:

(config)# **interface shdsl 1/1**
(config-shdsl 1/1)# **linerate 264**

loopback network

Use the **loopback network** command to initiate a loopback test on the SHDSL interface, looping the data toward the network. Use the **no** form of this command to deactivate the loopback.

Syntax Description

No subcommands.

Default Values

No default necessary for this command.

Command Modes

(config-shdsl 1/1)#	SHDSL Interface Configuration Mode required.

Command History

Release 3.1	Command was introduced

Usage Examples

The following example initiates a loopback on the SHDSL interface:

```
(config)# interface shdsl 1/1
(config-shdsl 1/1)# loopback network
```


loopback remote

Use the **loopback remote** command to send a loopback request to the remote unit. This command is functional only in CO operating mode (see the section *equipment-type [co | cpe] on page 372*). Use the **no** form of this command to send a loopdown code to the remote unit to deactivate the loopback.

Syntax Description

No subcommands.

Default Values

No defaults necessary for this command.

Command Modes

(config-shdsl 1/1)#	SHDSL Interface Configuration Mode required.

Command History

Release 3.1	Command was introduced

Usage Examples

The following example initiates a remote line loopback:

(config)# **interface shdsl 1/1**
(config-shdsl 1/1)# **loopback remote**

outage-retrain

Use the **outage-retrain** command to cause the SHDSL interface to force the SHDSL retrain sequence (which takes the line down temporarily) if the interface detects more than ten consecutive errored seconds. A retrain is forced in hopes that the newly retrained line will attain better performance that the previous training state. Use the **no** version of the command to disable this feature.

Syntax Description

No subcommands.

Default Values

By default, this feature is disabled.

Command Modes

(config-shdsl 1/1)#	SHDSL Interface Configuration Mode required

Command History

Release 3.1	Command was introduced

Usage Examples

The following example forces a retrain sequence on the SHDSL interface:

(config)# **interface shdsl 1/1**
(config-shdsl 1/1)# **outage-retrain**

snmp trap

Use the **snmp trap** command to enable all supported Simple Network Management Protocol (SNMP) traps for the SHDSL interface.

Syntax Description

No subcommands

Default Values

By default, all interfaces (except virtual frame relay interfaces and sub-interfaces) have SNMP traps enabled.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), DDS (dds 1/1), serial (ser 1/1), virtual frame relay (fr 1), and SHDSL (shdsl 1/1) interfaces.

Command History

Release 1.1	Command was introduced
Release 3.1	Command was extended to the SHDSL interface

Usage Examples

The following example enables SNMP on the SHDSL interface:

(config)# **interface shdsl 1/1**
(config-shdsl 1/1)# **snmp trap**

snmp trap link-status

Use the **snmp trap link-status** command to control the SNMP variable ifLinkUpDownTrapEnable (RFC 2863), which enables (or disables) the interface to send SNMP traps when there is an interface status change. Use the **no** form of this command to disable this trap.

Syntax Description

No subcommands

Default Values

By default, the ifLinkUpDownTrapEnable OID is set to enabled for all interfaces (except virtual frame relay interfaces).

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), T1 (t1 1/1), DSX-1 (t1 1/2), serial (ser 1/1), DDS (dds 1/1), virtual frame relay (fr 1), virtual PPP (ppp 1), and SHDSL (shdsl 1/1) interfaces.

Command History

Release 1.1	Command was introduced
Release 3.1	Command was extended to the SHDSL interface

Functional Notes

The **snmp trap link-status** command is used to control the RFC 2863 ifLinkUpDownTrapEnable OID (OID number 1.3.6.1.2.1.31.1.1.1.14.0).

Usage Examples

The following example disables the link-status trap on the SHDSL interface:

(config)# **interface shdsl 1/1**
(config-shdsl 1/1)# **no snmp trap link-status**

test-pattern

Use the **test-pattern** command to activate the built-in pattern generator and begin sending the 2^15 test pattern toward the network. This pattern generation can be used to verify a data path when used in conjunction with an active loopback. Use the **no** form of this command to cease pattern generation.

Syntax Description

2^15	Sends a 2^15 test pattern toward the network.
clear	Clears the test pattern results on the SHDSL interface.
insert	Inserts an error into the generated test pattern being transmitted on the SHDSL interface. The injected error result is displayed using the **test-pattern show** command.
show	Displays the results of the test pattern.

Default Values

No defaults necessary for this command.

Command Modes

(config-shdsl 1/1)#	SHDSL Interface Configuration Mode required.

Command History

Release 3.1	Command was introduced

Usage Examples

The following example sends a 2^15 test pattern:

(config)# **interface shdsl 1/1**
(config-shdsl 1/1)# **test-pattern 2^15**

MODEM INTERFACE CONFIGURATION COMMAND SET

To activate the Modem Interface Configuration command set, enter the **interface modem** command at the Global Configuration Mode prompt. For example:

```
Router> enable
Router# configure terminal
Router(config)# interface modem 1/2
Router(config-modem 1/2)#
```

*The modem interface number in the example above is shown as **modem 1/2**. This number is based on the interface's location (slot/port) and could vary depending on the unit's configuration. Use the **do show interfaces** command to determine the appropriate interface number.*

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

alias <"*text*"> on page 556

description on page 557

do on page 558

end on page 559

exit on page 560

shutdown on page 561

All other commands for this command set are described in this section in alphabetical order.

dialin on page 383

iq-handshake on page 384

dialin

Use the **dialin** command to enable the modem for remote console dialin, disabling the use of the modem for dial-backup.

Syntax Description

No subcommands.

Default Values

By default, dialin is disabled.

Command Modes

(config-modem 1/2)#	Modem Interface Configuration Mode

Command History

Release 3.1	Command was introduced

Usage Examples

The following example enables remote console dialin:

(config)# **interface modem 1/2**
(config-modem 1/2)# **dialin**

iq-handshake

Use the **iq-handshake** command for dial-backup with the ADTRAN IQ and Express family products. Using the **iq-handshake** command enables the association of incoming calls with packet endpoints (in cases where there is a single call-in number (hunt group) and no Caller ID information available). Use the **no** form of this command to disable the IQ handshake.

Syntax Description

No subcommands

Default Values

*By default, the **iq-handshake** feature is disabled.*

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: BRI (bri 1/2), modem (modem 1/2 modem 1/3), virtual frame relay sub-interfaces (fr 1.6), and virtual PPP interfaces (ppp 1).

Command History

Release 2.1	Command was introduced

Functional Notes

The ADTRAN IQ and Express products implement a proprietary dial-backup mechanism to transfer configuration information (such as DLCIs) when entering dial-backup. The **iq-handshake** command enables this proprietary data transfer in the ADTRAN OS.

Usage Examples

The following example enables the proprietary data transfer (to an IQ or Express product) on the modem dial-backup interface:

(config)# **interface modem 1/2**
(config-modem 1/2)# **iq-handshake**

BRI INTERFACE CONFIGURATION COMMAND SET

To activate the BRI Interface Configuration command set, enter the **interface bri** command at the Global Configuration Mode prompt. For example:

```
Router> enable
Router# configure terminal
Router(config)# interface bri 1/2
Router(config-bri 1/2)#
```

*The BRI interface number in the example above is shown as **bri 1/2**. This number is based on the interface's location (slot/port) and could vary depending on the unit's configuration. Use the **do show interfaces** command to determine the appropriate interface number.*

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

alias <"text"> on page 556

description on page 557

do on page 558

end on page 559

exit on page 560

shutdown on page 561

All other commands for this command set are described in this section in alphabetical order.

bonding txadd-timer <seconds> on page 386

bonding txcid-timer <seconds> on page 387

bonding txdeq-timer <seconds> on page 388

bonding txfa-timer <seconds> on page 389

bonding txinit-timer <seconds> on page 390

bonding txnull-timer <seconds> on page 391

iq-handshake on page 392

isdn spid1 <spid> <ldn> on page 393

isdn spid2 <spid> <ldn> on page 395

isdn switch-type <type> on page 397

bonding txadd-timer *<seconds>*

Use the **bonding txadd-timer** command to specify the value (in seconds) for the aggregate call connect timeout. Use the **no** form of this command to return to the default value.

Syntax Description

<seconds>	Specifies the number of seconds the endpoint will wait for additional channels (to add to the bonded aggregate) before considering the BONDING negotiation a failure

Default Values

<seconds>	**50** seconds

Command Modes

(config-bri 1/2)#	BRI Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

Specifies the length of time both endpoints wait for additional calls to be connected at the end of negotiation before deciding that the BONDING call has failed. The factory default setting is sufficient for most calls to connect, although when dialing overseas it may be necessary to lengthen this timer to allow for slower call routing.

Usage Examples

The following example defines a txadd-timer value of 95 seconds:

(config)# **interface bri 1/2**
(config-bri 1/2)# **bonding txadd-timer 95**

bonding txcid-timer <*seconds*>

Use the **bonding txcid-timer** command to specify the value (in seconds) for the bearer channel (B-channel) negotiation timeout. Use the **no** form of this command to return to the default value.

Syntax Description

<*seconds*>	Specifies the number of seconds the endpoint allots for negotiating data rates and channel capacities before considering the BONDING negotiation a failure

Default Values

<*seconds*>	**5** seconds

Command Modes

(config-bri 1/2)#	BRI Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

Specifies the length of time both endpoints attempt to negotiate an agreeable value for bearer channels and channel capacities before deciding the BONDING call has failed.

Usage Examples

The following example defines a txcid-timer value of 8 seconds:

(config)# **interface bri 1/2**
(config-bri 1/2)# **bonding txcid-timer 8**

bonding txdeq-timer *<seconds>*

Use the **bonding txdeq-timer** command to specify the value (in seconds) for the network delay equalization timeout. Use the **no** form of this command to return to the default value.

Syntax Description

<seconds>	Specifies the number of seconds the endpoint allots for attempting to equalize the network delay between bearer channels before considering the BONDING negotiation a failure

Default Values

<seconds>	**50** seconds

Command Modes

(config-bri 1/2)#	BRI Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

Specifies the length of time both endpoints allot to attempt to equalize the network delay between the bearer channels before deciding the BONDING call has failed.

Usage Examples

The following example defines a txdeq-timer value of 80 seconds:

(config)# **interface bri 1/2**
(config-bri 1/2)# **bonding txdeq-timer 80**

bonding txfa-timer *<seconds>*

Use the **bonding txfa-timer** command to specify the value (in seconds) for the frame pattern detection timeout. Use the **no** form of this command to return to the default value.

Syntax Description

<seconds>	Specifies the number of seconds the endpoint allots for attempting to detect the BONDING frame pattern (when a call is connected) before considering the BONDING negotiation a failure

Default Values

<seconds>	**10** seconds

Command Modes

(config-bri 1/2)#	BRI Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

Specifies the length of time both endpoints attempt to detect the BONDING frame pattern when a call is connected before deciding the BONDING call has failed. When operating with other manufacturers' BONDING equipment, it may be necessary to change this time so that it matches TXADD01.

Usage Examples

The following example defines a txfa-timer value of 15 seconds:

(config)# **interface bri 1/2**
(config-bri 1/2)# **bonding txfa-timer 15**

bonding txinit-timer <*seconds*>

Use the **bonding txinit-timer** command to specify the value (in seconds) for the originating endpoint negotiation timeout. Use the **no** form of this command to return to the default value.

Syntax Description

<*seconds*>	Specifies the number of seconds the endpoint waits to detect the BONDING negotiation frame pattern from the remote endpoint (when a call is connected) before considering the BONDING negotiation a failure

Default Values

<*seconds*>	**10** seconds

Command Modes

(config-bri 1/2)#	BRI Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

Specifies the length of time the originating endpoint attempts to detect the BONDING negotiation pattern from the answering endpoint before deciding the BONDING call has failed.

Usage Examples

The following example defines a txinit-timer value of 15 seconds:

(config)# **interface bri 1/2**
(config-bri 1/2)# **bonding txinit-timer 15**

bonding txnull-timer *<seconds>*

Use the **bonding txnull-timer** command to specify the value (in seconds) for the answering endpoint negotiation timeout. Use the **no** form of this command to return to the default value.

Syntax Description

<seconds>	Specifies the number of seconds the endpoint waits to detect the BONDING negotiation frame pattern from the originating endpoint (after answering a call) before considering the BONDING negotiation a failure

Default Values

<seconds>	**10** seconds

Command Modes

(config-bri 1/2)#	BRI Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

Specifies the length of time the answering endpoint attempts to detect the BONDING negotiation pattern from the originating endpoint before deciding the BONDING call has failed. It may be necessary to shorten this timer if the DTE equipment using the BONDING module also has timer constraints for completing non-BONDING parameter negotiation.

Usage Examples

The following example defines a txnull-timer value of 8 seconds:

(config)# **interface bri 1/2**
(config-bri 1/2)# **bonding txnull-timer 8**

iq-handshake

Use the **iq-handshake** command for dial-backup operation with ADTRAN IQ and Express family products. Using the **iq-handshake** command enables the association of incoming calls with packet endpoints (in cases where there is a single call-in number (hunt group) and no Caller ID information available). Use the **no** form of this command to disable the IQ handshake.

Syntax Description

No subcommands

Default Values

*By default, the **iq-handshake** feature is disabled.*

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: BRI (bri 1/2), Modem (modem 1/2 modem 1/3), virtual frame relay sub-interfaces (fr 1.6), and virtual PPP interfaces (ppp 1).

Command History

Release 2.1	Command was introduced

Functional Notes

The ADTRAN IQ and Express products implement a proprietary dial-backup mechanism to transfer configuration information (such as DLCIs) when entering dial-backup mode. The **iq-handshake** command enables this proprietary data transfer in the ADTRAN OS.

Usage Examples

The following example enables the proprietary data transfer (to an IQ or Express product) on the BRI dial-backup interface:

(config)# **interface bri 1/2**
(config-bri 1/2)# **iq-handshake**

isdn spid1 <*spid*> <*ldn*>

Use the **isdn spid1** command to specify the Service Profile Identifiers (SPIDs). Use the **no** form of this command to remove a configured SPID.

*The BRI Module requires all incoming calls to be directed to the Local Directory Number (LDN) associated with the SPID programmed using the **isdn spid1** command. All calls to the LDN associated with SPID 2 will be rejected (unless part of a BONDing call).*

Syntax Description

<*spid*>	Specifies the 8 to 14 digit number identifying your Basic Rate ISDN (BRI) line in the Central Office Switch. A SPID is generally created using the area code and phone number associated with the line and a four-digit suffix. For example, the following SPIDs may be provided on a BRI line with phone numbers 555-1111 and 555-1112: SPID1: 701 555 1111 0101 SPID2: 701 555 1112 0101
<*ldn*> ***Optional**	Local Directory Number (LDN) assigned to the circuit by the service provider. The LDN is the number used by remote callers to dial into the ISDN circuit. If the <*ldn*> field is left blank, the ADTRAN OS will not accept inbound dial-backup calls to the BRI module.

Default Values

By default, there are no configured SPIDs

Command Modes

(config-bri 1/2)#	BRI Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

The ADTRAN OS does not support “spid-less” 5ESS signaling. SPIDs are required for all configured BRI endpoints.

Usage Examples

The following example defines a SPID of 704 555 1111 0101 with an LDN of 555-1111:

(config)# **interface bri 1/2**
(config-bri 1/2)# **isdn spid1 70455511110101 5551111**

isdn spid2 <*spid*> <*ldn*>

Use the **isdn spid2** command to specify the Service Profile Identifiers (SPIDs). Use the **no** form of this command to remove a configured SPID.

> NOTE
>
> *The BRI Module requires all incoming calls to be directed to the Local Directory Number (LDN) associated with the SPID programmed using the **isdn spid1** command. All calls to the LDN associated with SPID 2 will be rejected (unless part of a BONDing call).*

Syntax Description

<*spid*>	Specifies the 8 to 14 digit number identifying your Basic Rate ISDN (BRI) line in the Central Office Switch. A SPID is generally created using the area code and phone number associated with the line and a four-digit suffix. For example, the following SPIDs may be provided on a BRI line with phone numbers 555-1111 and 555-1112: SPID1: 701 555 1111 0101 SPID2: 701 555 1112 0101
<*ldn*> *****Optional	Local Directory Number (LDN) assigned to the circuit by the service provider. The LDN is the number used by remote callers to dial into the ISDN circuit. If the <*ldn*> field is left blank, the ADTRAN OS will not accept inbound dial-backup calls to the BRI module.

Default Values

By default, there are no configured SPIDs

Command Modes

(config-bri 1/2)#	BRI Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

The ADTRAN OS does not support “spid-less” 5ESS signaling. SPIDs are required for all configured BRI endpoints.

Usage Examples

The following example defines a SPID of 704 555 1111 0101:

(config)# **interface bri 1/2**
(config-bri 1/2)# **isdn spid2 70455511110101 5551111**

isdn switch-type <*type*>

Use the **isdn switch-type** command to specify the ISDN signaling type configured on the Basic Rate ISDN (BRI) interface. The type of ISDN signaling implemented on the BRI interface does not always match the manufacturer of the Central Office Switch. Use the **no** form of this command to return to the default value.

Syntax Description

<*type*>	Specifies the signaling type on the BRI interface (configured by the service provider on the Central Office Switch)
basic-5ess	Specifies Lucent/AT&T 5ESS signaling on the BRI interface
basic-dms	Specifies Nortel DMS-100 custom signaling on the BRI interface The **basic-dms** signaling type is not compatible with proprietary SL-1 DMS signaling.
basic-ni	Specifies National ISDN-1 signaling on the BRI interface

Default Values

<*type*>	**basic-ni**

Command Modes

(config-bri 1/2)#	BRI Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

The **isdn switch-type** command specifies the type of ISDN signaling implemented on the BRI interface, not the manufacturer of the Central Office Switch. It is quite possible to have a Lucent Central Office Switch providing National ISDN signaling on the BRI interface.

Usage Examples

The following example configures a BRI interface for a circuit with Lucent 5ESS (custom) signaling:

(config)# **interface bri 1/2**
(config-bri 1/2)# **isdn switch-type basic-5ess**

