CoSine IPNOS Command Line Reference

Release 2.1

Part Number: 050-00116-01

CoSine Communications, Inc.
1200 Bridge Parkway
Redwood City, California 94065
650-637-4777
info@cosinecom.com
www.cosinecom.com

CoSine Communications and the CoSine Communications logo are registered trademarks of CoSine Communications, Inc. All CoSine product names are trademarks of CoSine Communications, Inc. Other brand and product names are trademarks of their respective holders.

Copyright © 1998-2001 CoSine Communications, Inc. All rights reserved.

Contents

Preface

Audience. xxiii
Organization xxiv
Conventions xxv
Related Documentation xxvi
Technical Assistance Center xxvii

Chapter 1 Command Usage and Structure

Command Line Interface 1-1
CLI Access 1-2
InVision Service Management System. 1-2
Command Structure 1-2
Config and Show 1-3
Global and Local 1-3
Physical and Virtual 1-3
Command Prompts. 1-3
Command Shortcuts 1-4

Chapter 2 Global Commands

? 2-2
abort 2-3
cls 2-4
commit 2-5
config. 2-6
control-blade-failover 2-7
help 2-8
history 2-9
load 2-10
logout 2-11
reboot 2-12
save 2-14
show 2-15

sysname 2-16
sysuser 2-17
top 2-18
up 2-19
version 2-20

Chapter 3 Config Commands

blade 3-3
blade-stop-fs 3-3
display 3-5
reset-blade 3-7
tftpget 3-8
tftpput 3-10
bootcfg 3-12
bootfile1, bootfile2, bootfile3, bootfile4, bootfile5, bootfile6 3-12
configfile 3-14
display 3-15
gateway 3-15
host 3-16
port 3-17
savefile 3-17
target 3-18
chassis 3-19
display 3-20
fan-tray 3-21
display 3-23
event-trap 3-25
power-supply 3-25
display 3-26
event-trap 3-27
status-panel 3-27
display 3-28
vsio 3-28
display 3-29
ipnos-diag 3-30
stop-fs 3-31
syscontact 3-32
syslocation 3-33
temp-sensor 3-34
display 3-34
controller 3-35
create 3-36
delete 3-37

display 3-38
filesystem 3-50
copy 3-51
delete 3-52
dir 3-54
rename 3-55
show 3-56
interface 3-57
system-up-time 3-59
time 3-60
verbose-load 3-62
vpn 3-63

Chapter 4 Control and Ethernet Blade Commands

ethernet 4-2
autoneg 4-3
display 4-4
duplex 4-5
encapsulation 4-6
speed 4-7
status 4-8
vlanid 4-9
userpriority 4-10
vlanname 4-11

Chapter 5 Channelized DS3 Blade Commands

cablelength 5-2
channel 5-3
clock 5-5
display 5-6
framing 5-9
loopback 5-10
shutdown 5-12
t1 5-13
serial 5-17
display 5-18
encapsulation 5-20

Chapter 6 Unchannelized DS3 Blade Commands

serial 6-2
cablelength 6-2

clock 6-3
display 6-4
encapsulation 6-5
framing 6-6
loopback 6-7
shutdown 6-8

Chapter 7 Channelized E3 Blade Commands

channel 7-2
display 7-4
e1 7-6
linecode 7-10
loopback 7-11
national bit 7-13
shutdown 7-14
serial 7-15
display 7-16
encapsulation 7-18

Chapter 8 Unchannelized E3 Blade Commands

serial 8-2
clock 8-2
display 8-3
encapsulation 8-4
loopback 8-7
shutdown 8-9

Chapter 9 OC-3c Commands

oc3 9-2

Chapter 10 ATM Commands

atm 10-3
atm 10-4
display 10-17
atm 10-23
arp 10-24
arp-age 10-28
arp-max-entries 10-29
clear 10-30
display 10-32

enable 10-33
encapsulation. 10-34
icr. 10-36
ip-addr 10-37
mbs 10-38
mcr 10-40
oam-loopback 10-41
pcr 10-41
pppoe 10-42
display 10-43
pppoe 10-45
session 10-48
protocol 10-50
receive-size. 10-54
scr 10-55
service-category 10-56
transmit-size 10-65
vci 10-66
vpi 10-69

Chapter 11 POS Commands

pos. 11-2
display 11-2
encapsulation. 11-7

Chapter 12 Frame Relay Commands

display 12-2
frame-relay 12-3
pos. 12-7
serial 12-8
display 12-9
frame-relay. 12-10