FRAME RELAY INTERFACE CONFIG COMMAND SET

To activate the Frame Relay Interface Configuration command set, enter the **interface frame-relay** command at the Global Configuration Mode prompt. For example:

```
Router> enable
Router# configure terminal
Router(config)# interface frame-relay 1
Router(config-fr 1)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

All other commands for this command set are described in this section in alphabetical order.

encapsulation frame-relay ietf

Use the **encapsulation frame-relay ietf** command to configure the encapsulation on a virtual frame relay interface as IETF (RFC 1490). Currently, this is the only encapsulation setting. Settings for this option must match the far-end router's settings in order for the frame relay interface to become active.

Syntax Description

No subcommands.

Default Values

By default, all frame relay interfaces use IETF encapsulation.

Command Modes

(config-fr 1)#	Virtual Frame Relay Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example configures the endpoint for IETF encapsulation:

(config)# **interface frame-relay 1**
(config-fr 1)# **encapsulation frame-relay ietf**

frame-relay intf-type *<type>*

Use the **frame-relay intf-type** command to define the frame relay signaling role needed for the endpoint. Use the **no** form of this command to return to the default value.

Syntax Description

<type>	Specifies the frame relay interface types as DTE, DCE, or NNI
dce	DCE or Network signaling role. Use this interface type when you need the unit to emulate the frame switch.
dte	DTE or User signaling role. Use this interface type when connecting to a frame relay switch (or piece of equipment emulating a frame switch).
nni	Configures the interface to support both network and user signaling (DTE or DCE) when necessary.

Default Values

<type>	**dte**

Command Modes

(config-fr 1)#	Virtual Frame Relay Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example configures the frame relay endpoint for DCE signaling:

(config)# **interface frame-relay 1**
(config-fr 1)# **frame-relay intf-type dce**

frame-relay lmi-n391dce *<seconds>*

Use the **frame-relay lmi-n391dce** command to set the n391 full status polling counter for the DCE endpoint. Typical applications should leave the default value for this timer. Use the **no** form of this command to return to the default value.

Syntax Description

<seconds>	Sets the timer value in seconds (valid range: 1-255)

Default Values

<seconds>	**6 seconds**

Command Modes

(config-fr 1)#	Virtual Frame Relay Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

The N391 timer determines how many link integrity polls occur in between full status polls.

Usage Examples

The following example sets the N391 timer for 20 seconds:

(config)# **interface frame-relay 1**
(config-fr 1)# **frame-relay lmi-n391dce 20**

frame-relay lmi-n391dte <*seconds*>

Use the **frame-relay lmi-n391dte** command to set the N391 full status polling counter for the DTE endpoint. Typical applications should leave the default value for this timer. Use the **no** form of this command to return to the default value.

Syntax Description

<*seconds*>	Sets the timer value in seconds (valid range: 1-255)

Default Values

<*seconds*>	**6 seconds**

Command Modes

(config-fr 1)#	Virtual Frame Relay Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

The N391 timer determines how many link integrity polls occur in between full status polls.

Usage Examples

The following example sets the N391 timer for 20 seconds:

(config)# **interface frame-relay 1**
(config-fr 1)# **frame-relay lmi-n391dte 20**

frame-relay lmi-n392dce *<threshold>*

Use the **frame-relay lmi-n392dce** command to set the N392 error threshold for the DCE endpoint. Typical applications should leave the default value for this setting. Use the **no** form of this command to return to the default value.

Syntax Description

<threshold>	Sets the threshold value (valid range: 1-10)

Default Values

<threshold>	**3 errors**

Command Modes

(config-fr 1)#	Virtual Frame Relay Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example sets the N392 timer for 5 seconds:

(config)# **interface frame-relay 1**
(config-fr 1)# **frame-relay lmi-n392dce 5**

frame-relay lmi-n392dte <*threshold*>

Use the **frame-relay lmi-n392dte** command to set the N392 error threshold for the DTE endpoint. Typical applications should leave the default value for this setting. Use the **no** form of this command to return to the default value.

Syntax Description

<*threshold*>	Sets the threshold value (valid range: 1-10)

Default Values

<*threshold*>	**3 errors**

Command Modes

(config-fr 1)#	Virtual Frame Relay Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

If the error threshold is met, the signaling state status is changed to down, which indicates a service-affecting condition. This condition is cleared once N393 consecutive error-free events are received. N392 defines the number of errors required in a given event window, while N393 defines the number of polling events in each window.

For example:
If N392=3 and N393=4, then if three errors occur within any four events, the interface is determined inactive.

Usage Examples

The following example sets the N392 threshold for 5 errors:

(config)# **interface frame-relay 1**
(config-fr 1)# **frame-relay lmi-n392dte 5**

frame-relay lmi-n393dce *<counter>*

Use the **frame-relay lmi-n393dce** to set the N393 LMI monitored event counter for the DCE endpoint. Typical applications should leave the default value for this counter. Use the **no** form of this command to return to the default value.

Syntax Description

<counter>	Sets the counter value (valid range: 1-10)

Default Values

<counter>	**4 events**

Command Modes

(config-fr 1)#	Virtual Frame Relay Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example sets the N393 threshold for 5 events:

(config)# **interface frame-relay 1**
(config-fr 1)# **frame-relay lmi-n393dce 5**

frame-relay lmi-n393dte <*counter*>

Use the **frame-relay lmi-n393dte** command to set the N393 LMI monitored event counter for the DTE endpoint. Typical applications should leave the default value for this counter. Use the **no** form of this command to return to the default value.

Syntax Description

<*counter*>	Sets the counter value (valid range: 1-10)

Default Values

<*counter*>	**4 events**

Command Modes

(config-fr 1)#	Virtual Frame Relay Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example sets the N393 threshold for 5 events:

(config)# **interface frame-relay 1**
(config-fr 1)# **frame-relay lmi-n393dte 5**

frame-relay lmi-t391dte <*seconds*>

Use the **frame-relay lmi-t391dte** command to set the T391 signal polling timer for the DTE endpoint. Typical applications should leave the default value for this timer. Use the **no** form of this command to return to the default value.

Syntax Description

<*seconds*>	Sets the timer value in seconds (valid range: 5-30)

Default Values

<*seconds*>	**10 seconds**

Command Modes

(config-fr 1)#	Virtual Frame Relay Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

The T391 timer sets the time (in seconds) between polls to the frame relay network.

Usage Examples

The following example sets the T391 timer for 15 seconds:

(config)# **interface frame-relay 1**
(config-fr 1)# **frame-relay lmi-t391dte 15**

frame-relay lmi-t392dce *<seconds>*

Use the **frame-relay lmi-t392dce** command to set the T392 polling verification timer for the DCE endpoint. Typical applications should leave the default value for this timer. Use the **no** form of this command to return to the default value.

Syntax Description

<seconds>	Sets the timer value in seconds (valid range: 5-30)

Default Values

<seconds>	**10 seconds**

Command Modes

(config-fr 1)#	Virtual Frame Relay Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

The T392 sets the timeout (in seconds) between polling intervals. This parameter needs to be a few seconds longer than the T391 setting of the attached frame relay device.

Usage Examples

The following example sets the T392 timer for 15 seconds:

(config)# **interface frame-relay 1**
(config-fr 1)# f**rame-relay lmi-t392dce 15**

frame-relay lmi-type *<type>*

Use the **frame-relay lmi-type** command to define the frame relay signaling (LMI) type. Use the **no** form of the command to return to the default value.

Syntax Description

<type>	Sets the signaling type for the endpoint
ansi	Annex D signaling method
auto	Automatically determine signaling type by messages received on the frame circuit
cisco	Group of 4 signaling method
none	Turns off signaling on the endpoint. This is used for dial-backup connections to ADTRAN IQ and EXPRESS series products.
q933a	Annex A signaling method

Default Values

<type>	**ansi**

Command Modes

(config-fr 1)#	Virtual Frame Relay Interface Configuration Mode required

Command History

Release 1.1	Command was introduced
Release 2.1	Added signaling type **none** to provide support for dial-backup to ADTRAN IQ and EXPRESS series products.

Usage Examples

The following example sets the signaling method for the endpoint to **cisco**:

(config)# **interface frame-relay 1**
(config-fr 1)# **frame-relay lmi-type cisco**

snmp trap

Use the **snmp trap** command to enable all supported Simple Network Management Protocol (SNMP) traps on the interface.

Syntax Description

No subcommands

Default Values

By default, all interfaces (except virtual frame relay interfaces and sub-interfaces) have SNMP traps enabled.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), DDS (dds 1/1), serial (ser 1/1), virtual frame relay (fr 1), and SHDSL (shdsl 1/1) interfaces.

Command History

Release 1.1	Command was introduced

Usage Examples

The following example enables SNMP on the virtual frame relay interface:

(config)# **interface frame-relay 1**
(config-fr 1)# **snmp trap**

snmp trap link-status

Use the **snmp trap link-status** command to control the SNMP variable ifLinkUpDownTrapEnable (RFC 2863), which enables (or disables) the interface to send SNMP traps when there is an interface status change. Use the **no** form of this command to disable this trap.

Syntax Description

No subcommands

Default Values

By default, the ifLinkUpDownTrapEnable OID is set to enabled for all interfaces except virtual frame relay interfaces.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), T1 (t1 1/1), DSX-1 (t1 1/2), serial (ser 1/1), DDS (dds 1/1), virtual frame relay (fr 1), virtual PPP (ppp 1), and SHDSL (shdsl 1/1).

Command History

Release 1.1	Command was introduced

Functional Notes

The **snmp trap link-status** command is used to control the RFC 2863 ifLinkUpDownTrapEnable OID (OID number 1.3.6.1.2.1.31.1.1.1.14.0).

Usage Examples

The following example disables the link-status trap on the frame relay interface:

```
(config)# interface frame-relay 1
(config-fr 1)# no snmp trap link-status
```

FRAME RELAY SUB-INTERFACE CONFIG COMMAND SET

To activate the Frame Relay Interface Configuration command set, enter the **interface frame-relay** command (and specify a sub-interface) at the Global Configuration Mode prompt. For example:

```
Router> enable
Router# configure terminal
Router(config)# interface frame-relay 1.16
Router(config-fr 1.16)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

do on page 558

end on page 559

exit on page 560

All other commands for this command set are described in this section in alphabetical order.

access-policy <*policyname*> on page 413

bandwidth <*value*> on page 416

bridge-group <*group#*> on page 417

bridge-group <*group#*> *path-cost* <*value*> on page 418

bridge-group <*group#*> *priority* <*value*> on page 419

bridge-group <*group#*> *spanning-disabled* on page 420

crypto map <*mapname*> on page 421

dial-backup commands begin on page 423

frame-relay bc <*committed burst value*> on page 440

frame-relay be <*committed burst value*> on page 441

frame-relay interface-dlci <*dlci*> on page 442

ip commands begin on page 443

mtu <*size*> on page 459

access-policy *<policyname>*

Use the **access-policy** command to assign a specified access policy for the inbound traffic on an interface. Use the **no** form of this command to remove an access policy association.

*Configured access policies will only be active if the **ip firewall** command has been entered at the Global Configuration Mode prompt to enable the ADTRAN OS security features. All configuration parameters are valid, but no security data processing will be attempted unless the security features are enabled.*

Syntax Description

<policyname>	Alphanumeric descriptor for identifying the configured access policy (all access policy descriptors are case-sensitive)

Default Values

By default, there are no configured access policies associated with an interface.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), PPP virtual interfaces (ppp 1), and frame relay virtual sub-interfaces (fr 1.20).

Command History

Release 2.1	Command was introduced

Functional Notes

To assign an access policy to an interface, enter the Interface Configuration Mode for the desired interface and enter **access policy** <*policy name*>.

Usage Examples

The following example associates the access policy **UnTrusted** (to allow inbound traffic to the Web server) to the frame relay sub-interface labeled 1.16:

Enable the ADTRAN OS security features:
(config)# **ip firewall**

Usage Examples (Continued)

Create the access list (this is the packet selector):
(config)# **ip access-list extended InWeb**
(config-ext-nacl)# **permit tcp any host 63.12.5.253 eq 80**

Create the access policy that contains the access list **InWeb**:
(config)# **ip policy-class UnTrusted**
(config-policy-class)# **permit list InWeb**

Associate the access list with the ethernet 0/1 interface:
(config)# **interface frame-relay 1.16**
(config-fr 1.16)# **access-policy UnTrusted**

Technology Review

Creating access policies and lists to regulate traffic through the routed network is a four-step process:

Step 1:

Enable the security features of the ADTRAN OS using the **ip firewall** command.

Step 2:

Create an access list to permit or deny specified traffic. Standard access lists provide pattern matching for source IP addresses only. (Use extended access lists for more flexible pattern matching.) IP addresses can be expressed in one of three ways:

1. Using the keyword **any** to match any IP address. For example, entering **deny any** will effectively shut down the interface that uses the access list because all traffic will match the **any** keyword.
2. Using the **host** <A.B.C.D> to specify a single host address. For example, entering **permit 196.173.22.253** will allow all traffic from the host with an IP address of 196.173.22.253.
3. Using the <A.B.C.D> <wildcard> format to match all IP addresses in a “range”. Wildcard masks work in reverse logic from subnet mask. Specifying a one in the wildcard mask equates to a “don’t care”. For example, entering **deny 192.168.0.0 0.0.0.255** will deny all traffic from the 192.168.0.0/24 network.

Step 3:

Create an access policy that uses a configured access list. ADTRAN OS access policies are used to permit, deny, or manipulate (using NAT) data for each physical interface. Each ACP consists of a selector (access list) and an action (allow, discard, NAT). When packets are received on an interface, the configured ACPs are applied to determine whether the data will be processed or discarded. Possible actions performed by the access policy are as follows:

allow list <*access list names*>
All packets passed by the access list(s) entered will be allowed to enter the router system.

discard list <*access list names*>
All packets passed by the access list(s) entered will be dropped from the router system.

Technology Review (Continued)

allow list <*access list names*> **dest-policy** <*access policy name*>
All packets passed by the access list(s) entered and destined for the interface using the access policy listed will be permitted to enter the router system. This allows for configurations to permit packets to a single interface and not the entire system.

discard list <*access list names*> **dest-policy** <*access policy name*>
All packets passed by the access list(s) entered and destined for the interface using the access policy listed will be blocked from the router system. This allows for configurations to deny packets on a specified interface.

nat source list <*access list names*> **address** <*IP address*> **overload**
All packets passed by the access list(s) entered will be modified to replace the source IP address with the entered IP address. The **overload** keyword allows multiple source IP addresses to be replaced with the single IP address entered. This hides private IP addresses from outside the local network.

nat source list <*access list names*> **interface** <*interface*> **overload**
All packets passed by the access list(s) entered will be modified to replace the source IP address with the primary IP address of the listed interface. The **overload** keyword allows multiple source IP addresses to be replaced with the single IP address of the specified interface. This hides private IP addresses from outside the local network.

nat destination list <*access list names*> **address** <*IP address*>
All packets passed by the access list(s) entered will be modified to replace the destination IP address with the entered IP address. The **overload** keyword is not an option when performing NAT on the destination IP address; each private address must have a unique public address. This hides private IP addresses from outside the local network.

Step 4:

Apply the created access policy to an interface. To assign an access policy to an interface, enter Interface Configuration Mode for the desired interface and enter **access policy** <*policy name*>. The following example assigns access policy **MatchAll** to the frame relay sub-interface:

(config)# **interface frame-relay 1.16**
(config-fr 1.16)# **access-policy MatchAll**

bandwidth <*value*>

Use the **bandwidth** command to provide the bandwidth value of an interface to the higher-level protocols. This value is used in cost calculations. Use the **no** form of this command to restore the default values.

Syntax Description

<*value*>	Enter bandwidth in kbps.

Default Values

*To view default values use the **show interfaces** command.*

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), frame relay virtual sub-interface (fr 1.20), virtual PPP (ppp 1), and loopback interfaces.

Command History

Release 3.1	Command was introduced

Functional Notes

The **bandwidth** command is an informational value that is communicated to the higher-level protocols to be used in cost calculations. This is a routing parameter only and does not affect the physical interface.

Usage Examples

The following example sets bandwidth of the frame relay interface to 10 Mbps:

(config)# **interface frame-relay 1.7**
(config-fr 1.7)# **bandwidth 10000**

bridge-group *<group#>*

Use the **bridge-group** command to assign an interface to the specified bridge group. This command is supported on all Ethernet interfaces, PPP virtual interfaces, and frame relay virtual sub-interfaces. Use the **no** form of this command to remove the interface from the bridge group.

Syntax Description

<group#>	Bridge group number (1 to 255) specified using the **bridge-group** command

Default Values

By default, there are no configured bridge groups.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), and virtual frame relay sub-interfaces (fr 1.20).

Command History

Release 1.1	Command was introduced

Functional Notes

A bridged network can provide excellent traffic management to reduce collisions and limit the amount of bandwidth wasted with unnecessary transmissions when routing is not necessary. Any two interfaces can be bridged (Ethernet to T1 bridge, Ethernet to frame relay sub-interface).

Usage Examples

The following example assigns the frame relay sub-interface labeled 1.16 to bridge-group 1:

(config)# **interface frame-relay 1.16**
(config-fr 1.16)# **bridge-group 1**

bridge-group <*group#*> **path-cost** <*value*>

Use the **bridge-group path-cost** command to assign a cost to a bridge group that is used when computing the spanning-tree root path. To return to the default path-cost value, use the **no** form of this command.

Syntax Description

<*group#*>	Bridge group number (1 to 255) specified using the **bridge-group** command
<*value*>	Number assigned to the bridge interface to be used as the path cost in spanning calculations (valid range: 0 to 65535)

Default Values

<*value*>	19

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), and virtual frame relay interfaces (fr 1).

Command History

Release 1.1	Command was introduced

Functional Notes

The specified value is inversely proportional to the likelihood the bridge interface will be chosen as the root path. Set the path-cost value lower to increase the chance the interface will be the root. To obtain the most accurate spanning-tree calculations, develop a system for determining path costs for links and apply it to all bridged interfaces.

Usage Examples

The following example assigns a path cost of 100 for bridge group 17 on a frame relay sub-interface:

(config)# **interface frame-relay 1.16**
(config-fr 1.16)# **bridge-group 17 path-cost 100**

Technology Review

Spanning-tree protocol provides a way to prevent loopback or parallel paths in bridged networks. Using the priority values and path costs assigned to each bridging interface, the spanning-tree protocol determines the root path and identifies whether to block or allow other paths.

bridge-group <*group#*> **priority** <*value*>

Use the **bridge-group priority** command to select the priority level of a port associated with a bridge. To return to the default bridge-group priority value, use the **no** version of this command.

Syntax Description

<*group#*>	Bridge group number (1 to 255) specified using the **bridge-group** command
<*value*>	Priority value for the bridge group; the lower the value, the higher the priority (valid range: 0 to 255)

Default Values

<*value*>	128

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), and virtual frame relay sub-interfaces (fr 1.20).

Command History

Release 1.1	Command was introduced

Functional Notes

The only time that this priority level is used is when two interfaces with a path to the root have equal cost. At that point, the level set in this command will determine which port the bridge will use. Set the priority value lower to increase the chance the interface will be used.

Usage Examples

The following example sets the maximum priority on the frame relay sub-interface labeled 1.16 in bridge group 17:

(config)# **interface frame-relay 1.16**
(config-fr 1.16)# **bridge-group 17 priority 0**

bridge-group <*group#*> spanning-disabled

Use the **bridge-group spanning-disabled** command to transparently bridge two interfaces on a network (that have no parallel or redundant paths) without the overhead of spanning-tree protocol calculations. To enable the spanning-tree protocol on an interface, use the **no** form of this command.

Syntax Description

<*group#*>	Bridge group number (1 to 255) specified using the **bridge-group** command

Default Values

By default, spanning-tree protocol is enabled on all created bridge groups.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), and virtual frame relay interfaces (fr 1).

Command History

Release 1.1	Command was introduced

Functional Notes

When no parallel (redundant) paths exist within a bridged network, disabling the spanning tree protocol reduces traffic on the bridged interface. This traffic reduction can be helpful when bridging over a WAN link.

Before disabling the spanning-tree protocol on a bridged interface, verify that no redundant loops exist.

Usage Examples

The following example disables the spanning-tree protocol for bridge group 17 on the frame relay sub-interface labeled 1.16:

(config)# **interface frame-relay 1.16**
(config-fr 1.16)# **bridge-group 17 spanning-disabled**

Technology Review

Spanning-tree protocol provides a way to prevent loopback or parallel paths in bridged networks. Using the priority values and path costs assigned to each bridging interface, the spanning-tree protocol determines the root path and identifies whether to block or allow other paths.

crypto map <*mapname*>

Use the **crypto map** command to associate crypto maps with the interface.

When you apply a map to an interface, you are applying all crypto maps with the given map name. This allows you to apply multiple crypto maps if you have created maps which share the same name but have different map index numbers.

For VPN configuration example scripts, refer to the technical support note ***Configuring VPN*** *located on the* ***NetVanta 3000 Series System Manual*** *CD provided with your unit.*

Syntax Description

<*mapname*>	Enter the crypto map name that you wish to assign to the interface.

Default Values

By default, no crypto maps are assigned to an interface.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), virtual frame relay sub-interfaces (fr 1.20), and loopback interfaces

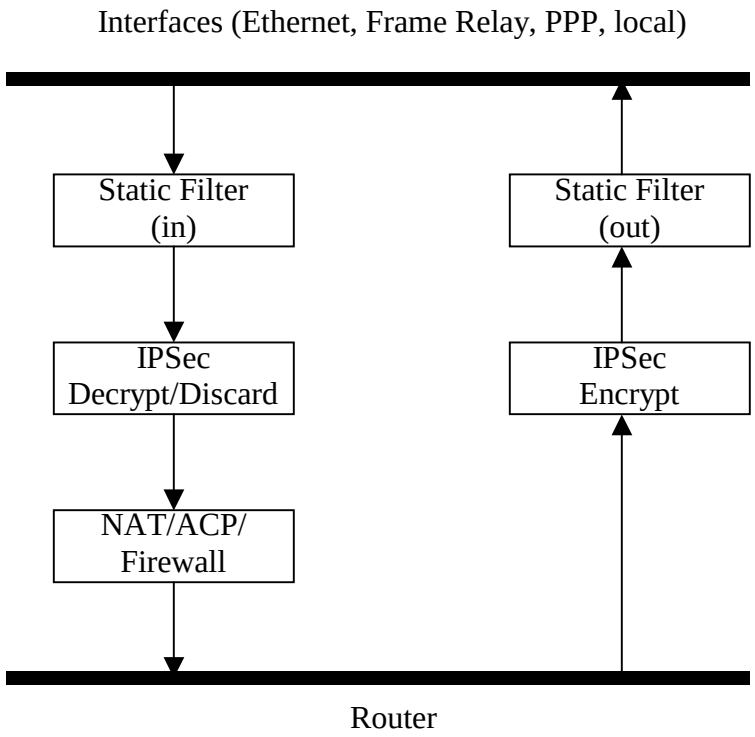
Command History

Release 4.1	Command was introduced

Functional Notes

When configuring a system to use both the stateful inspection firewall and IKE negotiation for VPN, keep the following notes in mind.

When defining the policy-class and associated access-control lists (ACLs) that describe the behavior of the firewall, do not forget to include the traffic coming into the system over a VPN tunnel terminated by the system. The firewall should be set up with respect to the un-encrypted traffic that is destined to be sent or received over the VPN tunnel. The following diagram represents typical AOS data-flow logic.



As shown in the diagram above, data coming into the product is first processed by the static filter associated with the interface on which the data is received. This access-group is a true static filter and is available for use regardless of whether the firewall is enabled or disabled. Next (if the data is encrypted) it is sent to the IPSec engine for decryption. The decrypted data is then processed by the stateful inspection firewall. Therefore, given a terminating VPN tunnel, only un-encrypted data is processed by the firewall.

The ACLs for a crypto map on an interface work in reverse logic to the ACLs for a policy-class on an interface. When specifying the ACLs for a crypto map, the source information is the private local-side, un-encrypted source of the data. The destination information will be the far-end, un-encrypted destination of the data. However, ACLs for a policy-class work in reverse. The source information for the ACL in a policy-class is the far-end. The destination information is the local-side.

Usage Examples

The following example applies all crypto maps with the name **MyMap** to the frame-relay interface:

(config-fr 1.16)# **crypto map MyMap**

dial-backup auto-backup

Use the **dial-backup auto-backup** command to configure the sub-interface to automatically attempt a dial-backup upon failure.

Syntax Description

No subcommands.

Default Values

By default, all backup endpoints will automatically attempt dial-backup upon a failure.

Command Modes

(config-fr 1.16)#	Virtual Frame Relay Sub-Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following enables automatic dial-backup on the endpoint:

(config)# **interface frame-relay 1.16**
(config-fr 1.16)# **dial-backup auto-backup**

dial-backup auto-restore

Use the **dial-backup auto-restore** command to configure the sub-interface to automatically discontinue dial backup when all network conditions are operational. Use the **no** form of this command to disable the auto-restore feature.

Syntax Description

No subcommands.

Default Values

By default, all backup endpoints will automatically restore the primary connection when the failure condition clears.

Command Modes

(config-fr 1.16)#	Virtual Frame Relay Sub-Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following configures the ADTRAN OS to automatically restore the primary connection when the failure condition clears:

(config)# **interface frame-relay 1.16**
(config-fr 1.16)# **dial-backup auto-restore**

dial-backup backup-delay *<seconds>*

Use the **dial-backup backup-delay** command to configure the amount of time the router will wait after the failure condition is recognized before attempting to backup the link. Use the **no** form of this command to return to the default value.

Syntax Description

<seconds>	Specifies the delay period (in seconds) a failure must be active before the ADTRAN OS will enter backup operation on the interface (valid range: 10 to 86400 seconds)

Default Values

<seconds>	**10 seconds**

Command Modes

(config-fr 1.16)#	Virtual Frame Relay Sub-Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following configures the ADTRAN OS to wait 60 seconds (on an endpoint with an active alarm condition) before attempting dial-backup operation:

(config)# **interface frame-relay 1.16**
(config-fr 1.16)# **dial-backup backup-delay 60**

dial-backup call-mode <*role*>

Use the **dial-backup call-mode** command to combine user data with pattern data to ensure data does not mirror standard DDS loop codes (use only on 64 kbps circuits without frame relay signaling). Use the **no** form of this command to return to the default value.

Syntax Description

<*role*>	Selects the role the router will take in backup of this sub-interface.
answer	Answer and backup primary link on failure
answer-always	Answer and backup regardless of primary link state
originate	Originate backup call on primary link failure
originate-answer	Originate or answer call on primary link failure
originate-answer-always	Originate on failure answer and backup always

Default Values

<*role*>	**originate-answer**

Command Modes

(config-fr 1.16)#	Virtual Frame Relay Sub-Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following configures the ADTRAN OS to answer dial-backup calls on this endpoint but never generate calls:

(config)# **interface frame-relay 1.16**
(config-fr 1.16)# **dial-backup call-mode answer-always**

dial-backup connect-timeout *<seconds>*

Use the **dial-backup connect-timeout** command to specify the number of seconds to wait for a connection after a call is attempted before trying to call again or dialing a different number. It is recommended this number be greater than 60.

Syntax Description

<seconds>	Selects the amount of time in seconds that the router will wait for a connection before attempting another call (valid range: 10 to 300)

Default Values

<seconds>	**60 seconds**

Command Modes

(config-fr 1.16)#	Virtual Frame Relay Sub-Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following configures the ADTRAN OS to wait 120 seconds before retrying a failed dial-backup call:

(config)# **interface fr 1.16**
(config-fr 1.16)# **dial-backup connect-timeout 120**

dial-backup force <*state*>

Use the **dial-backup force** command to manually override the automatic dial backup feature. This can be used to force a link into backup to allow maintenance to be performed on the primary link without disrupting data. Use the **no** form of this command to return to the normal dial-backup operation state.

Syntax Description

<*state*>	Selects the forced backup state of the sub-link.
backup	Force backup regardless of primary link state
primary	Force primary link regardless of its state

Default Values

By default, this feature is disabled.

Command Modes

(config-fr 1.16)#	Virtual Frame Relay Sub-Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following configures the ADTRAN OS to force this endpoint into dial-backup:

(config)# **interface fr 1.16**
(config-fr 1.16)# **dial-backup force backup**

dial-backup iq-handshake

Use the **dial-backup iq-handshake** command for dial-backup operation with ADTRAN IQ and Express family products. Using the **dial-backup iq-handshake** command enables the association of incoming calls with packet endpoints in cases where there is a single call-in number (hunt group) and no Caller ID information available. Use the **no** form of this command to disable the IQ handshake.

Syntax Description

No subcommands

Default Values

*By default, the **dial-backup iq-handshake** feature is disabled.*

Command Modes

(config-fr 1.16)#	Virtual Frame Relay Sub-Interface Configuration Mode required

Command History

Release 2.1	Command was introduced

Functional Notes

The ADTRAN IQ and Express products implement a proprietary dial-backup mechanism to transfer configuration information (such as DLCIs) when entering dial-backup. The **iq-handshake** command enables this proprietary data transfer in the ADTRAN OS.

Usage Examples

The following example enables the proprietary data transfer (to an IQ or Express product) on the dial-backup interface:

(config)# **interface fr 1.16**
(config-fr 1.16)# **dial-backup iq-handshake**

dial-backup maximum-retry *<attempts>*

Use the **dial-backup maximum-retry** command to select the number of calls the router will make when attempting to backup a link. Use the **no** form of this command to return to the default state.

Syntax Description

<attempts>	Selects the number of call retries that will be made after a sub-link failure (valid range: 0 to 15). Setting this value to 0 will allow unlimited retries during the time the network is failed.

Default Values

<attempts>	**0 attempts**

Command Modes

(config-fr 1.16)#	Virtual Frame Relay Sub-Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example configures the ADTRAN OS to retry a dial-backup call 4 times before considering backup operation not available:

(config)# **interface fr 1.16**
(config-fr 1.16)# **dial-backup maximum-retry 4**

dial-backup number *<digits> <call type> <isdn min chan> <isdn max chan>*

Use the **dial-backup number** command to configure the phone number and the call type the router will dial upon network failure. Multiple entries can be made for a sub-interface to allow alternate sites to be dialed.

Syntax Description

<digits>	Enter the phone numbers to call when the backup is initiated
<call type>	Selects the type of call the router will attempt
analog	Number connects to an analog modem
digital-56k	Number belongs to a digital 56 kbps per DS0 connection
digital-64k	Number belongs to a digital 64kbps per DS0 connection

Default Values

By default, there are no configured dial-backup numbers.

Command Modes

(config-fr 1.16)#	Virtual Frame Relay Sub-Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example configures the ADTRAN OS to dial 704-555-1212 (digital 64 kbps connection) to initiate dial-backup operation on this endpoint:

(config)# **interface fr 1.16**
(config-fr 1.16)# **dial-backup number 7045551212 digital-64k**

dial-backup priority <*value*>

Use the **dial-backup priority** command to select the backup priority for this interface. Allows the user to establish the highest priority backup link and ensure that link will override backups attempted by lower priority links. Use the **no** form of this command to return to the default value.

Syntax Description

<*value*>	Sets the relative priority to this link (valid range: 0 to 100). A value of 100 designates the highest priority.

Default Values

<*value*>	**50**

Command Modes

(config-fr 1.16)#	Virtual Frame Relay Sub-Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example assigns the highest priority to this endpoint:

(config)# **interface fr 1.16**
(config-fr 1.16)# **dial-backup priority 100**

dial-backup protocol *<protocol-type>*

Use the **dial-backup protocol** command to select the protocol to use over the backup link. Use the **no** form of this command to return to the default value.

Syntax Description

<protocol-type>	
frame-relay	The router will perform frame relay signaling on the backup link. This protocol is supported in all products using the ADTRAN OS as well as ADTRAN's ATLAS IQ and Express series products.

Default Values

<protocol-type>	**frame-relay**

Command Modes

(config-fr 1.16)#	Virtual Frame Relay Sub-Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example configures this endpoint for frame relay dial-backup operation:

(config)# **interface fr 1.16**
(config-fr 1.16)# **dial-backup protocol frame-relay**

dial-backup randomize-timers

Use the **dial-backup randomize-timers** command to randomize the call timers to minimize potential contention for resources. Use the **no** form of this command to return to the default value.

Syntax Description

No subcommands.

Default Values

By default, the ADTRAN OS does not randomize the dial-backup call timers.

Command Modes

(config-fr 1.16)#	Virtual Frame Relay Sub-Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example configures the ADTRAN OS to randomize the dial-backup timers associated with this endpoint:

(config)# **interface fr 1.16**
(config-fr 1.16)# **dial-backup randomize-timers**

dial-backup redial-delay *<seconds>*

Use the **dial-backup redial-delay** command to configure the delay after an unsuccessful call until the call will be re-tried.

Syntax Description

<seconds>	The delay is seconds between attempting to re-dial a failed backup attempt (valid range: 10 to 3600)

Default Values

<seconds>	**10 seconds**

Command Modes

(config-fr 1.16)#	Virtual Frame Relay Sub-Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example configures a redial delay of 25 seconds on this endpoint:

(config)# **interface fr 1.16**
(config-fr 1.16)# **dial-backup redial-delay 25**

dial-backup remote-dlci <*dlci*>

Use the **dial-backup remote-dlci** command to select the Data Link Connection Identifier (DLCI) for the remote unit. When backing up to an ADTRAN IQ device, the DLCI value should match the primary DLCI configured on the IQ unit. When backing up to another AOS device, the two AOS devices must have the same DLCI value entered for this command. They do not necessarily have to match the primary DLCI on the frame relay interface. Use the **no** form of this command to remove a configured DLCI.

Syntax Description

<*DLCI*>	Remote DLCI number (valid range: 16 to 1007)

Default Values

<*DLCI*>	1500

Command Modes

(config-fr 1.16)#	Virtual Frame Relay Sub-Interface Configuration Mode required

Command History

Release 2.1	Command was introduced

Usage Examples

The following example configures a remote DLCI of 70 on this endpoint:

(config)# **interface fr 1.16**
(config-fr 1.16)# **dial-backup remote-dlci 70**

dial-backup restore-delay *<seconds>*

Use the **dial-backup restore-delay** command to configure the amount of time the router will wait after the network is restored before disconnecting the backup link and reverting to the primary. This setting is used to prevent disconnecting the backup link if the primary link is "bouncing" in and out of alarm.

Syntax Description

<seconds>	Specifies the number of seconds the ADTRAN OS will wait (after a primary link is restored) before disconnecting dial-backup operation (valid range: 10 to 86400)

Default Values

<seconds>	**10 seconds**

Command Modes

(config-fr 1.16)#	Virtual Frame Relay Sub-Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example configures the ADTRAN OS to wait 30 seconds before disconnecting dial-backup operation and restoring the primary connection for this endpoint:

(config)# **interface fr 1.16**
(config-fr 1.16)# **dial-backup restore-delay 30**

dial-backup schedule <*type*>

Use the **dial-backup schedule** command to set the time of day that backup will be enabled. Use this command if backup is desired only during normal business hours and on specific days of the week. Use the **no** form of this command to disable dial-backup (as specified).

Syntax Description

<*type*>	Time is entered in 24-hour format (00:00)
day	Sets the days to allow backup (Valid Monday-Sunday)
enable-time	Sets the time of day to enable backup
disable-time	Sets the time of day to disable backup

Default Values

By default, dial-backup is enabled for all days and times if the dial-backup auto-backup command has been issued and the dial-backup schedule has not been entered.

Command Modes

(config-fr 1.16)#	Virtual Frame Relay Sub-Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example enables dial-backup Monday through Friday 8:00 am to 7:00 pm:

(config)# **interface fr 1.16**
(config-fr 1.16)# **dial-backup schedule enable-time 08:00**
(config-fr 1.16)# **dial-backup schedule disable-time 19:00**
(config-fr 1.16)# **no dial-backup schedule day Saturday**
(config-fr 1.16)# **no dial-backup schedule day Sunday**

dial-backup shutdown

Use the **dial-backup shutdown** command to deactivate all dial-backup functionality in the unit. Dial-backup configuration parameters are kept intact, but the unit will not initiate (or respond) to dial-backup sequences in the event of a network outage. Use the **no** form of this command to reactivate the dial-backup interface.

Syntax Description

No subcommands.

Default Values

By default, all ADTRAN OS interfaces are disabled.

Command Modes

(config-fr 1.16)#	Virtual Frame Relay Sub-Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example deactivates the configured dial-backup interface:

(config)# **interface fr 1.16**
(config-fr 1.16)# **dial-backup shutdown**

frame-relay bc <*committed burst value*>

Use the **frame-relay bc** command to set the b_C (committed burst) value for a frame relay sublink. The value is in bits. The time interval is always one second, so this can also be considered bits per second. Shaping is performed on a sliding one-second window to make maximum use of configured bandwidth.

Syntax Description

<*committed burst value*>	Enter the committed burst value (in bits) for the sublink.

Default Values

The default is 0 (no limit).

Command Modes

(config-fr 1.1)#	Virtual Frame Relay Sub-Interface Configuration Mode required

Command History

Release 4.1	Command was introduced

Usage Examples

The following example configures sublink fr 1.1 with a committed burst value of 128000 bits:

```
(config)# interface fr 1.1
(config-fr 1.1)# frame-relay bc 128000
```


frame-relay be <*committed burst value*>

Use the **frame-relay be** command to set the b_e (excessive burst) value for a frame relay sublink. The value is in bits. The time interval is always one second, so this can also be considered bits per second. Shaping is performed on a sliding one-second window to make maximum use of configured bandwidth. Note that when both b_c and b_e are non-zero, shaping is performed on the virtual circuit. The circuit is limited to the sum of b_c and b_e.

Syntax Description

<*committed burst value*>	Enter the committed burst value (in bits) for the sublink.

Default Values

The default is 0 (no limit).

Command Modes

(config-fr 1.1)#	Virtual Frame Relay Sub-Interface Configuration Mode required

Command History

Release 4.1	Command was introduced

Usage Examples

The following example configures sublink fr 1.1 with an excessive burst value of 64000 bits:

```
(config)# interface fr 1.1
(config-fr 1.1)# frame-relay be 64000
```

frame-relay interface-dlci *<dlci>*

Use the **frame-relay interface-dlci** command to configure the Data Link Connection Identifier (DLCI) for the frame relay sub-interface. This setting should match the DLCI supplied by your frame relay service provider. Use the **no** form of this command to remove the configured DLCI.

Syntax Description

<dlci>	Enter numeric value supplied by your provider

Default Values

<dlci>	The default DLCI is populated with the sub-interface identifier. For example, if configuring the virtual frame relay sub-interface labeled **fr 1.20** , the default DLCI is **20**.

Command Modes

(config-fr 1.1)#	Virtual Frame Relay Sub-Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example configures a DLCI of 72 for this frame relay endpoint:

(config)# **interface fr 1.16**
(config-fr 1.16)# **frame-relay interface-dlci 72**

ip access-group <*listname*> [**in | out**]

Use the **ip access-group** command to create an access list to be used for packets transmitted to/from the asynchronous host. Use the **no** form of this command to disable this type of control.

Syntax Description

listname	Assigned IP access list name.
in	Enables access control on packets *transmitted from* the host.
out	Enables access control on packets *sent to* the host.

Default Values

By default, these commands are disabled.

Command Modes

(config-interface)#	Interface Configuration Mode required.

Command History

Release 3.1	Command was introduced

Functional Notes

When this command is enabled, the IP destination address of each packet must be validated before being passed through. If the packet is not acceptable per these settings, it is dropped.

Usage Examples

The following example sets up the router to only allow Telnet traffic into the frame relay sub-interface:

(config)# **ip access-list extended TelnetOnly**
(config-ext-nacl)#**permit tcp any any eq telnet**
(config-ext-nacl)#**int frame-relay 1.16**
(config-fr 1.16)#**ip access-group TelnetOnly in**

ip address dhcp

Use the **ip address dhcp** command to use Dynamic Host Configuration Protocol (DHCP) to obtain an address on the interface. Use the **no** form of this command to remove a configured IP address (using DHCP) and disable DHCP operation on the interface.

ip address dhcp {client-id [<*interface*> | <*identifier*>] hostname "<*string*>" }

Syntax Description

client-id ***Optional**	Specifies the client identifier used when obtaining an IP address from a DHCP server.
<*interface*>	Specifying an interface defines the client identifier as the hexadecimal MAC address of the specified interface (including a hexadecimal number added to the front of the MAC address to identify the media type). For example, specifying the **client-id ethernet 0/1** (where the Ethernet interface has a MAC address of d217.0491.1150) defines the client identifier as **01:d2:17:04:91:11:50** (where 01 defines the media type as Ethernet).
<*identifier*>	Specifies a custom client-identifier using a text string (that is converted to a hexadecimal equivalent) or 7 to 28 hexadecimal numbers (with colon delimiters). For example, a custom client identifier of **0f:ff:ff:ff:ff:51:04:99:a1** may be entered using the <*identifier*> option.
host-name ***Optional**	Specifies a text string (to override the global router name) to use as the name in the DHCP option 12 field.
"<*string*>"	String (encased in quotation marks) of up to 35 characters to use as the name of the host for DHCP operation.

Default Values

client-id ***Optional**	By default, the client identifier is populated using the following formula: TYPE: INTERFACE SPECIFIC INFO : MAC ADDRESS Where TYPE specifies the media type in the form of one hexadecimal byte (refer to the **hardware-address** command for a detailed listing of media types), and the MAC ADDRESS is the Media Access Control (MAC) address assigned to the first Ethernet interface in the unit in the form of six hexadecimal bytes. (For units with a single Ethernet interface, the MAC ADDRESS assigned to ethernet 0/1 is used in this field). INTERFACE SPECIFIC INFO is only used for frame relay interfaces and can be determined using the following: FR_PORT# : Q.922 ADDRESS Where the FR_PORT# specifies the label assigned to the virtual frame relay

Default Values (Continued)

interface using four hexadecimal bytes. For example, a virtual frame relay interface labeled 1 would have a FR_PORT# of 00:00:00:01.

The Q.922 ADDRESS field is populated using the following:

8	7	6	5	4	3	2	1
DLCI (high order)						C/R	EA
DLCI (lower)				FECN	BECN	DE	EA

Where the FECN, BECN, C/R, DE, and high order EA bits are assumed to be 0 and the lower order extended address (EA) bit is set to 1.

The following list provides a few example DLCIs and associated Q.922 address:
DLCI (decimal) / Q.922 address (hex)

16 / 0x0401
50 / 0x0C21
60 / 0x0CC1
70 / 0x1061
80 / 0x1401

host-name
Optional

"<string>" By default, the hostname is the name configured using the Global Configuration **hostname** command.

Command Modes

(config-ppp 1)# Interface Configuration Mode required.

Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), and virtual frame relay sub-interfaces (fr 1.20)

Command History

Release 2.1 Command was introduced

Functional Notes

DHCP allows interfaces to acquire a dynamically assigned IP address from a configured DHCP server on the network. Many Internet Service Providers (ISPs) require the use of DHCP when connecting to their services. Using DHCP reduces the number of dedicated IP addresses the ISP must obtain. Consult your ISP to determine the proper values for the **client-id** and **hostname** fields.

Usage Examples

The following example enables DHCP operation on the virtual frame-relay interface (labeled 1.16):

(config)# **interface frame-relay 1.16**
(config-fr 1.16)# **ip address dhcp**

ip address <*address*> <*mask*> secondary

Use the **ip address** command to define an IP address on the specified interface. Use the optional **secondary** keyword to define a secondary IP address. Use the **no** form of this command to remove a configured IP address.

Syntax Description

<*address*>	Defines the IP address for the interface in dotted decimal notation (for example: 192.22.73.101).
<*mask*>	Specifies the subnet mask that corresponds to the listed IP address.
secondary **Optional*	Optional keyword used to configure a secondary IP address for the specified interface.

Default Values

By default, there are no assigned IP addresses.

Command Modes

(config-interface)#	Interface Configuration Mode required.

Command History

Release 1.1	Command was introduced

Functional Notes

Use secondary IP addresses to allow dual subnets on a single interface (when you need more IP addresses than the primary subnet can provide). When using secondary IP addresses, avoid routing loops by verifying that all devices on the network segment are configured with secondary IP addresses on the secondary subnet.

Usage Examples

The following example configures a secondary IP address of **192.22.72.101/30**:

(config)# **interface frame-relay 1.16**
(config-fr 1.16)# **ip address 192.22.72.101 255.255.255.252 secondary**

ip dhcp [release | renew]

Use the **ip dhcp** command to release or renew the DHCP IP address. This command is only applicable when using DHCP for IP address assignment.

Syntax Description

release	Use this keyword to release DHCP IP address.
renew	Use this keyword to renew DHCP IP address.

Default Values

No default values required for this command.

Command Modes

(config-interface)#	Interface Configuration Mode required (applies only to virtual interfaces)

Command History

Release 3.1	Command was introduced

Usage Examples

The following example releases the IP DHCP address for the virtual interface:

(config)# **interface frame-relay 1.3**
(config-fr 1.3)# **ip dhcp release**

ip helper-address *<address>*

Use the **ip helper-address** command to configure the ADTRAN OS to forward User Datagram Protocol (UDP) broadcast packets received on the interface. Use the **no** form of this command to disable forwarding packets.

> NOTE *The **ip helper** command must be used in conjunction with the **ip forward-protocol** command to configure the ADTRAN OS to forward UDP broadcast packets. See ip forward-protocol udp <port number> on page 200 for more information.*

Syntax Description

<address>	Specifies the destination IP address (in dotted decimal notation) for the forwarded UDP packets.

Default Values

By default, broadcast UDP packets are not forwarded.

Command Modes

(config-interface)#	Interface Configuration Mode required (applies to all physical interfaces as well as virtual interfaces)

Command History

Release 1.1	Command was introduced

Functional Notes

When used in conjunction with the **ip forward-protocol** command, the **ip helper-address** feature allows you to customize which broadcast packets are forwarded.