Chapter 13 HDLC Commands

display 13-1

Chapter 14 PPP Commands

display 14-2
event-log 14-3
ppp 14-7

bundle 14-9
admin-status 14-10
display 14-11
link 14-12
max-links 14-13
queue-size 14-13
reorder-timeout 14-14
site-name 14-14

Chapter 15 VPN Commands

create 15-2
delete 15-5
display 15-6
name 15-7
show 15-8
vr 15-11

Chapter 16 VR Commands

access-list 16-5
acl 16-7
display 16-8
dstaddr 16-9
dstport 16-11
protocol 16-11
srcaddr 16-12
srcport 16-14
tos 16-14
arp 16-16
certificate 16-17
certificate 16-18
display 16-19
flag 16-20
private-key 16-20
clear 16-22
create 16-24
crypto 16-25
decr-ttl 16-27
delete 16-28
dhcp-relay 16-29
display 16-37
dns 16-43
ds-byte 16-44

export-policy 16-46
acl 16-46
action 16-47
bgp-as 16-48
bgp-community 16-49
display 16-51
metric 16-52
peer-as 16-53
protocol 16-54
rip-ip 16-56
froip 16-57
fw 16-58
hostname 16-60
import-policy 16-61
acl 16-62
action 16-62
bgp-community 16-63
display 16-64
home-as 16-65
origin 16-65
peer-as 16-66
preference 16-67
ip 16-68
l2tp 16-78
display 16-80
l2tp 16-82
lac 16-90
lns 16-91
ppp 16-93
tunnel 16-96
display 16-97
session 16-98
tunnel-switch 16-99
display 16-100
mode 16-100
switch-to 16-103
name 16-106
nbns 16-107
packet-filter 16-108
ping 16-112
pptp 16-113
display 16-114
pptp 16-115

qos 16-118
display 16-118
ds-byte-filter 16-119
radius-server 16-122
reassembly 16-128
router 16-129
service-selection 16-130
connection-profile 16-131
display 16-132
ip-local-pool-name 16-132
lns-address 16-133
vpn 16-134
vr 16-134
display 16-135
interface 16-136
service-selection-template 16-136
connection-profile 16-137
display 16-138
show 16-139
snmp-server 16-144
syslogd 16-151
telnet 16-157
display 16-157
enable 16-158
kill 16-159
show 16-160
timeout 16-160
traceroute 16-162
ttl 16-163
type 16-164
username 16-165
vi 16-167
wins 16-168

Chapter 17 Crypto Commands

ipsec 17-3
transform-set 17-3
display 17-4
initvector 17-5
mode 17-6
isakmp 17-8
parameter 17-8
display 17-9

expire-timer 17-10
flag 17-10
local-secret-size 17-14
max-retry-timer 17-14
max-sa-count 17-15
retry-limit 17-16
retry-timer-sec 17-17
retry-timer-usec 17-18
secret-recreate 17-19
token-hash-type 17-20
policy 17-20
authentication 17-21
display 17-23
encryption 17-24
group 17-25
hash 17-26
lifetime 17-27
map 17-29
ipsec-dialup 17-30
display 17-30
match 17-32
set 17-33
ipsec-isakmp 17-39
display 17-39
match 17-40
set 17-40
ipsec-manual 17-46
display 17-47
match 17-47
set 17-48

Chapter 18 Firewall Commands

access 18-5
action 18-5
destination 18-6
display 18-7
policy 18-7
port 18-8
source 18-9
aol-proxy 18-10
cs-proxy 18-11
event-log 18-12
display 18-12
level. 18-13

ftp-proxy 18-15
child-limit 18-16
display 18-16
listen. 18-17
policy 18-18
display 18-19
dynamic-pkt-filter 18-21
function 18-22
handoff 18-25
logged 18-26
timeout 18-29
http-proxy 18-31
child-limit 18-31
display 18-33
listen. 18-34
policy 18-35
display 18-36
feature 18-37
function 18-38
handoff 18-40
header 18-41
logged 18-43
ourname 18-45
timeout 18-46
ldap-proxy 18-47
lnotes-proxy 18-48
netm-proxy 18-49
nntp-proxy 18-50
plug-proxy 18-51
child-limit 18-52
display 18-53
listen. 18-54
policy 18-54
display 18-55
remote-port 18-56
remote-server 18-57
timeout 18-57
pop3-proxy 18-59
child-limit 18-59
display 18-60
listen. 18-61
policy 18-63
display 18-63
pop3-server 18-64