To implement the helper address feature, assign a helper-address(es) (specifying the device that needs to receive the broadcast traffic) to the interface closest to the host that transmits the broadcast packets. When broadcast packets (of the specified type forwarded using the **ip forward-protocol** command) are received on the interface, they will be forwarded to the device that needs the information.

Only packets meeting the following criteria are considered eligible by the **ip helper-address** feature:
1. The packet IP protocol is User Datagram Protocol (UDP).
2. Any UDP port specified using the **ip forward-protocol** command.
3. The media access control (MAC) address of the frame is an all-ones broadcast address (ffff.ffff.ffff).
4. The destination IP address is broadcast defined by all ones (255.255.255.255) or a subnet broadcast (for example, 192.33.4.251 for the 192.33.4.248/30 subnet).

Usage Examples

The following example forwards all DNS broadcast traffic to the DNS server with IP address 192.33.5.99:

(config)# **ip forward-protocol udp domain**
(config)# **interface frame-relay 1.16**
(config-fr 1.16)# **ip helper-address 192.33.5.99**

ip ospf

Use the **ip ospf** command to customize OSPF settings (if needed).

Syntax Description

authentication-key <*password*>	Assign a simple-text authentication password to be used by other routers using the OSPF simple password authentication.
cost <*value*>	Specify the OSPF cost of sending a packet on the interface. This value overrides any computed cost value. Range: 1-65535.
dead-interval <*seconds*>	Set the maximum interval allowed between hello packets. If the maximum is exceeded, neighboring devices will determine that the device is down. Range: 0-32767.
hello-interval <*seconds*>	Specify the interval between hello packets sent on the interface. Range: 0-32767.
message-digest-key <*keyid*> **md5** <*key*>	Configure OSPF Message Digest 5 (MD5) authentication (16-byte max) keys.
priority <*value*>	Set the OSPF priority. The value set in this field helps determine the designated router for this network. Range: 0-255.
retransmit-interval <*seconds*>	Specify the time between link-state advertisements (LSAs). Range: 0-32767.
transmit-delay <*seconds*>	Set the estimated time required to send an LSA on the interface. Range: 0-32767.

Default Values

retransmit-interval <*seconds*>	5 seconds
transmit-delay <*seconds*>	1 second
hello-interval <*seconds*>	10 seconds: Ethernet, point-to-point, frame relay, and ppp
dead-interval <*seconds*>	40 seconds

Command Modes

(config-interface)#	Valid interfaces include: Ethernet (eth 0/1), virtual frame relay (fr 1), and virtual PPP (ppp 1).

Command History

Release 3.1	Command was introduced

ip ospf authentication [message-digest | null]

Use the **ip ospf authentication** command to authenticate an interface that is performing OSPF authentication.

Syntax Description

message-digest **Optional*	Select message-digest authentication type.
null **Optional*	Select for no authentication to be used.

Default Values

By default, this is set to null (meaning no authentication is used).

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), virtual frame relay sub-interfaces (fr 1.20), and loopback interfaces

Command History

Release 3.1	Command was introduced

Usage Examples

The following example specifies that no authentication will be used on the frame-relay interface:

(config-fr 1.16)# **ip ospf authentication null**

ip ospf network [broadcast | point-to-point]

Use the **ip ospf network** command to specify the type of network on this interface.

Syntax Description

broadcast	Set the network type for broadcast.
point-to-point	Set the network type for point-to-point.

Default Values

By default, Ethernet defaults to broadcast. PPP and frame relay default to point-to-point.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), virtual frame relay sub-interfaces (fr 1.20), and loopback interfaces

Command History

Release 3.1	Command was introduced

Functional Notes

A point-to-point network will not elect designated routers.

Usage Examples

The following example designates a broadcast network type:

(config-fr 1.16)# **ip ospf network broadcast**

ip proxy-arp

Use the **ip proxy-arp** to enable proxy Address Resolution Protocol (ARP) on the interface. Use the **no** form of this command to disable this feature.

Syntax Description

<address>	Defines the IP address for the interface in dotted decimal notation (for example: 192.22.73.101)
<subnet mask>	Specifies the subnet mask that corresponds to the listed IP address

Default Values

By default, proxy arp is enabled.

Command Modes

(config-interface)#	Interface Configuration Mode required (applies to all physical interfaces as well as virtual interfaces)

Command History

Release 1.1	Command was introduced

Functional Notes

In general, the principle of proxy-arp allows a router to insert its IP address in the source IP address field of a packet (if the packet is from a host on one of its subnetworks). This allows hosts to reach devices on other subnetworks without implementing routing or specifying a default gateway.

If proxy-arp is enabled, the ADTRAN OS will respond to all proxy-arp requests with its specified MAC address and forward packets accordingly.

Enabling proxy-arp on an interface may introduce unnecessary ARP traffic on the network.

Usage Examples

The following enables proxy-arp on the frame relay sub-interface:

(config)#**interface frame-relay 1.16**
(config-fr 1.16)# **ip proxy-arp**

ip rip receive version *<version>*

Use the **ip rip receive version** command to configure the RIP version the unit accepts in all RIP packets received on the interface. Use the **no** form of this command to restore the default value.

Syntax Description

<version>	Specifies the RIP version
1	Only accept received RIP version 1 packets on the interface
2	Only accept received RIP version 2 packets on the interface

Default Values

*By default, all interfaces implement RIP version 1 (the default value for the **version** command).*

Command Modes

(config-interface)#	Interface Configuration Mode required (applies to all physical interfaces as well as virtual interfaces)

Command History

Release 1.1	Command was introduced

Functional Notes

Use the **ip rip receive version** to specify a RIP version that will override the **version** (in the Router RIP command set) configuration.

The ADTRAN OS only accepts one version (either 1 or 2) on a given interface.

Usage Examples

The following example configures a frame relay sub-interface to accept only RIP version 2 packets:

(config)# **interface frame-relay 1.16**
(config-fr 1.16)# **ip rip receive version 2**

ip rip send version <*version*>

Use the **ip rip send version** command to configure the RIP version the unit sends in all RIP packets transmitted on the interface. Use the **no** form of this command to restore the default value.

Syntax Description

<*version*>	Specifies the RIP version
1	Only transmits RIP version 1 packets on the interface
2	Only transmits RIP version 2 packets on the interface

Default Values

*By default, all interfaces transmit RIP version 1 (the default value for the **version** command)*

Command Modes

(config-interface)#	Interface Configuration Mode required (applies to all physical interfaces as well as virtual interfaces)

Command History

Release 1.1	Command was introduced

Functional Notes

Use the **ip rip send version** to specify a RIP version that will override the **version** (in the Router RIP command set) configuration.

The ADTRAN OS only transmits one version (either 1 or 2) on a given interface.

Usage Examples

The following example configures a frame relay sub-interface to transmit only RIP version 2 packets:

(config)# **interface frame-relay 1.16**
(config-fr 1.16)# **ip rip send version 2**

ip route-cache <*address*>

Use the **ip route-cache** command to enable fast-cache switching on the interface. Use the **no** form of this command to disable fast-cache switching and return to process switching mode.

*Using Network Address Translation (NAT) or the ADTRAN OS firewall capabilities on an interface requires process switching mode (using the **no ip route-cache** command).*

Syntax Description

No subcommands.

Default Values

By default, fast-cache switching is enabled on all Ethernet and virtual frame relay sub-interfaces. IP route-cache is disabled for all virtual PPP interfaces.

Command Modes

(config-interface)#	Interface Configuration Mode required Valid interfaces include: Ethernet (eth 0/1), virtual frame relay sub-interfaces (fr 1.16), and virtual PPP interfaces (ppp 1).

Command History

Release 2.1	Command was introduced

Functional Notes

Fast switching allows an IP interface to provide optimum performance when processing IP traffic.

Usage Examples

The following example enables fast switching on a frame relay sub-interface:

(config)# **interface frame-relay 1.16**
(config-fr 1.16)# **ip route-cache**

ip unnumbered *<interface>*

Use the **ip unnumbered** command to use the IP address assigned to the specified interface for all IP processing on the active interface. Use the **no** form of this command to remove the unnumbered configuration.

Syntax Description

<interface>	Specifies the interface (in the format **type slot/port**) that contains the IP address to use as the source address for all packets transmitted on this interface. Valid interfaces include: Ethernet (eth 0/1), virtual frame relay sub-interfaces (fr 1.16), virtual PPP (ppp 1), and loopback interfaces.

Default Values

*By default, all interfaces are configured to use a specified IP address (using the **ip address** command).*

Command Modes

(config-interface)#	Interface Configuration Mode required Valid interfaces include: Ethernet (eth 0/1), virtual frame relay sub-interfaces (fr 1.16), and virtual PPP interfaces (ppp 1).

Command History

Release 1.1	Command was introduced

Functional Notes

If **ip unnumbered** is enabled on an interface, all IP traffic from the interface will use a source IP address taken from the specified interface. For example, specifying **ip unnumbered eth 0/1** while in the Frame Relay Sub-Interface Configuration Mode configures the frame relay sub-interface to use the IP address assigned to the Ethernet interface for all IP processing. In addition, the ADTRAN OS uses the specified interface information when sending route updates over the unnumbered interface.

Usage Examples

The following example configures the frame relay interface (labeled **frame-relay 1.16**) to use the IP address assigned to the Ethernet interface (**eth 0/1**):

(config)# **interface frame-relay 1.16**
(config-fr 1.16)# **ip unnumbered eth 0/1**

mtu <*size*>

Use the **mtu** command to configure the maximum transmit unit (MTU) size for the active interface. Use the **no** form of this command to return to the default value.

Syntax Description

<*size*>	Configures the window size for transmitted packets. The valid ranges for the various interfaces are listed below:	
	Ethernet (eth 0/1)	64 to 1500
	virtual frame relay sub-interfaces (fr 1.16)	64 to 1520
	virtual PPP interfaces (ppp 1)	64 to 1500
	loopback interfaces	64 to 1500

Default Values

<*size*>	The default values for the various interfaces are listed below:	
	Ethernet (eth 0/1)	1500
	virtual frame relay sub-interfaces (fr 1.16)	1500
	virtual PPP interfaces (ppp 1)	1500
	loopback interfaces	1500

Command Modes

(config-interface)#	Interface Configuration Mode required (applies only to IP interfaces) Valid interfaces include: Ethernet (eth 0/1), virtual frame relay sub-interfaces (fr 1.16), virtual PPP interfaces (ppp 1), and loopback interfaces.

Command History

Release 1.1	Command was introduced

Functional Notes

OSPF will not become adjacent on links where the MTU sizes do not match. If router A and router B are exchanging hello packets but their MTU sizes do not match, they will never reach adjacency. This is by design and required by the RFC.

Usage Examples

The following example specifies an MTU of 1200 on the frame-relay interface:

```
(config)# interface fr 1.16
(config-fr 1.16)# mtu 1200
```

PPP INTERFACE CONFIGURATION COMMAND SET

To activate the PPP Interface Configuration command set, enter the **interface ppp** command at the Global Configuration Mode prompt. For example:

```
Router> enable
Router# configure terminal
Router(config)# interface ppp 1
Router(config-ppp 1)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

description on page 557

do on page 558

end on page 559

exit on page 560

shutdown on page 561

All other commands for this command set are described in this section in alphabetical order.

access-policy <policyname> on page 461

alias link<"text"> on page 464

bandwidth <value> on page 465

bridge-group <group#> on page 466

bridge-group <group#> spanning-disabled on page 467

crypto map <mapname> on page 468

ip commands begin on page 470

keepalive <seconds> on page 482

mtu <size> on page 483

peer default ip address <address> on page 484

ppp authentication <protocol> on page 485

ppp chap hostname <hostname> on page 489

ppp chap password <password> on page 490

ppp pap sent-username <username> password <password> on page 491

snmp trap link-status on page 492

username <username> password <password> on page 493

access-policy *<policyname>*

Use the **access-policy** command to assign a specified access policy to an interface. Use the **no** form of this command to remove an access policy association.

Syntax Description

<policyname>	Alphanumeric descriptor for identifying the configured access policy.

All access policy descriptors are case-sensitive.

Default Values

By default, there are no configured access policies associated with an interface.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), and virtual frame relay sub-interfaces (fr 1.20).

Command History

Release 2.1	Command was introduced

Usage Examples

The following example associates the access policy **UnTrusted** (to allow inbound traffic to the Web server) to the virtual PPP interface:

Enable the ADTRAN OS security features:
(config)# **ip firewall**

Create the access list (this is the packet selector):
(config)# **ip access-list extended InWeb**
(config-ext-nacl)# **permit tcp any host 63.12.5.253 eq 80**

Create the access policy that contains the access list **InWeb**:
(config)# **ip policy-class UnTrusted**
(config-policy-class)# **allow list InWeb**

Usage Examples (Continued)

Associate the access list with the PPP virtual interface (labeled 1):
(config)# **interface ppp 1**
(config-ppp 1)# **access-policy UnTrusted**

Technology Review

Creating access policies and lists to regulate traffic through the routed network is a four-step process:

Step 1:

Enable the security features of the ADTRAN OS using the **ip firewall** command.

Step 2:

Create an access list to permit or deny specified traffic. Standard access lists provide pattern matching for source IP addresses only. (Use extended access lists for more flexible pattern matching.) IP addresses can be expressed in one of three ways:

1. Using the keyword **any** to match any IP address. For example, entering **deny any** will effectively shut down the interface that uses the access list because all traffic will match the **any** keyword.
2. Using the **host** <A.B.C.D> to specify a single host address. For example, entering **permit host 196.173.22.253** will allow all traffic from the host with an IP address of 196.173.22.253.
3. Using the <A.B.C.D> <wildcard> format to match all IP addresses in a “range”. Wildcard masks work in reverse logic from subnet mask. Specifying a one in the wildcard mask equates to a “don’t care”. For example, entering **deny 192.168.0.0 0.0.0.255** will deny all traffic from the 192.168.0.0/24 network.

*The command **permit <A.B.C.D>** will also be assumed to mean **permit host <A.B.C.D>.***

Step 3:

Create an access policy that uses a configured access list. ADTRAN OS access policies are used to permit, deny, or manipulate (using NAT) data for each physical interface. Each ACP consists of a selector (access list) and an action (allow, discard, NAT). When packets are received on an interface, the configured ACPs are applied to determine whether the data will be processed or discarded. Possible actions performed by the access policy are as follows:

allow list <*access list names*>
All packets passed by the access list(s) entered will be allowed to enter the router system.

discard list <*access list names*>
All packets passed by the access list(s) entered will be dropped from the router system.

allow list <*access list names*> **dest-policy** <*access policy name*>
All packets passed by the access list(s) entered and destined for the interface using the access policy listed will be permitted to enter the router system. This allows for configurations to permit packets to a single interface and not the entire system.

discard list <*access list names*> **dest-policy** <*access policy name*>
All packets passed by the access list(s) entered and destined for the interface using the access policy listed will be blocked from the router system. This allows for configurations to deny packets on a specified interface.

Technology Review (Continued)

nat source list <*access list names*> **address** <*IP address*> **overload**
All packets passed by the access list(s) entered will be modified to replace the source IP address with the entered IP address. The **overload** keyword allows multiple source IP addresses to be replaced with the single IP address entered. This hides private IP addresses from outside the local network. This function is also known as “many-to-one NAT”.

nat source list <*access list names*> **interface** <*interface*> **overload**
All packets passed by the access list(s) entered will be modified to replace the source IP address with the primary IP address of the listed interface. The **overload** keyword allows multiple source IP addresses to be replaced with the single IP address of the specified interface. This hides private IP addresses from outside the local network. This function is also known as “many-to-one NAT”.

nat destination list <*access list names*> **address** <*IP address*>
All packets passed by the access list(s) entered will be modified to replace the destination IP address with the entered IP address. The **overload** keyword is not an option when performing NAT on the destination IP address; each private address must have a unique public address. This hides private IP addresses from outside the local network. This function is also known as “port forwarding”.

Step 4:

Apply the created access policy to an interface. To assign an access policy to an interface, enter the Interface Configuration Mode for the desired interface and enter **access policy** <*policy name*>. The following example assigns access policy **MatchAll** to the virtual PPP interface labeled 1:

(config)# **interface ppp 1**

(config-ppp 1)# **access-policy MatchAll**

alias link<*"text"*>

Each configured PPP interface (when referenced using SNMP) contains a link (physical port) and a bundle (group of links). RFC 1471 (for Link Connection Protocol) provides an interface table to manage lists of bundles and associated links. The **alias link** command provides the management station an identifying description for each link (PPP physical).

Syntax Description

<*"text"*>	Alphanumeric character string describing the interface (for SNMP) — must be encased in quotation marks

Default Values

<*"text"*>	"" (EMPTY)

Command Modes

(config-ppp 1)#	PPP Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

The **alias link** string should be used to uniquely identify a PPP link. Enter a string that clearly identifies the link.

Usage Examples

The following example defines a unique character string for the virtual PPP interface (1):

(config)# **interface ppp 1**
(config-ppp 1)# **alias link "PPP_link_1"**

Technology Review

Please refer to RFC 1990 for a more detailed discussion on PPP links and bundles.

bandwidth <*value*>

Use the **bandwidth** command to provide the bandwidth value of an interface to the higher-level protocols. This value is used in cost calculations. Use the **no** form of this command to restore the default values.

Syntax Description

<*value*>	Enter bandwidth in kbps.

Default Values

*To view default values, use the **show interfaces** command.*

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), Frame Relay Virtual Sub-interfaces (fr 1.20), virtual PPP (ppp 1), and loopback interfaces

Command History

Release 3.1	Command was introduced

Functional Notes

The **bandwidth** command is an informational value that is communicated to the higher-level protocols to be used in cost calculations. This is a routing parameter only and does not affect the physical interface.

Usage Examples

The following example sets bandwidth of the PPP interface to 10 Mbps:

(config)# **interface ppp 1**
(config-ppp 1)# **bandwidth 10000**

bridge-group *<group#>*

Use the **bridge-group** command to assign an interface to the specified bridge group. This command is supported on all Ethernet interfaces, PPP virtual interfaces, and frame relay virtual sub-interfaces.

Syntax Description

<group#>	Bridge group number (1 to 255) specified using the **bridge-group** command

Default Values

By default, there are no configured bridge groups.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), and virtual frame relay sub-interfaces (fr 1.20).

Command History

Release 1.1	Command was introduced

Functional Notes

A bridged network can provide excellent traffic management to reduce collisions and limit the amount of bandwidth wasted with unnecessary transmissions when routing is not necessary. Any two interfaces can be bridged (Ethernet to T1 bridge, Ethernet to frame relay sub-interface, etc.).

Usage Examples

The following example assigns the PPP interface to bridge-group 1:

(config)# **interface ppp 1**
(config-ppp 1)# **bridge-group 1**

bridge-group <*group#*> spanning-disabled

Use the **bridge-group spanning-disabled** command to transparently bridge two interfaces on a network (that have no parallel or redundant paths) without the overhead of spanning-tree protocol calculations. To enable the spanning-tree protocol on an interface, use the **no** form of this command.

Syntax Description

<*group#*>	Bridge group number (1 to 255) specified using the **bridge-group** command

Default Values

By default, spanning-tree protocol is enabled on all created bridge groups.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), and virtual frame relay interfaces (fr 1).

Command History

Release 1.1	Command was introduced

Functional Notes

When no parallel (redundant) paths exist within a bridged network, disabling the spanning tree protocol reduces traffic on the bridged interface. This traffic reduction can be helpful when bridging over a WAN link.

Before disabling the spanning-tree protocol on a bridged interface, verify that no redundant loops exist.

Usage Examples

The following example disables the spanning-tree protocol for bridge group 17 on the PPP interface labeled 1:

(config)# **interface ppp 1**
(config-ppp 1)# **bridge-group 17 spanning-disabled**

Technology Review

Spanning-tree protocol provides a way to prevent loopback or parallel paths in bridged networks. Using the priority values and path costs assigned to each bridging interface, the spanning-tree protocol determines the root path and identifies whether to block or allow other paths.

crypto map *<mapname>*

Use the **crypto map** command to associate crypto maps with the interface.

When you apply a map to an interface, you are applying all crypto maps with the given map name. This allows you to apply multiple crypto maps if you have created maps which share the same name but have different map index numbers.

For VPN configuration example scripts, refer to the technical support note ***Configuring VPN*** *located on the* ***NetVanta 3000 Series System Manual*** *CD provided with your unit.*

Syntax Description

<mapname>	Enter the crypto map name that you wish to assign to the interface.

Default Values

By default, no crypto maps are assigned to an interface.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), virtual frame relay sub-interfaces (fr 1.20), and loopback interfaces

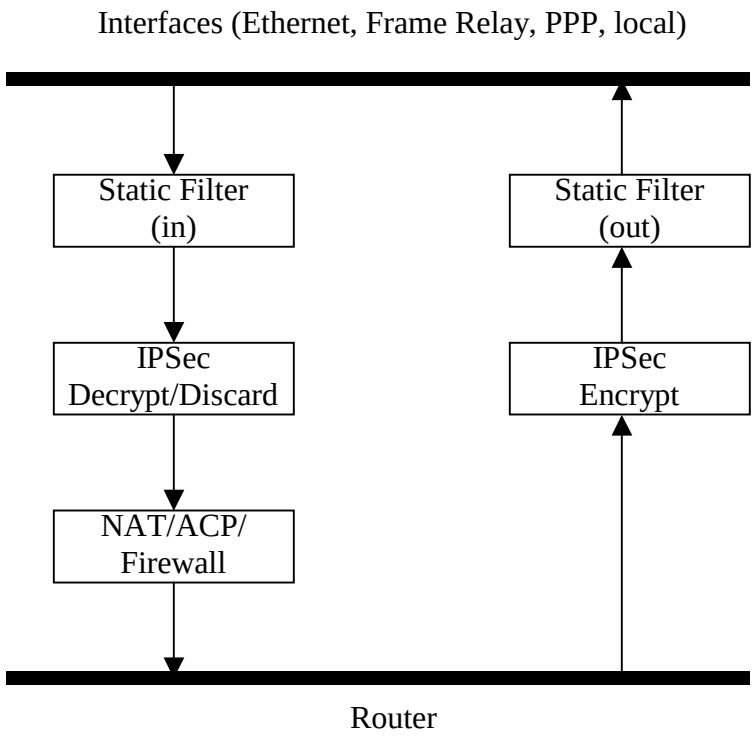
Command History

Release 4.1	Command was introduced

Functional Notes

When configuring a system to use both the stateful inspection firewall and IKE negotiation for VPN, keep the following notes in mind.

When defining the policy-class and associated access-control lists (ACLs) that describe the behavior of the firewall, do not forget to include the traffic coming into the system over a VPN tunnel terminated by the system. The firewall should be set up with respect to the un-encrypted traffic that is destined to be sent or received over the VPN tunnel. The following diagram represents typical AOS data-flow logic.



As shown in the diagram above, data coming into the product is first processed by the static filter associated with the interface on which the data is received. This access-group is a true static filter and is available for use regardless of whether the firewall is enabled or disabled. Next (if the data is encrypted) it is sent to the IPSec engine for decryption. The decrypted data is then processed by the stateful inspection firewall. Therefore, given a terminating VPN tunnel, only un-encrypted data is processed by the firewall.

The ACLs for a crypto map on an interface work in reverse logic to the ACLs for a policy-class on an interface. When specifying the ACLs for a crypto map, the source information is the private local-side, un-encrypted source of the data. The destination information will be the far-end, un-encrypted destination of the data. However, ACLs for a policy-class work in reverse. The source information for the ACL in a policy-class is the far-end. The destination information is the local-side.

Usage Examples

The following example applies all crypto maps with the name **MyMap** to the PPP interface:

(config-ppp 1)# **crypto map MyMap**

ip access-group <*listname*> [**in | out**]

Use the **ip access-group** command to create an access list to be used for packets transmitted to/from the asynchronous host. Use the **no** form of this command to disable this type of control.

Syntax Description

listname	Assigned IP access list name.
in	Enables access control on packets *transmitted from* the host.
out	Enables access control on packets *sent to* the host.

Default Values

By default, these commands are disabled.

Command Modes

(config-interface)#	Interface Configuration Mode required.

Command History

Release 3.1	Command was introduced

Functional Notes

When this command is enabled, the IP destination address of each packet must be validated before being passed through. If the packet is not acceptable per these settings, it is dropped.

Usage Examples

The following example sets up the router to only allow Telnet traffic into the PPP interface:

(config)# **ip access-list extended TelnetOnly**
(config-ext-nacl)#**permit tcp any any eq telnet**
(config-ext-nacl)#**int ppp 1**
(config-ppp 1)#**ip access-group TelnetOnly in**

ip address <*address*> <*mask*> secondary

Use the **ip address** command to define an IP address on the specified interface. Use the optional **secondary** keyword to define a secondary IP address. Use the **no** form of this command to remove a configured IP address.

Syntax Description

<*address*>	Defines the IP address for the interface in dotted decimal notation (for example: 192.22.73.101).
<*mask*>	Specifies the subnet mask that corresponds to the listed IP address.
secondary **Optional*	Optional keyword used to configure a secondary IP address for the specified interface.

Default Values

By default, there are no assigned IP addresses.

Command Modes

(config-interface)#	Interface Configuration Mode required (applies to all physical interfaces as well as virtual interfaces)

Command History

Release 1.1	Command was introduced

Functional Notes

Use secondary IP addresses to allow dual subnets on a single interface (when you need more IP addresses than the primary subnet can provide). When using secondary IP addresses, avoid routing loops by verifying that all devices on the network segment are configured with secondary IP addresses on the secondary subnet.

Usage Examples

The following example configures a secondary IP address of **192.22.72.101/30**:

(config)# **interface ppp 1**
(config-ppp 1)# **ip address 192.22.72.101 255.255.255.252 secondary**

ip helper-address *<address>*

Use the **ip helper-address** command to configure the ADTRAN OS to forward User Datagram Protocol (UDP) broadcast packets received on the interface. Use the **no** form of this command to disable forwarding packets.

> NOTE *The **ip helper** command must be used in conjunction with the **ip forward-protocol** command to configure the ADTRAN OS to forward UDP broadcast packets. See ip forward-protocol udp <port number> on page 200 for more information.*

Syntax Description

<address>	Specifies the destination IP address (in dotted decimal notation) for the forwarded UDP packets.

Default Values

By default, broadcast UDP packets are not forwarded.

Command Modes

(config-interface)#	Interface Configuration Mode required (applies to all physical interfaces as well as virtual interfaces)

Command History

Release 1.1	Command was introduced

Functional Notes

When used in conjunction with the **ip forward-protocol** command, the **ip helper-address** feature allows you to customize which broadcast packets are forwarded.

To implement the helper address feature, assign a helper-address(es) (specifying the device that needs to receive the broadcast traffic) to the interface closest to the host that transmits the broadcast packets. When broadcast packets (of the specified type forwarded using the **ip forward-protocol** command) are received on the interface, they will be forwarded to the device that needs the information.

Only packets meeting the following criteria are considered eligible by the **ip helper-address** feature:
1. The packet IP protocol is User Datagram Protocol (UDP).
2. Any UDP port specified using the **ip forward-protocol** command.
3. The media access control (MAC) address of the frame is an all-ones broadcast address (ffff.ffff.ffff).
4. The destination IP address is broadcast defined by all ones (255.255.255.255) or a subnet broadcast (for example, 192.33.4.251 for the 192.33.4.248/30 subnet).

Usage Examples

The following example forwards all DNS broadcast traffic to the DNS server with IP address 192.33.5.99:

(config)# **ip forward-protocol udp domain**
(config)# **interface ppp 1**
(config-ppp 1)# **ip helper-address 192.33.5.99**

ip ospf

Use the **ip ospf** command to customize OSPF settings (if needed).

Syntax Description

authentication-key <*password*>	Assign a simple-text authentication password to be used by other routers using the OSPF simple password authentication.
cost <*value*>	Specify the OSPF cost of sending a packet on the interface. This value overrides any computed cost value. Range: 1-65535.
dead-interval <*seconds*>	Set the maximum interval allowed between hello packets. If the maximum is exceeded, neighboring devices will determine that the device is down. Range: 0-32767.
hello-interval <*seconds*>	Specify the interval between hello packets sent on the interface. Range: 0-32767.
message-digest-key <*keyid*> **md5** <*key*>	Configure OSPF Message Digest 5 (MD5) authentication (16-byte max) keys.
priority <*value*>	Set the OSPF priority. The value set in this field helps determine the designated router for this network. Range: 0-255.
retransmit-interval <*seconds*>	Specify the time between link-state advertisements (LSAs). Range: 0-32767.
transmit-delay <*seconds*>	Set the estimated time required to send an LSA on the interface. Range: 0-32767.

Default Values

retransmit-interval <*seconds*>	5 seconds
transmit-delay <*seconds*>	1 second
hello-interval <*seconds*>	10 seconds: Ethernet, point-to-point, frame relay, and ppp
dead-interval <*seconds*>	40 seconds

Command Modes

(config-interface)#	Valid interfaces include: Ethernet (eth 0/1), virtual frame relay (fr 1), and virtual PPP (ppp 1).

Command History

Release 3.1	Command was introduced

ip ospf authentication [message-digest | null]

Use the **ip ospf authentication** command to authenticate an interface that is performing OSPF authentication.

Syntax Description

message-digest **Optional*	Select message-digest authentication type.
null **Optional*	Select for no authentication to be used.

Default Values

By default, this is set to null (meaning no authentication is used).

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), virtual frame relay sub-interfaces (fr 1.20), and loopback interfaces

Command History

Release 3.1	Command was introduced

Usage Examples

The following example specifies that no authentication will be used on the PPP interface:

(config-ppp 1)# **ip ospf authentication null**

ip ospf network [broadcast | point-to-point]

Use the **ip ospf network** command to specify the type of network on this interface.

Syntax Description

broadcast	Set the network type for broadcast.
point-to-point	Set the network type for point-to-point.

Default Values

By default, Ethernet defaults to broadcast. PPP and frame relay default to point-to-point.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), virtual frame relay sub-interfaces (fr 1.20), and loopback interfaces

Command History

Release 3.1	Command was introduced

Functional Notes

A point-to-point network will not elect designated routers.

Usage Examples

The following example designates a broadcast network type:

(config-ppp 1)# **ip ospf network broadcast**

ip proxy-arp

Use the **ip proxy-arp** to enable proxy Address Resolution Protocol (ARP) on the interface. Use the **no** form of this command to disable this feature.

Syntax Description

<*address*>	Defines the IP address for the interface in dotted decimal notation (for example: 192.22.73.101)
<*subnet mask*>	Specifies the subnet mask that corresponds to the listed IP address

Default Values

By default, proxy-arp is enabled.

Command Modes

(config-interface)#	Interface Configuration Mode required (applies to all physical interfaces as well as virtual interfaces)

Command History

Release 1.1	Command was introduced

Functional Notes

In general, the principle of proxy-arp allows a router to insert its IP address in the source IP address field of a packet (if the packet is from a host on one of its subnetworks). This allows hosts to reach devices on other subnetworks without implementing routing or specifying a default gateway.

If proxy-arp is enabled, the ADTRAN OS will respond to all proxy-arp requests with its specified MAC address and forward packets accordingly.

Enabling proxy-arp on an interface may introduce unnecessary ARP traffic on the network.

Usage Examples

The following enables proxy-arp on the virtual PPP interface:

(config)#**interface ppp 1**
(config-ppp 1)# **ip proxy-arp**

ip rip receive version *<version>*

Use the **ip rip receive version** command to configure the RIP version the unit accepts in all RIP packets received on the interface. Use the **no** form of this command to restore the default value.

Syntax Description

<version>	Specifies the RIP version
1	Only accept received RIP version 1 packets on the interface
2	Only accept received RIP version 2 packets on the interface

Default Values

*By default, all interfaces implement RIP version 1 (the default value for the **version** command).*

Command Modes

(config-interface)#	Interface Configuration Mode required (applies to all physical interfaces as well as virtual interfaces)

Command History

Release 1.1	Command was introduced

Functional Notes

Use the **ip rip receive version** to specify a RIP version that overrides the **version** (in the Router RIP command set) configuration.

The ADTRAN OS only accepts one version (either 1 or 2) on a given interface.

Usage Examples

The following example configures the virtual PPP interface to accept only RIP version 2 packets:

(config)# **interface ppp 1**
(config-ppp 1)# **ip rip receive version 2**

ip rip send version *<version>*

Use the **ip rip send version** command to configure the RIP version the unit sends in all RIP packets transmitted on the interface. Use the **no** form of this command to restore the default value.

Syntax Description

<version>	Specifies the RIP version
1	Only transmits RIP version 1 packets on the interface
2	Only transmits RIP version 2 packets on the interface

Default Values

*By default, all interfaces transmit RIP version 1 (the default value for the **version** command)*

Command Modes

(config-interface)#	Interface Configuration Mode required (applies to all physical interfaces as well as virtual interfaces)

Command History

Release 1.1	Command was introduced

Functional Notes

Use the **ip rip send version** to specify a RIP version that overrides the **version** (in the Router RIP command set) configuration.

The ADTRAN OS only transmits one version (either 1 or 2) on a given interface.

Usage Examples

The following example configures the virtual PPP interface to transmit only RIP version 2 packets:

```
(config)# interface ppp 1
(config-ppp 1)# ip rip send version 2
```

ip route-cache

Use the **ip route-cache** command to enable fast-cache switching on the interface. Use the **no** form of this command to disable fast-cache switching and return to process switching mode.

*Using Network Address Translation (NAT) or the ADTRAN OS firewall capabilities on an interface requires process switching mode (using the **no ip route-cache** command).*

Syntax Description

No subcommands.

Default Values

By default, fast-cache switching is enabled on all Ethernet and virtual frame relay sub-interfaces. IP route-cache is disabled for all virtual PPP interfaces.

Command Modes

(config-interface)#	Interface Configuration Mode required Valid interfaces include: Ethernet (eth 0/1), virtual frame relay sub-interfaces (fr 1.16), and virtual PPP interfaces (ppp 1).

Command History

Release 2.1	Command was introduced

Functional Notes

Fast switching allows an IP interface to provide optimum performance when processing IP traffic.

Usage Examples

The following example enables fast switching on the virtual PPP interface:

```
(config)# interface ppp 1
(config-ppp 1)# ip route-cache
```


ip unnumbered *<interface>*

Use the **ip unnumbered** command to use the IP address assigned to the specified interface for all IP processing on the active interface. Use the **no** form of this command to remove the unnumbered configuration.

Syntax Description

<interface>	Specifies the interface (in the format **type slot/port**) that contains the IP address to use as the source address for all packets transmitted on this interface. Valid interfaces include: Ethernet (eth 0/1), virtual frame relay sub-interfaces (fr 1.16), virtual PPP (ppp 1), and loopback interfaces.

Default Values

*By default, all interfaces are configured to use a specified IP address (using the **ip address** command).*

Command Modes

(config-interface)#	Interface Configuration Mode required Valid interfaces include: Ethernet (eth 0/1), virtual frame relay sub-interfaces (fr 1.16), and virtual PPP interfaces (ppp 1).

Command History

Release 1.1	Command was introduced

Functional Notes

If **ip unnumbered** is enabled on an interface, all IP traffic from the interface will use a source IP address taken from the specified interface. For example, specifying **ip unnumbered eth 0/1** while in the PPP Interface Configuration Mode configures the PPP interface to use the IP address assigned to the Ethernet interface for all IP processing. In addition, the ADTRAN OS uses the specified interface information when sending route updates over the unnumbered interface. Static routes may either use the interface name (ppp 1) or the far-end address (if it will be discovered).

Usage Examples

The following example configures the PPP interface (labeled **ppp 1**) to use the IP address assigned to the Ethernet interface (**eth 0/1**):

```
(config)# interface ppp 1
(config-ppp 1)# ip unnumbered eth 0/1
```

keepalive <*seconds*>

Use the **keepalive** command to enable the transmission of keepalive packets on the interface and specify the time interval in seconds between transmitted packets.

Syntax Description

<*seconds*>	Defines the time interval (in seconds) between transmitted keepalive packets (valid range: 0 to 32767 seconds)

Default Values

<*seconds*>	10 seconds

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1) and virtual PPP interfaces (ppp 1)

Command History

Release 1.1	Command was introduced

Functional Notes

If three keepalive packets are sent to an interface with no response, the interface is considered down. To detect interface failures quickly, specify a smaller keepalive time.

Usage Examples

The following example specifies a keepalive time of 5 seconds on the virtual PPP interface:

(config)# **interface ppp 1**
(config-ppp 1)# **keepalive 5**

mtu <*size*>

Use the **mtu** command to configure the maximum transmit unit (MTU) size for the active interface. Use the **no** form of this command to return to the default value.

Syntax Description

<*size*>	Configures the window size for transmitted packets. The valid ranges for the various interfaces are listed below:	
	Ethernet (eth 0/1)	64 to 1500
	virtual frame relay sub-interfaces (fr 1.16)	64 to 1520
	virtual PPP interfaces (ppp 1)	64 to 1500
	loopback interfaces	64 to 1500

Default Values

<*size*>	The default values for the various interfaces are listed below:	
	Ethernet (eth 0/1)	1500
	virtual frame relay sub-interfaces (fr 1.16)	1500
	virtual PPP interfaces (ppp 1)	1500
	loopback interfaces	1500

Command Modes

(config-interface)#	Interface Configuration Mode required (applies only to IP interfaces) Valid interfaces include: Ethernet (eth 0/1), virtual frame relay sub-interfaces (fr 1.16), virtual PPP interfaces (ppp 1), and loopback interfaces.

Command History

Release 1.1	Command was introduced

Functional Notes

OSPF will not become adjacent on links where the MTU sizes do not match. If router A and router B are exchanging hello packets but their MTU sizes do not match, they will never reach adjacency. This is by design and required by the RFC.

Usage Examples

The following example specifies an MTU of 1200 on the virtual PPP interface:

```
(config)# interface ppp 1
(config-ppp 1)# mtu 1200
```

peer default ip address <*address*>

Use the **peer default ip address** command to specify the default IP address of the remote end of this interface.

Syntax Description

<*address*>	Specifies the default IP address for the remote end (*A.B.C.D*).

Default Values

By default, there is no assigned peer default IP address.

Command Modes

(config-ppp 1)#	PPP Interface Configuration Mode required

Command History

Release 3.1	Command was introduced

Functional Notes

This command is useful if the peer does not send the IP address option during PPP negotiations.

Usage Examples

The following example sets the default peer IP address to 192.22.71.50:

(config)#**interface ppp 1**
(config-ppp 1)# **peer default ip address 192.22.71.50**

ppp authentication <*protocol*>

Use the **ppp authentication** command to specify the authentication protocol on the PPP virtual interface that the peer should use to authenticate itself.

Syntax Description

<*protocol* >	Specifies the authentication protocol used on this interface
chap	Configures CHAP authentication on the interface
eap	Configures EAP authentication on the interface
pap	Configures PAP authentication on the interface

Default Values

By default, PPP endpoints have no authentication configured.

Command Modes

(config-ppp 1)#	PPP Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Technology Review

CHAP and PAP are two authentication methods that enjoy widespread support. Both methods are included in the ADTRAN OS and are easily configured.

The authentication method set up on the local router can be different from that on the peer. Also, just because one router requires authentication from its peer does not mean it also has to authenticate itself to the peer.

Defining PAP

The Password Authentication Protocol (PAP) is used to verify that the PPP peer is a permitted device by checking a username and password configured on the peer. The username and password are both sent unencrypted across the connecting private circuit.

PAP requires two-way message passing. First, the router that is required to be authenticated (say the peer) sends an authentication request with its username and password to the router requiring authentication (say the local router). The local router then looks up the username and password in the username database within the PPP interface, and if they match sends an authentication acknowledge back to the peer.

The PPP username and password database is separate and distinct from the global username password database. For PAP and CHAP, use the database under the PPP interface configuration.

Technology Review (Continued)

Several example scenarios are given below for clarity.

Configuring PAP Example 1: Only the local router requires the peer to authenticate itself.
On the local router (hostname Local):
Local(config-ppp 1)# **ppp authentication pap**
Local(config-ppp 1)# **username farend password same**

On the peer (hostname Peer):
Peer(config-ppp 1)# **ppp pap sent-username farend password same**

The first line of the configuration sets the authentication mode as PAP. This means the peer is required to authenticate itself to the local router via PAP. The second line is the username and password expected to be sent from the peer. On the peer, the **ppp pap sent-username** command is used to specify the appropriate matching username and password.

Configuring PAP Example 2: Both routers require the peer to authenticate itself.
On the local router (hostname Local):
Local(config-ppp 1)# **ppp authentication pap**
Local(config-ppp 1)# **username farend password far**
Local(config-ppp 1)# **ppp pap sent-username nearend password near**

On the peer (hostname Peer):
Peer(config-ppp 1)# **ppp authentication pap**
Peer(config-ppp 1)# **username nearend password near**
Peer(config-ppp 1)# **ppp pap sent-username farend password far**

Now both routers send the authentication request, verify that the sent-username and password match what is expected in the database, and send an authentication acknowledge.

Defining CHAP
The Challenge-Handshake Authentication Protocol (CHAP) is a three-way authentication protocol composed of a challenge response and success or failure. The MD5 protocol is used to protect usernames and passwords in the response.

First, the local router (requiring its peer to be authenticated) sends a "challenge" containing only its own unencrypted username to the peer. The peer then looks up the username in the username database within the PPP interface, and if found takes the corresponding password and its own hostname and sends a "response" back to the local router. This data is encrypted. The local router verifies that the username and password are in its own username database within the PPP interface, and if so sends a "success" back to the peer.

The PPP username and password database is separate and distinct from the global username password database. For PAP and CHAP, use the database under the PPP interface configuration.

Technology Review (Continued)

Several example scenarios are given below for clarity.

Configuring CHAP Example 1: Only the local router requires the peer to authenticate itself.
On the local router (hostname Local):
Local(config-ppp 1)# **ppp authentication chap**
Local(config-ppp 1)# **username Peer password same**

On the peer (hostname Peer):
Peer(config-ppp 1)# **username Local password same**

The first line of this configuration sets the authentication mode to CHAP. This means the peer is required to authenticate itself to the local router via CHAP. The second line is the username and password expected to be sent from the peer. The peer must also have the **username** command set up both to verify the incoming username from the local router and to use the password (along with its hostname) in the response to the local router.

Both ends must have identical passwords.

Configuring CHAP Example 2: Both routers require the peer to authenticate itself.
On the local router (hostname Local):
Local(config-ppp 1)# **ppp authentication chap**
Local(config-ppp 1)# **username Peer password same**

On the peer (hostname Peer):
Peer(config-ppp 1)# **ppp authentication chap**
Peer(config-ppp 1)# **username Local password same**

This is basically identical to Example 1 except that both routers will now challenge each other and respond.

Technology Review (Continued)

Configuring CHAP Example 3: Using the ppp chap hostname command as an alternate solution.
On the local router (hostname Local):
Local(config-ppp 1)# **ppp authentication chap**
Local(config-ppp 1)# **username Peer password same**
Local(config-ppp 1)# **ppp chap hostname nearend**

On the peer (hostname Peer):
Peer(config-ppp 1)# **username nearend password same**

Notice the peer is expecting username “nearend” even though the local router's hostname is “Local". Therefore the local router can use the **ppp chap hostname** command to send the correct name on the challenge.

Configuring CHAP Example 4: Using the ppp chap password command as an alternate solution.
On the local router (hostname Local):
Local(config-ppp 1)# **ppp authentication chap**
Local(config-ppp 1)# **username Peer password different**

On the peer (hostname Peer):
Peer(config-ppp 1)# **username Local password same**
Peer(config-ppp 1)# **ppp chap password different**

Here the local router challenges with hostname "Local". The peer verifies the name in the username database, but instead of sending the password "same" in the response, it uses the one in the **ppp chap password** command. The local router then verifies that user "Peer" with password "different" is valid and sends a "success".

ppp chap hostname <*hostname*>

Use the **ppp chap hostname** command to configure an alternate hostname for CHAP PPP authentication. Use the **no** form of this command to remove a configured hostname. For more information on PAP and CHAP functionality, see the **Technology Review** section for the command *ppp authentication <protocol>* on page 485.

Syntax Description

<*hostname*>	Alphanumeric string up to 30 characters in length

Default Values

By default, there are no configured PPP CHAP hostnames.

Command Modes

(config-ppp 1)#	PPP Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example specifies a PPP CHAP hostname of **my_host**:

```
(config)# interface ppp 1
(config-ppp 1)# ppp chap hostname my_host
```

ppp chap password <*password*>

Use the **ppp chap password** command to configure an alternate password when the peer requires CHAP PPP authentication. Use the **no** form of this command to remove a configured password. For more information on PAP and CHAP functionality, see the **Technology Review** section for the command *ppp authentication <protocol>* on page 485.

Syntax Description

<*password*>	Alphanumeric string up to 30 characters in length

Default Values

By default, there is no defined PPP CHAP password.

Command Modes

(config-ppp 1)#	PPP Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example specifies a PPP CHAP password of **my_password**:

```
(config)# interface ppp 1
(config-ppp 1)# ppp chap password my_password
```

ppp pap sent-username <*username*> password <*password*>

Use the **ppp pap sent-username/password** command to configure a username and password when the peer requires PAP PPP authentication. Use the **no** form of this command to remove a configured password. For more information on PAP and CHAP functionality, see the **Technology Review** section for the command *ppp authentication <protocol>* on page 485.

Syntax Description

<*username*>	Alphanumeric string up to 30 characters in length (the username is case-sensitive)
<*password*>	Alphanumeric string up to 30 characters in length (the password is case-sensitive)

Default Values

By default, there is no defined ppp pap sent-username and password.

Command Modes

(config-ppp 1)#	PPP Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example specifies a PPP PAP sent-username of **local** and a password of **my_password**:

```
(config)# interface ppp 1
(config-ppp 1)# ppp pap sent-username local password my_password
```

snmp trap link-status

Use the **snmp trap link-status** command to control the SNMP variable that enables (or disables) the interface to send SNMP traps when there is an interface status change (ifLinkUpDownTrapEnable of RFC 2863). Use the **no** form of this command to disable this trap.

Syntax Description

No subcommands

Default Values

By default, the ifLinkUpDownTrapEnable OID is set to enabled for all interfaces except virtual frame relay interfaces.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), T1 (t1 1/1), DSX-1 (t1 1/2), serial (ser 1/1), DDS (dds 1/1), virtual frame relay (fr 1), virtual PPP (ppp 1), and SHDSL (shdsl 1/1) interfaces.

Command History

Release 1.1	Command was introduced

Functional Notes

The **snmp trap link-status** command is used to control the RFC 2863 ifLinkUpDownTrapEnable OID (OID number 1.3.6.1.2.1.31.1.1.1.14.0).

Usage Examples

The following example disables the link-status trap on the virtual PPP interface:

(config)# **interface ppp 1**
(config-ppp 1)# **no snmp trap link-status**

username *<username>* **password** *<password>*

Configures the username and password of the peer to use for PPP authentication.

Syntax Description

<username>	Alphanumerical string up to 30 characters in length (the username is case-sensitive)
<password>	Alphanumerical string up to 30 characters in length (the password is case-sensitive)

Default Values

By default, there is no established username and password.

Command Modes

(config-ppp 1)#	Interface Configuration Mode (valid only for virtual PPP interfaces)

Command History

Release 1.1	Command was introduced

Functional Notes

PAP uses this entry to check received information from the peer. CHAP uses this entry to check the received peer hostname and a common password.

Usage Examples

The following example creates a username of **ADTRAN** with password **ADTRAN** for the PPP link labeled 5:

(config)# **interface ppp 5**
(config-ppp 5)# **username ADTRAN password ADTRAN**

LOOPBACK INTERFACE CONFIGURATION COMMAND SET

To activate the Loopback Interface Configuration command set, enter the **interface loopback** command at the Global Configuration Mode prompt. For example:

```
Router> enable
Router# configure terminal
Router(config)# interface loopback 1
Router(config-loop 1)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

alias <"text"> on page 556

description on page 557

do on page 558

end on page 559

exit on page 560

shutdown on page 561

All other commands for this command set are described in this section in alphabetical order.

access-policy <policyname> on page 495

bandwidth <value> on page 498

crypto map <mapname> on page 499

ip commands begin on page 501

loopback remote inband on page 513

mtu <size> on page 514

snmp trap link-status on page 515

access-policy *<policyname>*

Use the **access-policy** command to assign a specified access policy to an interface. Use the **no** form of this command to remove an access policy association.

Syntax Description

<policyname>	Alphanumeric descriptor for identifying the configured access policy (all access policy descriptors are case-sensitive).

Default Values

By default, there are no configured access policies associated with an interface.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), virtual frame relay sub-interfaces (fr 1.20), and loopback interfaces

Command History

Release 2.1	Command was introduced

Functional Notes

To assign an access policy to an interface, enter the interface configuration mode for the desired interface and enter **access policy** <*policy name*>.

Usage Examples

The following example associates the access policy **UnTrusted** (to allow inbound traffic to the Web server) to the loopback interface:

Enable the ADTRAN OS security features:
(config)# **ip firewall**

Create the access list (this is the packet selector):
(config)# **ip access-list extended InWeb**
(config-ext-nacl)# **permit tcp any host 63.12.5.253 eq 80**

Create the access policy that contains the access list **InWeb**:
(config)# **ip policy-class UnTrusted**
(config-policy-class)# **allow list InWeb**

Usage Examples (Continued)

Associate the access policy with the loopback interface:
(config)# **interface loopback 1**
(config-loop 1) **access-policy UnTrusted**

Technology Review

Creating access policies and lists to regulate traffic through the routed network is a four-step process:

Step 1:

Enable the security features of the ADTRAN OS using the **ip firewall** command.

Step 2:

Create an access list to permit or deny specified traffic. Standard access lists provide pattern matching for source IP addresses only. (Use extended access lists for more flexible pattern matching.) IP addresses can be expressed in one of three ways:

1. Using the keyword **any** to match any IP address. For example, entering **deny any** will effectively shut down the interface that uses the access list because all traffic will match the **any** keyword.
2. Using the **host** <A.B.C.D> to specify a single host address. For example, entering **permit host 196.173.22.253** will allow all traffic from the host with an IP address of 196.173.22.253.
3. Using the <A.B.C.D> <wildcard> format to match all IP addresses in a “range”. Wildcard masks work in reverse logic from subnet mask. Specifying a one in the wildcard mask equates to a “don’t care”. For example, entering **deny 192.168.0.0 0.0.0.255** will deny all traffic from the 192.168.0.0/24 network.

Step 3:

Create an IP policy class that uses a configured access list. ADTRAN OS access policies are used to permit, deny, or manipulate (using NAT) data for each physical interface. Each ACP consists of a selector (access list) and an action (allow, discard, NAT). When packets are received on an interface, the configured ACPs are applied to determine whether the data will be processed or discarded. Possible actions performed by the access policy are as follows:

allow list <*access list names*>
All packets passed by the access list(s) entered will be allowed to enter the router system.

discard list <*access list names*>
All packets passed by the access list(s) entered will be dropped from the router system.

allow list <*access list names*> **dest-policy** <*access policy name*>
All packets passed by the access list(s) entered and destined for the interface using the access policy listed will be permitted to enter the router system. This allows for configurations to permit packets to a single interface and not the entire system.

allow list <*access list names*> **dest-policy** <*access policy name*>
All packets passed by the access list(s) entered and destined for the interface using the access policy listed will be blocked from the router system. This allows for configurations to deny packets on a specified interface.

Technology Review (Continued)

nat source list <*access list names*> **address** <*IP address*> **overload**
All packets passed by the access list(s) entered will be modified to replace the source IP address with the entered IP address. The **overload** keyword allows multiple source IP addresses to be replaced with the single IP address entered. This hides private IP addresses from outside the local network.

nat source list <*access list names*> **interface** <*interface*> **overload**
All packets passed by the access list(s) entered will be modified to replace the source IP address with the primary IP address of the listed interface. The **overload** keyword allows multiple source IP addresses to be replaced with the single IP address of the specified interface. This hides private IP addresses from outside the local network.

nat destination list <*access list names*> **address** <*IP address*>
All packets passed by the access list(s) entered will be modified to replace the destination IP address with the entered IP address. The **overload** keyword is not an option when performing NAT on the destination IP address; each private address must have a unique public address. This hides private IP addresses from outside the local network.

Step 4:

Apply the created access policy to an interface. To assign an access policy to an interface, enter the interface configuration mode for the desired interface and enter **access policy** <*policy name*>. The following example assigns access policy **MatchAll** to the loopback interface:

(config)# **interface loopback 1**

(config-loop 1)# **access-policy MatchAll**

bandwidth <*value*>

Use the **bandwidth** command to provide the bandwidth value of an interface to the higher-level protocols. This value is used in cost calculations. Use the **no** form of this command to restore the default values.

Syntax Description

<*value*>	Enter bandwidth in kbps.

Default Values

*To view default values, use the **show interfaces** command.*

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), Frame Relay Virtual Sub-interfaces (fr 1.20), virtual PPP (ppp 1), and loopback interfaces

Command History

Release 3.1	Command was introduced

Functional Notes

The **bandwidth** command is an informational value that is communicated to the higher-level protocols to be used in cost calculations. This is a routing parameter only and does not affect the physical interface.

Usage Examples

The following example sets bandwidth of the loopback interface to 10 Mbps:

(config)# **interface loopback 1**
(config-loop 1)# **bandwidth 10000**

crypto map <*mapname*>

Use the **crypto map** command to associate crypto maps with the interface.

When you apply a map to an interface, you are applying all crypto maps with the given map name. This allows you to apply multiple crypto maps if you have created maps which share the same name but have different map index numbers.

For VPN configuration example scripts, refer to the technical support note ***Configuring VPN*** *located on the* ***NetVanta 3000 Series System Manual*** *CD provided with your unit.*

Syntax Description

<*mapname*>	Enter the crypto map name that you wish to assign to the interface.

Default Values

By default, no crypto maps are assigned to an interface.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), virtual frame relay sub-interfaces (fr 1.20), and loopback interfaces

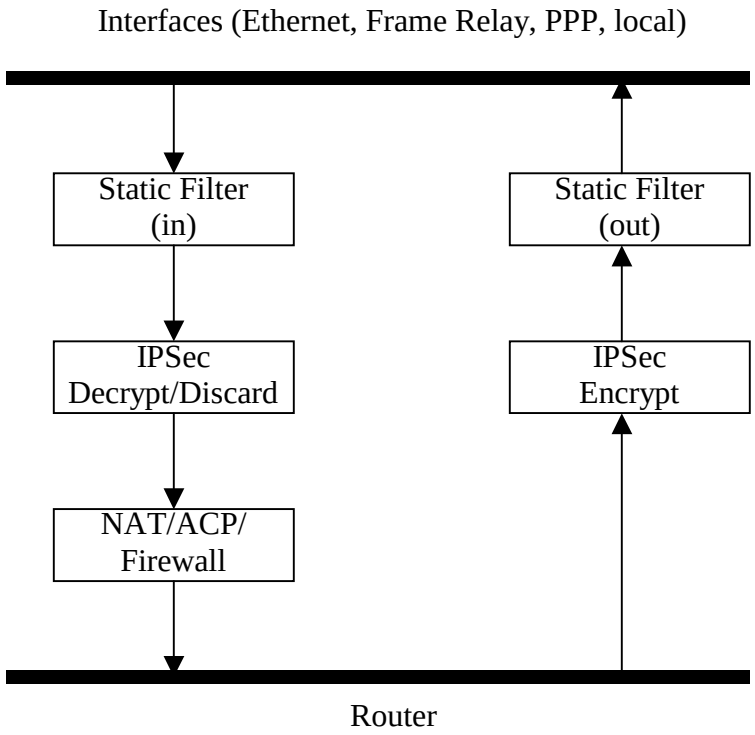
Command History

Release 4.1	Command was introduced

Functional Notes

When configuring a system to use both the stateful inspection firewall and IKE negotiation for VPN, keep the following information in mind:

When defining the policy-class and associated access-control lists (ACLs) that describe the behavior of the firewall, do not forget to include the traffic coming into the system over a VPN tunnel terminated by the system. The firewall should be set up with respect to the un-encrypted traffic that is destined to be sent or received over the VPN tunnel. The following diagram represents typical AOS data-flow logic.



As shown in the diagram above, data coming into the product is first processed by the static filter associated with the interface on which the data is received. This access-group is a true static filter and is available for use regardless of whether the firewall is enabled or disabled. Next (if the data is encrypted) it is sent to the IPSec engine for decryption. The decrypted data is then processed by the stateful inspection firewall. Therefore, given a terminating VPN tunnel, only un-encrypted data is processed by the firewall.

The ACLs for a crypto map on an interface work in reverse logic to the ACLs for a policy-class on an interface. When specifying the ACLs for a crypto map, the source information is the private local-side, un-encrypted source of the data. The destination information will be the far-end, un-encrypted destination of the data. However, ACLs for a policy-class work in reverse. The source information for the ACL in a policy-class is the far-end. The destination information is the local-side.

Usage Examples

The following example applies all crypto maps with the name **MyMap** to the loopback interface:

(config-loop 1)# **crypto map MyMap**

ip access-group <*listname*> [**in | out**]

Use the **ip access-group** command to create an access list to be used for packets transmitted to/from the asynchronous host. Use the **no** form of this command to disable this type of control.

Syntax Description

listname	Assigned IP access list name.
in	Enables access control on packets *transmitted from* the host.
out	Enables access control on packets *sent to* the host.

Default Values

By default, these commands are disabled.

Command Modes

(config-interface)#	Interface Configuration Mode required.

Command History

Release 3.1	Command was introduced

Functional Notes

When this command is enabled, the IP destination address of each packet must be validated before being passed through. If the packet is not acceptable per these settings, it is dropped.

Usage Examples

The following example sets up the router to allow only Telnet traffic into the loopback interface:

(config)# **ip access-list extended TelnetOnly**
(config-ext-nacl)#**permit tcp any any eq telnet**
(config-ext-nacl)#**interface loopback 1**
(config-loop 1)#**ip access-group TelnetOnly in**

ip address <*address*> <*mask*> secondary

Use the **ip address** command to define an IP address on the specified interface. Use the optional **secondary** keyword to define a secondary IP address. Use the **no** form of this command to remove a configured IP address.

Syntax Description

<*address*>	Defines the IP address for the interface in dotted decimal notation (for example: 192.22.73.101).
<*mask*>	Specifies the subnet mask that corresponds to the listed IP address.
secondary **Optional*	Optional keyword used to configure a secondary IP address for the specified interface.

Default Values

By default, there are no assigned IP addresses.

Command Modes

(config-interface)#	Interface Configuration Mode required (applies to all physical interfaces as well as virtual interfaces)

Command History

Release 1.1	Command was introduced
Release 2.1	Added **ip address dhcp** for DHCP client support

Functional Notes

Use secondary IP addresses to allow dual subnets on a single interface (when you need more IP addresses than the primary subnet can provide). When using secondary IP addresses, avoid routing loops by verifying that all devices on the network segment are configured with secondary IP addresses on the secondary subnet.

Usage Examples

The following example configures a secondary IP address of **192.22.72.101/30**:

(config)# **interface loopback 1**
(config-loop 1)# **ip address 192.22.72.101 255.255.255.252 secondary**

ip helper-address *<address>*

Use the **ip helper-address** command to configure the ADTRAN OS to forward User Datagram Protocol (UDP) broadcast packets received on the interface. Use the **no** form of this command to disable forwarding packets.

> NOTE
>
> *The **ip helper** command must be used in conjunction with the **ip forward-protocol** command to configure the ADTRAN OS to forward UDP broadcast packets. See ip forward-protocol udp <port number> on page 200 for more information.*

Syntax Description

<address>	Specifies the destination IP address (in dotted decimal notation) for the forwarded UDP packets.

Default Values

By default, broadcast UDP packets are not forwarded.

Command Modes

(config-interface)#	Interface Configuration Mode required (applies to all physical interfaces as well as virtual interfaces)

Command History

Release 1.1	Command was introduced

Functional Notes

When used in conjunction with the **ip forward-protocol** command, the **ip helper-address** feature allows you to customize which broadcast packets are forwarded.

To implement the helper address feature, assign a helper-address(es) (specifying the device that needs to receive the broadcast traffic) to the interface closest to the host that transmits the broadcast packets. When broadcast packets (of the specified type forwarded using the **ip forward-protocol** command) are received on the interface, they will be forwarded to the device that needs the information.

Only packets meeting the following criteria are considered eligible by the **ip helper-address** feature:
1. The packet IP protocol is User Datagram Protocol (UDP).
2. Any UDP port specified using the **ip forward-protocol** command.
3. The media access control (MAC) address of the frame is an all-ones broadcast address (ffff.ffff.ffff).
4. The destination IP address is broadcast defined by all ones (255.255.255.255) or a subnet broadcast (for example, 192.33.4.251 for the 192.33.4.248/30 subnet).

Usage Examples

The following example forwards all DNS broadcast traffic to the DNS server with IP address 192.33.5.99:

(config)# **ip forward-protocol udp domain**
(config)# **interface loopback 1**
(config-loop 1)# **ip helper-address 192.33.5.99**

ip ospf

Use the **ip ospf** command to customize OSPF settings (if needed).

Syntax Description

authentication-key <*password*>	Assign a simple-text authentication password to be used by other routers using the OSPF simple password authentication.
cost <*value*>	Specify the OSPF cost of sending a packet on the interface. This value overrides any computed cost value. Range: 1-65535.
dead-interval <*seconds*>	Set the maximum interval allowed between hello packets. If the maximum is exceeded, neighboring devices will determine that the device is down. Range: 0-32767.
hello-interval <*seconds*>	Specify the interval between hello packets sent on the interface. Range: 0-32767.
message-digest-key <*keyid*> **md5** <*key*>	Configure OSPF Message Digest 5 (MD5) authentication (16-byte max) keys.
priority <*value*>	Set the OSPF priority. The value set in this field helps determine the designated router for this network. Range: 0-255.
retransmit-interval <*seconds*>	Specify the time between link-state advertisements (LSAs). Range: 0-32767.
transmit-delay <*seconds*>	Set the estimated time required to send an LSA on the interface. Range: 0-32767.

Default Values

retransmit-interval <*seconds*>	5 seconds
transmit-delay <*seconds*>	1 second
hello-interval <*seconds*>	10 seconds: Ethernet, point-to-point, frame relay, and ppp
dead-interval <*seconds*>	40 seconds

Command Modes

(config-interface)#	Valid interfaces include: Ethernet (eth 0/1), virtual frame relay (fr 1), and virtual PPP (ppp 1).

Command History

Release 3.1	Command was introduced

ip ospf authentication [message-digest | null]

Use the **ip ospf authentication** command to authenticate an interface that is performing OSPF authentication.

Syntax Description

message-digest **Optional*	Select message-digest authentication type.
null **Optional*	Select for no authentication to be used.

Default Values

By default, this is set to null (meaning no authentication is used).

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), virtual frame relay sub-interfaces (fr 1.20), and loopback interfaces

Command History

Release 3.1	Command was introduced

Usage Examples

The following example specifies that no authentication will be used on the loopback interface:

(config-loop 1)# **ip ospf authentication null**

ip ospf network [broadcast | point-to-point]

Use the **ip ospf network** command to specify the type of network on this interface.

Syntax Description

broadcast	Set the network type for broadcast.
point-to-point	Set the network type for point-to-point.

Default Values

By default, Ethernet defaults to broadcast. PPP and frame relay default to point-to-point.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), virtual PPP interfaces (ppp 1), virtual frame relay sub-interfaces (fr 1.20), and loopback interfaces

Command History

Release 3.1	Command was introduced

Functional Notes

A point-to-point network will not elect designated routers.

Usage Examples

The following example designates a broadcast network type:

(config-loop 1)# **ip ospf network broadcast**

ip proxy-arp

Use the **ip proxy-arp** to enable proxy Address Resolution Protocol (ARP) on the interface. Use the **no** form of this command to disable this feature.

Syntax Description

<address>	Defines the IP address for the interface in dotted decimal notation (for example: 192.22.73.101).
<subnet mask>	Specifies the subnet mask that corresponds to the listed IP address.

Default Values

By default, proxy arp is enabled.

Command Modes

(config-interface)#	Interface Configuration Mode required (applies to all physical interfaces as well as virtual interfaces)

Command History

Release 1.1	Command was introduced

Functional Notes

In general, the principle of proxy-arp allows a router to insert its IP address in the source IP address field of a packet (if the packet is from a host on one of its subnetworks). This allows hosts to reach devices on other subnetworks without implementing routing or specifying a default gateway.

If proxy-arp is enabled, the ADTRAN OS will respond to all proxy-arp requests with its specified MAC address and forward packets accordingly.

Enabling proxy-arp on an interface may introduce unnecessary ARP traffic on the network.

Usage Examples

The following enables proxy-arp on the loopback interface:

(config)#**interface loopback 1**
(config-loop 1)# **ip proxy-arp**

ip rip receive version <*version*>

Use the **ip rip receive version** command to configure the RIP version the unit accepts in all RIP packets received on the interface.

Syntax Description

<*version*>	Specifies the RIP version.
1	Only accept received RIP version 1 packets on the interface.
2	Only accept received RIP version 2 packets on the interface.

Default Values

*By default, all interfaces implement RIP version 1 (the default value for the **version** command).*

Command Modes

(config-interface)#	Interface Configuration Mode required (applies to all physical interfaces as well as virtual interfaces)

Command History

Release 1.1	Command was introduced

Functional Notes

Use the **ip rip receive version** to specify a RIP version that overrides the **version** (in the Router RIP command set) configuration.

The ADTRAN OS only accepts one version (either 1 or 2) on a given interface.

Usage Examples

The following example configures the loopback interface to accept only RIP version 2 packets:

(config)# **interface loopback 1**
(config-loop 1)# **ip rip receive version 2**

ip rip send version *<version>*

Use the **ip rip send version** command to configure the RIP version the unit sends in all RIP packets transmitted on the interface.

Syntax Description

<version>	Specifies the RIP version.
1	Only transmits RIP version 1 packets on the interface.
2	Only transmits RIP version 2 packets on the interface.

Default Values

*By default, all interfaces transmit RIP version 1 (the default value for the **version** command).*

Command Modes

(config-interface)#	Interface Configuration Mode required (applies to all physical interfaces as well as virtual interfaces)

Command History

Release 1.1	Command was introduced

Functional Notes

Use the **ip rip send version** to specify a RIP version that overrides the **version** (in the Router RIP command set) configuration.

The ADTRAN OS only transmits one version (either 1 or 2) on a given interface.

Usage Examples

The following example configures the loopback interface to transmit only RIP version 2 packets:

(config)# **interface loopback 1**
(config-loop 1)# **ip rip send version 2**

ip route-cache

Use the **ip route-cache** command to enable fast-cache switching on the interface. Use the **no** form of this command to disable fast-cache switching and return to process switching mode.

*Using Network Address Translation (NAT) or the ADTRAN OS firewall capabilities on an interface requires process switching mode (using the **no ip route-cache** command).*

Syntax Description

No subcommands.

Default Values

By default, fast-cache switching is enabled on all Ethernet and virtual frame relay sub-interfaces. IP route-cache is disabled for all virtual PPP interfaces.

Command Modes

(config-interface)#	Interface Configuration Mode required Valid interfaces include: Ethernet (eth 0/1), virtual frame relay sub-interfaces (fr 1.16), virtual PPP interfaces (ppp 1), and loopback interfaces.

Command History

Release 2.1	Command was introduced

Functional Notes

Fast switching allows an IP interface to provide optimum performance when processing IP traffic.

Usage Examples

The following example enables fast switching on the loopback interface:

(config)# **interface loopback 1**
(config-loop 1)# **ip route-cache**

ip unnumbered *<interface>*

Use the **ip unnumbered** command to use the IP address assigned to the specified interface for all IP processing on the active interface. Use the **no** form of this command to remove the unnumbered configuration.

Syntax Description

<interface>	Specifies the interface (in the format **type slot/port**) that contains the IP address to be used as the source address for all packets transmitted on this interface Valid interfaces include: Ethernet (eth 0/1), virtual frame relay sub-interfaces (fr 1.16), virtual PPP interfaces (ppp 1), and loopback interfaces.

Default Values

*By default, all interfaces are configured to use a specified IP address (using the **ip address** command).*

Command Modes

(config-interface)#	Interface Configuration Mode required Valid interfaces include: Ethernet (eth 0/1), virtual frame relay sub-interfaces (fr 1.16), virtual PPP interfaces (ppp 1), and loopback interfaces

Command History

Release 1.1	Command was introduced

Functional Notes

If **ip unnumbered** is enabled on an interface, all IP traffic from the interface will use a source IP address taken from the specified interface. For example, specifying **ip unnumbered ppp 1** while in the Ethernet Interface Configuration Mode configures the Ethernet interface to use the IP address assigned to the PPP interface for all IP processing. In addition, the ADTRAN OS uses the specified interface information when sending route updates over the unnumbered interface.

Usage Examples

The following example configures the loopback interface (labeled **loop 1**) to use the IP address assigned to the PPP interface (**ppp 1**):

```
(config)# interface loopback 1
(config-loop 1)# ip unnumbered ppp 1
```


loopback remote inband

Use the **loopback remote inband** command to inject the selected inband loop-up pattern into the data stream to cause a loopback at the far-end. Use the **no** form of this command to inject a loop-down pattern into the data stream to cause an existing inband loopback at the far-end to cease.

Syntax Description

No subcommands

Default Values

By default, this command is enabled.

Command Modes

(config-loop 1)#	Loopback Interface Configuration Mode required

Command History

Release 4.1	Command was introduced

Usage Examples

The following example injects a loop-down pattern into the datastream, causing existing loopbacks at the far-end to stop:

(config-loop 1)# **no loopback remote inband**

mtu <*size*>

Use the **mtu** command to configure the maximum transmit unit size for the active interface. Use the **no** form of this command to return to the default value.

Syntax Description

<*size*>	Configures the window size for transmitted packets. The valid ranges for the various interfaces are listed below:	
	Ethernet (eth 0/1)	64 to 1500
	virtual frame relay sub-interfaces (fr 1.16)	64 to 1520
	virtual PPP interfaces (ppp 1)	64 to 1500
	loopback interfaces	64 to 1500

Default Values

<*size*>	The default values for the various interfaces are listed below:	
	Ethernet (eth 0/1)	1500
	virtual frame relay sub-interfaces (fr 1.16)	1500
	virtual PPP interfaces (ppp 1)	1500
	loopback interfaces	1500

Command Modes

(config-interface)#	Interface Configuration Mode required (applies only to IP interfaces) Valid interfaces include: Ethernet (eth 0/1), virtual frame relay sub-interfaces (fr 1.16), virtual PPP interfaces (ppp 1), and loopback interfaces.

Command History

Release 1.1	Command was introduced

Functional Notes

OSPF will not become adjacent on links where the MTU sizes do not match. If router A and router B are exchanging hello packets but their MTU sizes do not match, they will never reach adjacency. This is by design and required by the RFC.

Usage Examples

The following example specifies an MTU of 1200 on the loopback interface:

```
(config)# interface loopback 1
(config-loop 1)# mtu 1200
```


snmp trap link-status

Use the **snmp trap link-status** to control the SNMP variable ifLinkUpDownTrapEnable (RFC 2863) to enable (or disable) the interface to send SNMP traps when there is an interface status change. Use the **no** form of this command to disable this trap.

Syntax Description

No subcommands

Default Values

By default, the ifLinkUpDownTrapEnable OID is set to enabled for all interfaces except virtual frame relay interfaces.

Command Modes

(config-interface)#	Interface Configuration Mode Valid interfaces include: Ethernet (eth 0/1), T1 (t1 1/1), E1 (e1 1/1), DSX-1 (t1 1/2), serial (ser 1/1), DDS (dds 1/1), virtual frame relay (fr 1), virtual PPP (ppp 1), SHDSL (shdsl 1/1), and loopback interfaces.

Command History

Release 1.1	Command was introduced

Functional Notes

The **snmp trap link-status** command is used to control the RFC 2863 ifLinkUpDownTrapEnable OID (OID number 1.3.6.1.2.1.31.1.1.1.14.0).

Usage Examples

The following example disables the link-status trap on the loopback interface:

```
(config)# interface loopback 1
(config-loop 1)# no snmp trap link-status
```

LINE (CONSOLE) INTERFACE CONFIG COMMAND SET

To activate the Line (Console) Interface Configuration command set, enter the **line console 0** command at the Global Configuration Mode prompt. For example:

```
Router> enable
Router# configure terminal
Router(config)# line console 0
Router(config-con 0)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

do on page 558

end on page 559

exit on page 560

All other commands for this command set are described in this section in alphabetical order.

databits <*option*> on page 517

flowcontrol [none | software] on page 518

line-timeout <*minutes*> on page 519

login on page 520

login local-userlist on page 521

parity <*option*> on page 522

password <*password*> on page 523

speed <*rate*> on page 524

stopbits <*option*> on page 525

databits <*option*>

Use the **databits** command to set the number of databits per character for a terminal session. This value must match the configuration of your VT100 terminal or terminal emulator software. The default is 8 databits per character. Use the **no** form of this command to return to the default value.

Syntax Description

<*option*>	Specifies the number of databits per character
7	7 data bits
8	8 data bits

Default Values

<*option*>	**8**

Command Modes

(config-con 0)#	Console Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example configures 7 databits per character for the console terminal session:

(config)# **line console 0**
(config-con 0)# **databits 7**

flowcontrol [none | software]

Use the **flowcontrol** command to set flow control for the line console.

Syntax Description

none	Set no flow control.
software in	Configure AOS to derive flow control from the attached device.

Default Values

By default, flow control is set to none.

Command Modes

(config-con 0)#	Console Interface Configuration Mode required

Command History

Release 3.1	Command was introduced

Usage Examples

The following example configures no flow control for the line console:

(config)# **line console 0**
(config-con 0)# **flowcontrol none**

line-timeout <*minutes*>

Use the **line-timeout** command to specify the number of minutes a line session may remain inactive before the ADTRAN OS terminates the session. Use the **no** form of this command to return to the default value.

Syntax Description

<*minutes*>	Specifies the number of minutes a line session may remain inactive before the ADTRAN OS terminates the session Entering a **line-timeout** value of 0 disables the feature.

Default Values

<*minutes*>	**15 minutes** (Console and Telnet)

Command Modes

(config-line)#	Line Configuration Mode Valid interfaces include: Console (con 0) and Telnet (telnet X)

Command History

Release 1.1	Command was introduced

Usage Examples

The following example specifies a timeout of 2 minutes:

(config)# **line console 0**
(config-con 0)# **line-timeout 2**

login

Use the **login** command to enable security login on the line session requiring the password configured using the **password** command. Use the **no** form of this command to disable the login feature.

Syntax Description

No subcommands.

Default Values

By default, there is no login password set for access to the unit.

Command Modes

(config-line)#	Line Configuration Mode Valid interfaces include: Console (con 0) and Telnet (telnet X)

Command History

Release 1.1	Command was introduced

Usage Examples

The following example enables the security login feature and specifies a password on the available console session:

(config)# **line console 0**
(config-console 0)# **login**
(config-console 0)# **password mypassword**

login local-userlist

Use the **login local-userlist** command to enable security login for the terminal session requiring the usernames and passwords configured using the **username/password** Global Configuration command. Use the **no** form of this command to disable the login local-userlist feature.

*All user properties assigned using the **username/password** command are valid when using the **login local-userlist** command.*

Syntax Description

No subcommands.

Default Values

By default, there is no login password set for access to the unit.

Command Modes

(config-line)#	Line Interface Configuration Mode Valid interfaces include: Console (con 0) and Telnet (telnet X)

Command History

Release 1.1	Command was introduced

Usage Examples

The following example displays creating a local userlist and enabling the security login feature on the **CONSOLE** port:

(config)# **username my_user password my_password**
(config)# **line console 0**
(config-con 0)# **login local-userlist**

When connecting to the unit, the following prompts are displayed:
User Access Login
Username: **ADTRAN**
Password:
Router#

parity *<option>*

Use the **parity** command to specify the type of parity used as error correction. This value must match the configuration of your VT100 terminal or terminal emulator software. Use the **no** form of this command to return to the default value.

Syntax Description

<option>	Specifies the type of data parity on the interface
even	The parity bit is set to 0 if the number of 1 bits in the data sequence is odd, or set to 1 if the number of 1 bits is even.
mark	The parity bit is always set to 1.
none	No parity bit used.
odd	The parity bit is set to 1 if the number of 1 bits in the data sequence is even, or set to 1 if the number is odd.
space	The parity bit is always set to 0.

Default Values

<option>	**none**

Command Modes

(config-con 0)#	Console Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

Parity is the process used to detect whether characters have been altered during the data transmission process. Parity bits are appended to data frames to ensure that parity (whether it be odd or even) is maintained.

Usage Examples

The following example specifies mark parity for the console terminal session:

```
(config)# line console 0
(config-con 0)# parity mark
```


password <*password*>

Use the **password** command to configure the password required on the line session when security login is enabled (using the **login** command). Use the **no** form of this command to remove a configured password.

Syntax Description

<*password*>	Alphanumeric character string (up to 16 characters) used to specify the password for the line session

Default Values

By default, there is no login password set for access to the unit.

Command Modes

(config-line)#	Line Interface Configuration Valid interfaces include: Console (con 0) and Telnet (telnet X)

Command History

Release 1.1	Command was introduced

Usage Examples

The following example enables the security login feature and specifies a password on the **CONSOLE** port:

(config)# **line console 0**
(config-con 0)# **login**
(config-con 0)# **password mypassword**

speed *<rate>*

Use the **speed** command to specify the data rate for the **CONSOLE** port. This setting must match your VT100 terminal emulator or emulator software. Use the **no** form of this command to restore the default value.

Syntax Description

<rate>	Rate of data transfer on the interface (2400, 4800, 9600, 19200, 38400, 57600, or 115200 bps).

Default Values

<rate>	**9600 bps**

Command Modes

(config-con 0)#	Console Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example configures the **CONSOLE** port for 19200 bps:

(config)# **line console 0**
(config-con 0)# **speed 19200**

stopbits *<option>*

Use the **stopbits** command to set the number of stopbits per character for a terminal session. This value must match the configuration of your VT100 terminal or terminal emulator software. The default is 1 stopbit per character. Use the **no** form of this command to return to the default value.

Syntax Description

<option>	Specifies the number of stopbits per character
1	1 stopbit
2	2 stopbits

Default Values

<option>	**1**

Command Modes

(config-con 0)#	Console Interface Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example configures 2 stopbits per character for the console terminal session:

(config)# **line console 0**
(config-con 0)# **stopbits 2**

LINE (TELNET) INTERFACE CONFIG COMMAND SET

To activate the Line (Telnet) Interface Configuration command set, enter the **line telnet** command specifying a Telnet session(s) at the Global Configuration Mode prompt. For example:

```
Router> enable
Router# configure terminal
Router(config)# line telnet 0 4
Router(config-telnet0-4)#
```

You can select a single line by entering the **line telnet** command followed by the line number (0-4). For example:

```
Router> enable
Router# configure terminal
Router(config)# line telnet 2
Router(config-telnet2)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

do on page 558

end on page 559

exit on page 560

All other commands for this command set are described in this section in alphabetical order.

access-class <listname> in on page 527

line-timeout <minutes> on page 528

login on page 529

login local-userlist on page 530

password <password> on page 531

access-class <*listname*> in

Use the **access-class in** command to restrict Telnet access using a configured access list. Received packets passed by the access list will be allowed. Use the access list configuration to deny hosts or entire networks or to permit specified IP addresses.

Syntax Description

<*listname*>	Alphanumeric descriptor for identifying the configured access list (all access list descriptors are case-sensitive). See *ip access-list extended <listname>* on page 170 and *ip access-list standard <listname>* on page 176 for more information on creating access-control lists.

Default Values

By default, there are no configured access lists associated with Telnet sessions.

Command Modes

(config-telnet X)#	Line Configuration Mode required.

Command History

Release 1.1	Command was introduced

Functional Notes

When using the **access-class in** command to associate an access list with a Telnet session, remember to duplicate the **access-class in** command for all configured Telnet sessions 0 through 4. Telnet access to the unit using a particular Telnet session is not possible. Users will be assigned the first available Telnet session.

Usage Examples

The following example associates the access list **Trusted** (to allow Telnet sessions from the 192.22.56.0/24 network) with all Telnet sessions (0 through 4):

Create the access list:
(config)# **ip access-list standard Trusted**
(config)# **permit 192.22.56.0 0.0.0.255**

Enter the line (telnet) command set:
(config)# **line telnet 0 4**

Associate the access list with the Telnet session:
(config-telnet0-4)# **access-class Trusted in**

line-timeout <*minutes*>

Use the **line-timeout** command to specify the number of minutes a line session may remain inactive before the ADTRAN OS terminates the session. Use the **no** form of this command to return to the default value.

Syntax Description

<*minutes*>	Specifies the number of minutes a line session may remain inactive before the ADTRAN OS terminates the session. Entering a **line-timeout** value of 0 disables the feature.

Default Values

<*minutes*>	**15 minutes** (Console and Telnet)

Command Modes

(config-line)#	Line Configuration Mode Valid interfaces include: Console (con 0) and Telnet (telnet X)

Command History

Release 1.1	Command was introduced

Usage Examples

The following example specifies a timeout of 2 minutes:

(config)# **line telnet 0**
(config-telnet0)# **line-timeout 2**

login

Use the **login** command to enable security login on the line session requiring the password configured using the **password** command. Use the **no** form of this command to disable the login feature.

Syntax Description

No subcommands.

Default Values

By default, there is no login password set for access to the unit.

Command Modes

(config-line)#	Line Configuration Mode Valid interfaces include: Console (con 0) and Telnet (telnet X)

Command History

Release 1.1	Command was introduced

Usage Examples

The following example enables the security login feature and specifies a password on all the available Telnet sessions (0 through 4):

(config)# **line telnet 0 4**
(config-telnet0-4)# **login**
(config-telnet0-4)# **password mypassword**

login local-userlist

Use the **login local-userlist** command to enable security login for the terminal session requiring the usernames and passwords configured using the **username/password** Global Configuration command. Use the **no** form of this command to disable the login local-userlist feature.

*All user properties assigned using the **username/password** command are valid when using the **login local-userlist** command.*

Syntax Description

No subcommands.

Default Values

By default, there is no login password set for access to the unit.

Command Modes

(config-line)#	Line Configuration Mode Valid interfaces include: Console (con 0) and Telnet (telnet X)

Command History

Release 1.1	Command was introduced

Usage Examples

The following example displays creating a local userlist and enabling the security login feature:

(config)# **username my_user password my_password**
(config)# **line telnet 0**
(config-telnet0)# **login local-userlist**

When connecting to the unit, the following prompts are displayed:
User Access Login
Username: **my_user**
Password:
Router#

password *<password>*

Use the **password** command to configure the password required on the line session when security login is enabled (using the **login** command). Use the **no** form of this command to remove a configured password.

Syntax Description

<password>	Alphanumeric character string (up to 16 characters) used to specify the password for the line session

Default Values

By default, there is no login password set for access to the unit.

Command Modes

(config-line)#	Line Interface Configuration Valid interfaces include: Console (con 0) and Telnet (telnet X)

Command History

Release 1.1	Command was introduced

Usage Examples

The following example enables the security login feature and specifies a password for the Telnet session 0:

(config)# **line telnet 0**
(config-telnet0)# **login**
(config-telnet0)# **password mypassword**

ROUTER (RIP) CONFIGURATION COMMAND SET

To activate the Router (RIP) Configuration command set, enter the **router rip** command at the Global Configuration Mode prompt. For example:

```
Router> enable
Router# configure terminal
Router(config)# router rip
Router(config-rip)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

do on page 558

end on page 559

exit on page 560

All other commands for this command set are described in this section in alphabetical order.

auto-summary on page 533

default-metric <value> on page 534

network <address> <subnet mask> on page 535

passive-interface <interface> on page 536

redistribute connected on page 537

redistribute ospf [metric <value>] on page 538

redistribute static on page 539

version <version> on page 540

auto-summary

Use the **auto-summary** command to have RIP version 2 summarize subnets to the classful boundaries. Use the **no** form of this command to disable this summarization.

Syntax Description

No subcommands.

Default Values

By default, auto-summary is disabled.

Command Modes

(config-rip)#	Router (RIP) Configuration Mode required

Command History

Release 3.1	Command was introduced

Functional Notes

Use this command if you are subdividing a classful network into many subnets and these subnets are to be advertised over a slow link (≤ 64K) to a router that can only reach the classful network via the router you are configuring.

Usage Examples

The following example configures the router to not automatically summarize network numbers:

(config)# **router rip**
(config-rip)# **no auto-summary**

default-metric <*value*>

Use the **default-metric** command to set the default metric value for the RIP routing protocol. Use the **no** form of this command to return to the default settings.

Syntax Description

<*value*>	Set the default metric value (range: 1-4294967295 Mbps).

Default Values

By default, this value is set at 0.

Command Modes

(config-ospf)# (config-rip)#	Router (OSPF or RIP) Configuration Mode required

Command History

Release 3.1	Command was introduced

Functional Notes

The metric value defined using the **redistribute** command overrides the **default-metric** command's metric setting. See *redistribute ospf [metric <value>]* on page 538 for related information.

Usage Examples

The following example shows a router using both RIP and OSPF routing protocols. The example advertises OSPF-derived routes using the RIP protocol and assigns the OSPF-derived routes a RIP metric of 10.

```
(config)# router rip
(config-rip)# default-metric 10
(config-rip)# redistribute ospf
```


network *<address> <subnet mask>*

Use the **network** command to enable RIP on the specified network. The ADTRAN OS will only allow processing (sending and receiving) RIP messages on interfaces with IP addresses that are contained in the networks listed using this command. All RIP messages received on interfaces not listed using this command will be discarded. To allow for receiving and participating in RIP but not for transmitting, use the **passive-interface** command (see *passive-interface <interface>* on page 536). Use the **no** form of this command to remove a network from the list.

Syntax Description

<address>	IP address of the network on which RIP will be enabled
<subnet mask>	Subnet mask that corresponds to the entered IP address

Default Values

By default, RIP is not enabled.

Command Modes

(config-rip)#	Router (RIP) Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example enables RIP on the 102.22.72.252/30, 192.45.2.0/24, and 10.200.0.0/16 networks:

(config)# **router rip**
(config-rip)# **network 102.22.72.252 255.255.255.252**
(config-rip)# **network 192.45.2.0 255.255.255.0**
(config-rip)# **network 10.200.0.0 255.255.0.0**

passive-interface *<interface>*

Use the **passive-interface** command to disable the transmission of routing updates on the specified interface. All routing updates received on that interface will still be processed (and advertised to other interfaces), but no updates will be transmitted to the network connected to the specified interface. Multiple **passive-interface** commands may be used to create a customized list of interfaces. Use the **no** form of this command to enable the transmission of routing updates on an interface.

Syntax Description

<interface>	Specifies the interface that will not transmit routing updates. Valid interfaces include: Ethernet (eth 0/1), virtual frame relay sub-interfaces (fr 1.16), virtual PPP interfaces (ppp 1), and loopback interfaces

Default Values

By default, RIP is not enabled.

Command Modes

(config-rip)#	Router (RIP) Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example disables routing updates on the frame relay link (labeled 1.17) and the PPP link (labeled 1):

(config)# **router rip**
(config-rip)# **passive-interface frame-relay 1.17**
(config-rip)# **passive-interface ppp 1**

redistribute connected

Use the **redistribute connected** command to pass routes from one network to another, regardless of the routing protocol implemented on the routing domain. Using the **connected** keyword allows the propagation of routes connected to other interfaces using the RIP routing protocol. Use the **no** form of this command to disable the propagation of the specified route type.

Syntax Description

connected	Optional keyword that specifies the ADTRAN OS to only propagate connected routes to other networks

Default Values

By default, RIP is not enabled.

Command Modes

(config-rip)#	Router (RIP or OSPF) Configuration Mode required

Command History

Release 3.1	Command was introduced

Functional Notes

Redistributing connected routes imports those routes into RIP without the interfaces in question actually participating in RIP. The connected routes imported this way are not covered by a network command and therefore do not send/receive RIP traffic.

Usage Examples

The following example passes the connected routes found in the route table to other networks running the RIP routing protocol:

(config)# **router rip**
(config-rip)# **redistribute connected**

redistribute ospf [metric <*value*>]

Use the **redistribute ospf** command to advertise routes from one protocol to another, regardless of the routing protocol implemented on the routing domain. Using the **ospf** keyword allows the propagation of OSPF routes into RIP. Use the **no** form of this command to disable the propagation of the specified route type.

Syntax Description

ospf	Optional keyword that specifies the ADTRAN OS to import OSPF routes into RIP.
metric <*value*> **Optional*	Specifies the hop count to use for advertising redistributed OSPF routes in RIP.

Default Values

By default, this command is disabled.

Command Modes

(config-rip)#	Router (RIP) Configuration Mode required

Command History

Release 3.1	Command was introduced

Functional Notes

Redistributing OSPF routes imports those routes into RIP without the interfaces in question actually participating in RIP. The OSPF routes imported this way are not covered by a network command and therefore do not send/receive RIP traffic.

If r**edistribute ospf** is enabled and no metric value is specified, the value defaults to **0**. The metric value defined using the **redistribute ospf metric** command overrides the **default-metric** command's metric setting. See the section *default-metric <value>* on page 534 for more information.

Usage Examples

The following example imports OSPF routes into RIP:

```
(config)# router rip
(config-rip)# redistribute ospf
```

redistribute static

Use the **redistribute static** command to pass routes from one network to another, regardless of the routing protocol implemented on the routing domain. Using the **static** keyword allows the propagation of static routes to other interfaces using the RIP routing protocol. Use the **no** form of this command to disable the propagation of the specified route type.

The gateway network for the static route must participate in RIP by using the network command for the gateway network.

Syntax Description

static	Optional keyword that specifies the ADTRAN OS to only propagate static routes to other networks

Default Values

By default, RIP is not enabled.

Command Modes

(config-rip)#	Router (RIP or OSPF) Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

Redistributing static routes allows other network devices to learn about paths (not compatible with their system) without requiring manual input to each device on the network.

Usage Examples

The following example passes the static routes found in the route table to other networks running the RIP routing protocol:

(config)# **router rip**
(config-rip)# **redistribute static**

version <*version*>

Use the **version** command to specify (globally) the Routing Information Protocol (RIP) version used on all IP interfaces. This global configuration is overridden using the configuration commands **ip rip send version** and **ip rip receive version**. Use the **no** form of this command to return to the default value.

Syntax Description

<*version*>	Specifies the RIP version used globally
1	RIP version 1
2	RIP version 2

Default Values

By default, RIP is not enabled.

Command Modes

(config-rip)#	Router (RIP) Configuration Mode required

Command History

Release 1.1	Command was introduced

Usage Examples

The following example specifies RIP version 2 as the global RIP version:

(config)# **router rip**
(config-rip)# **version 2**

ROUTER (OSPF) CONFIGURATION COMMAND SET

To activate the Router (OSPF) Configuration command set, enter the **router ospf** command at the Global Configuration Mode prompt. For example:

```
Router> enable
Router# configure terminal
Router(config)# router ospf
Router(config-ospf)#
```

The following commands are common to multiple command sets and are covered in a centralized section of this guide. For more information, refer to the sections listed below:

do on page 558

end on page 559

exit on page 560

All other commands for this command set are described in this section in alphabetical order.

area <*area id*> **default-cost** <*value*>

Use this command to assign a cost of the default summary route sent into a stub area or not-so-stubby-area (NSSA). Use the **no** form of this command to delete the assigned cost.

Syntax Description

<*area id*>	Identifier for this area. Enter as an integer (range: 0-4294967295) or an IP address <A.B.C.D>.
<*value*>	Default summary route cost. Range: 0-166777214.

Default Values

<*area id*>	No default.
<*value*>	0

Command Modes

(config-ospf)#	Router (OSPF) Configuration Mode required

Command History

Release 3.1	Command was introduced

Usage Examples

The following example defines a default cost of 85 to a specific area:

(config)# **router ospf**
(config-ospf)# **area 192.22.72.0 default-cost 85**

area <*area id*> **range** <*ip address*> <*network mask*> **[advertise | not-advertise]**

Use this command to configure area route summarizations and to determine whether an address range is advertised to the networks.

Syntax Description

<*area id*>	Identifier for this area. Enter as an integer (range: 0-4294967295) or an IP address <A.B.C.D>.
<*ip address*>	The IP address of the advertised summary route.
<*network mask*>	The mask of the advertised summary route.
advertise	The specified address range will be advertised to other networks.
not-advertise	The specified address range will not be advertised to other networks.

Default Values

By default, OSPF is not enabled.

Command Modes

(config-ospf)#	Router (OSPF) Configuration Mode required

Command History

Release 3.1	Command was introduced

Usage Examples

(config)# **router ospf**
(config-ospf)# **area 11.0.0.0 range 11.0.0.0 255.0.0.0 advertise**

area <*area id*> **stub** [**no-summary**]

Use this command to configure an area as a stub area. Use the **no** form of this command to disable stub-designation for areas defined as stubs using this command.

Syntax Description

<*area id*>	Identifier for this stub area. Enter as an integer (range: 0-4294967295) or an IP address <A.B.C.D>.
no-summary ***Optional**	Use this optional keyword to designate the area as a total stub area. No summary link advertisements will be sent by the ABR into the stub area.

Default Values

By default, OSPF is not enabled.

Command Modes

(config-ospf)#	Router (OSPF) Configuration Mode required

Command History

Release 3.1	Command was introduced

Technology Review

It is important to coordinate configuration of all routers and access servers in the stub area. The **area stub** command must be configured for each of those pieces of equipment. Use the area router configuration command with the **area default-cost** command to specify the cost of a default internal router sent into a stub area by an ABR. See *area <area id> default-cost <value>* on page 542 for related information.

Usage Examples

(config)# **router ospf**
(config-ospf)# **area 2 stub**

auto-cost reference-bandwidth <*rate*>

Use the **auto-cost reference-bandwidth** command to assign a different interface cost to an interface. It may be necessary to assign a higher number to high-bandwidth links. This value is used in OSPF metric calculations.

Syntax Description

<*rate*>	Set the default reference-bandwidth rate (range: 1-4294967 Mbps).

Default Values

By default, the rate is set to 100.

Command Modes

(config-ospf)#	Router (OSPF) Configuration Mode required

Command History

Release 3.1	Command was introduced

Usage Examples

The following example sets the auto cost reference-bandwidth to 1000 Mbps:

(config)# **router ospf**
(config-ospf)# **auto-cost reference-bandwidth 1000**

default-information-originate [always] [metric *value*] [metric-type *type*]

Use the **default-information-originate** command to cause an ASBR to generate a default route. It must have its own default route before it generates one unless the **always** keyword is used.

Syntax Description

always ***Optional**	Always advertise default route.
metric <*value*> **Optional*	Configure metric value (range is 0-16777214).
metric type <*type*> **Optional*	Configure metric type (1 or 2).

Default Values

metric <*value*>	10
metric type <*type*>	2

Command Modes

(config-ospf)#	Router (OSPF) Configuration Mode required

Command History

Release 3.1	Command was introduced

Usage Examples

(config)# **router ospf**
(config-ospf)# **default-information-originate always metric 10000 metric-type 2**

default-metric <*value*>

Use the **default-metric** command to set a metric value for redistributed routes.

Syntax Description

<*value*>	Set the default metric value (range: 0-4294967295).

Default Values

By default, this value is set at 20.

Command Modes

(config-ospf)#	Router (OSPF or RIP) Configuration Mode required

Command History

Release 3.1	Command was introduced

Functional Notes

The metric value defined using the **redistribute** command overrides the **default-metric** command's metric setting. See *redistribute ospf [metric <value>]* on page 538 for related information.

Usage Examples

The following example shows a router using both RIP and OSPF routing protocols. The example advertises RIP-derived routes using the OSPF protocol and assigns the RIP-derived routes an OSPF metric of 10.

(config)# **router ospf**
(config-ospf)# **default-metric 10**
(config-ospf)# **redistribute rip**

network <*ip address*> <*wildcard*> **area** <*area id*>

Use the **network area** command to enable routing on an IP stack and to define area IDs for the interfaces on which OSPF will run. Use the **no** form of this command to disable OSPF routing for interfaces defined using this command.

Syntax Description

<*ip address*>	Network address <A.B.C.D>.
<*wildcard*>	The wildcard mask is in an IP-address-type format and includes “don’t care” bits.
<*area id*>	Identifier for this area. Enter as an integer (range: 0-4294967295) or an IP address <A.B.C.D>.

Default Values

No default values required for this command.

Command Modes

(config-ospf)#	Router (OSPF) Configuration Mode required

Command History

Release 3.1	Command was introduced

Technology Review

In order for OSPF to operate on an interface, the *primary* address for the interface must be included in the **network area** command. Assigning an interface to an OSPF area is done using the **network area** command. There is no limit to the number of network area commands used on a router. If the address ranges defined for different areas overlap, the first area in the **network area** command list is used and all other overlapping portions are disregarded. Try to avoid overlapping to avoid complications.

Usage Examples

In the following example, the OSPF routing process is enabled and two OSPF areas are defined.

(config)# **router ospf**
(config-ospf)# **network 192.22.72.101 0.0.0.255 area 0**
(config-ospf)# **network 10.0.0.0 0.255.255.255 area 10.0.0.0**

redistribute connected

Use the **redistribute connected** command to advertise routes from one protocol to another. Using the **connected** keyword allows the advertisement of connected routes into the OSPF routing protocol. This will advertise all connected routes on OSPF-enabled interfaces. It does not enable OSPF on all interfaces. Use the **no** form of this command to disable the propagation of the specified route type.

Syntax Description

connected	Optional keyword that specifies the ADTRAN OS to advertise connected routes to OSPF areas.

Default Values

By default, this command is disabled.

Command Modes

(config-ospf)#	Router (OSPF or RIP) Configuration Mode required

Command History

Release 3.1	Command was introduced

Functional Notes

Redistributing connected routes imports those routes into OSPF without the interfaces in question actually participating in OSPF. The connected routes imported this way are not covered by a network command and therefore do not send/receive OSPF traffic.

Usage Examples

The following example imports connected routes into OSPF:

(config)# **router ospf**
(config-ospf)# **redistribute connected**

redistribute rip

Use the **redistribute rip** command to advertise routes from one protocol to another, regardless of the routing protocol implemented on the routing domain. Using the **rip** keyword allows the propagation of RIP routes into OSPF. Use the **no** form of this command to disable the propagation of the specified route type.

Syntax Description

rip	Optional keyword that specifies the ADTRAN OS to import RIP routes into OSPF.

Default Values

By default, this command is disabled.

Command Modes

(config-ospf)#	Router (OSPF) Configuration Mode required

Command History

Release 3.1	Command was introduced

Functional Notes

Redistributing RIP routes imports those routes into OSPF without the interfaces in question actually participating in OSPF. The RIP routes imported this way are not covered by a network command and therefore do not send/receive OSPF traffic.

Usage Examples

The following example imports RIP routes into OSPF:

```
(config)# router ospf
(config-ospf)# redistribute rip
```


redistribute static

Use the **redistribute static** command to advertise routes from one protocol to another. Using the **static** keyword allows the advertisement of static routes into the OSPF routing protocol. This will advertise all static routes on OSPF-enabled interfaces. It does not enable OSPF on all interfaces. Use the **no** form of this command to disable the propagation of the specified route type.

Syntax Description

static	Optional keyword that specifies the ADTRAN OS to import static routes into OSPF.

Default Values

By default, this command is disabled.

Command Modes

(config-ospf)#	Router (OSPF or RIP) Configuration Mode required

Command History

Release 1.1	Command was introduced

Functional Notes

Redistributing static routes imports those routes into OSPF without the interfaces in question actually participating in OSPF. The static routes imported this way are not covered by a network command and therefore do not send/receive OSPF traffic.

Usage Examples

The following example imports static routes into OSPF:

(config)# **router ospf**
(config-ospf)# **redistribute static**

summary-address <*address*> <*mask* | *prefix mask*> **not-advertise**

Use this command to control address summarization of routes that are redistributed into OSPF from other sources (e.g., RIP-to-OSPF, static-to-OSPF, etc.). The **not-advertise** option causes suppression of routes that match the specified mask/prefix mask pair.

Syntax Description

<*address*>	IP address or Prefix A.B.C.D.
<*mask*	*prefix mask*>	Routes matching this mask/prefix mask pair will be suppressed if the **not-advertise** command is enabled.
not advertise **Optional*	Causes suppression of routes that match the specified mask/prefix mask pair.

Default Values

By default, this command is disabled.

Command Modes

(config-ospf)#	Router (OSPF) Configuration Mode required

Command History

Release 3.1	Command was introduced

Usage Examples

The following example suppresses advertisement of the routes which match the specified address/mask:

(config)# **router ospf**
(config-ospf)# **summary-address 11.0.0.0 255.0.0.0 not-advertise**

timers lsa-group-pacing *<seconds>*

Use the **timers lsa-group-pacing** command to change the link state advertisement (LSA) refresh interval.

Syntax Description

<seconds>	Set the LSA refresh interval in seconds (range: 10-1800).

Default Values

By default, this value is set at 240 seconds.

Command Modes

(config-ospf)#	Router (OSPF) Configuration Mode required

Command History

Release 3.1	Command was introduced

Usage Examples

The following example sets the refresh interval for six minutes:

(config)# **router ospf**
(config-ospf)# **timers lsa-group-pacing 360**

timers spf <*delay*> <*hold*>

Use the **timers spf** command to configure the shortest path first (SPF) calculation and hold intervals.

Syntax Description

<*delay*>	Time in seconds between OSPF's receipt of topology changes and the beginning of SPF calculations.
<*hold*>	Time in seconds between consecutive SPF calculations. Range: 10-1800 seconds.

Default Values

<*delay*>	5 seconds
<*hold*>	10 seconds

Command Modes

(config-ospf)#	Router (OSPF) Configuration Mode required

Command History

Release 3.1	Command was introduced

Usage Examples

The following example defines a delay of 10 seconds and a hold-time of 30 seconds:

(config)# **router ospf**
(config-ospf)# **timers spf 10 30**

COMMON COMMANDS

The following section contains descriptions of commands which are common across multiple command sets. These commands are listed in alphabetical order.

alias <*"text"*>

Use the **alias** command to populate the ifAlias OID (Interface Table MIB of RFC 2863) for all physical interfaces and frame relay virtual interfaces when using SNMP management stations.

Syntax Description

<input>	Alphanumeric character string describing the interface (for SNMP) — must be encased in quotation marks

Default Values

No defaults required for this command.

Command Modes

(config-interface)#	Interface Configuration Mode

Command History

Release 1.1	Command was introduced.

Functional Notes

The ifAlias OID is a member of the ifXEntry object-type (defined in RFC 2863) used to provide a non-volatile, unique name for various interfaces. This name is preserved through power cycles. Enter a string (using the **alias** command) which clearly identifies the interface.

Usage Examples

The following example defines a unique character string for the T1 interface:

(config)# **interface t1 1/1**
(config-t1 1/1)# **alias "CIRCUIT_ID_23-908-8887-401"**

Technology Review

Please refer to RFC 2863 for more detailed information on the ifAlias display string.

description

Use the **description** command as a comment line to enter an identifier for the specified interface (for example, circuit ID, contact information, etc.).

Syntax Description

No subcommands.

Default Values

No defaults required for this command.

Command Modes

Any Configuration Mode.

Command History

Release 1.1	Command was introduced

Usage Examples

The following example enters comment information using the **description** command:

```
(config)# interface t1 1/1
(config-t1 1/1)# description This is the Dallas office T1
```

do

Use the **do** command to execute any ADTRAN OS command, regardless of the active configuration mode.

Syntax Description

No subcommands

Default Values

No defaults required for this command.

Command Modes

Any Configuration Mode

Command History

Release 2.1	Command was introduced

Functional Notes

Use the **do** command to view configurations or interface states after configuration changes are made without exiting to the Enable mode.

Usage Examples

The **do** command provides a way to execute commands in other command sets without taking the time to exit the current command set and enter the desired one. The following example shows the **do** command used to view the frame relay interface configuration while currently in the T1 interface command set:

```
(config)# interface t1 1/1
(config-t1 1/1)# do show interfaces fr 7

fr 7 is ACTIVE
  Signaling type is ANSI signaling role is USER
  Polling interval is 10 seconds full inquiry interval is 6 polling intervals
Output queue: 0/0 (highest/drops)
    0 packets input 0 bytes
    0 pkts discarded 0 error pkts 0 unknown protocol pkts
    0 packets output 0 bytes
    0 tx pkts discarded 0 tx error pkts
```


end

Use the **end** command to exit the current Configuration Mode and enter the Enable Security Mode.

*When exiting the Global Configuration Mode, remember to perform a **copy running-config startup-config** to save all configuration changes.*

Syntax Description

No subcommands

Default Values

No defaults necessary for this command.

Command Modes

This command is valid for all command modes except the Enable Security Mode.

Command History

Release 1.1	Command was introduced

Usage Examples

The following example shows the end command being executed in the T1 Configuration Mode:

```
(config-t1 1/1)# end
#
```

- Enable Security Mode command prompt

exit

Use the **exit** command to exit the current Configuration Mode and enter the previous one. For example, using the **exit** command in the Interface Configuration Mode will activate the Global Configuration Mode. When using the **exit** command in the Basic Mode, the current session will be terminated.

*When exiting the Global Configuration Mode, remember to perform a **copy running-config startup-config** to save all configuration changes.*

Syntax Description

No subcommands

Default Values

No defaults necessary for this command.

Command Modes

This command is valid for all Configuration Modes.

Command History

Release 1.1	Command was introduced

Usage Examples

The following example shows the exit command being executed in the Global Configuration Mode:

```
(config)# exit
#
```

- Enable Security Mode command prompt

shutdown

Use the **shutdown** command to administratively disable the interface (no data will be passed through). Use the **no** form of this command to activate the interface.

Syntax Description

No subcommands

Default Values

By default, all interfaces are disabled.

Command Modes

Any Configuration Mode

Command History

Release 1.1	Command was introduced

Usage Examples

The following example administratively disables the modem interface:

(config)# **interface modem 1/2**
(config-modem 1/2)# **shutdown**

Index

D

E

R

S