timeout 18-65
rsh-proxy 18-66
child-limit. 18-67
display 18-67
listen 18-68
policy 18-69
display 18-70
remote-server 18-71
timeout 18-71
smtp-proxy 18-73
child-limit. 18-73
display 18-74
listen 18-75
policy 18-76
display 18-78
local-domain 18-78
log-relay 18-79
log-spam 18-80
logging 18-80
mail-hub 18-81
max-bytes 18-81
max-recip 18-82
mime-check 18-82
relay-host 18-83
relay-reply 18-84
spam-host 18-84
spam-reply 18-85
timeout 18-85
sql-proxy 18-87
child-limit. 18-87
display 18-88
listen 18-89
policy 18-90
display 18-91
remote-port 18-92
remote-server 18-92
timeout 18-93
telnet-proxy 18-95
child-limit. 18-96
display 18-96
listen 18-97
policy 18-98
display 18-99
timeout 18-100

Chapter 19 Frame Relay over IP Commands

default-service-slot 19-2
default-window-size 19-3
default-window-timeout 19-4
display 19-5
local-vi 19-6
max-transport 19-7
transport 19-8
admin-status 19-9
default-slot 19-9
display 19-10
dlci 19-11
bc 19-12
be 19-13
cir 19-14
display 19-15
route 19-16
event-window-size 19-17
keep-alive-period 19-17
permit-bad-event 19-18
window-size 19-19
window-timeout 19-19

Chapter 20 Routing Commands

bgp 20-3
bgp 20-3
default-metric 20-8
display 20-9
enable 20-10
event-log 20-11
neighbor 20-12
network 20-19
show 20-19
isis 20-21
area-address 20-21
area-password 20-22
default-info-originate 20-24
display 20-26
domain-password 20-26
enable 20-28
event-log 20-30
export-policy 20-31

level-2-to-1-leaking 20-32
metric-style 20-34
mpls 20-36
show 20-39
spf-interval 20-39
summary-address 20-41
system-id 20-43
system-type 20-44
ospf 20-46
area 20-46
display 20-50
enable 20-52
event-log 20-52
export-policy 20-54
show 20-54
rip 20-56
display 20-56
enable 20-57
event-log 20-58
export-policy 20-62
import-policy 20-63
show 20-64

Chapter 21 VI Commands

clear 21-3
connected-to 21-7
crypto 21-10
display 21-11
event-log 21-16
gre 21-19
checksum 21-20
display 21-21
key-control 21-22
key-value 21-23
sequence 21-24
ip 21-26
isis 21-37
packet-filter 21-53
qos 21-57
display 21-57
ds-byte-filter 21-58
rate-limit 21-61

remote 21-63
router 21-66
show 21-68
tunnel-keepalive 21-69
count 21-69
display 21-70
fails 21-71
frequency 21-72
target 21-73
timeout 21-73

Appendix A SNMP Traps

Appendix B Glossary

Index

Figures

Figure 16-1 Static and Dynamic NAT 16-68
Figure 16-2 Dual Static NAT 16-69
Figure 16-3 L2TP LAC 16-78
Figure 16-4 L2TP LNS. 16-79
Figure 16-5 L2TP Tunnel Switching 16-79
Figure 16-6 PPTP Process. 16-113
Figure 16-7 SNMP Process 16-144
Figure 21-1 GRE Packet Header 21-19

Tables

Table 1-1 IPNOS Command Prompts 1-3
Table 3-1 Parameters for ipnos-diag 3-30
Table 3-2 IPNOS File Descriptions 3-54
Table 3-3 Interface Modes Compared 3-57
Table 3-4 Time Fields 3-60
Table 5-1 Channel Configuration Parameters 5-3
Table 5-2 Performance Monitoring Parameters 5-6
Table 5-3 DS3 Framing Options 5-9
Table 5-4 Loopback Parameters 5-10
Table 5-5 T1 Parameters 5-13
Table 6-1 Performance Monitoring Parameters 6-4
Table 6-2 DS3 Framing Options 6-6
Table 6-3 Loopback Parameters 6-7
Table 7-1 Channel Configuration Settings 7-2
Table 7-2 Performance Monitoring Parameters 7-4
Table 7-3 E1 Parameters 7-6
Table 7-4 Linecode Parameter 7-10
Table 7-5 Loopback Parameters 7-11
Table 8-1 Performance Monitoring Parameters 8-3
Table 8-2 Loopback Parameters 8-7
Table 9-1 OC-3c Parameters 9-2
Table 10-1 ATM Parameters 10-4
Table 10-2 Performance Monitoring Parameters 10-17
Table 10-3 ATM subinterface Encapsulation Parameters 10-34
Table 10-4 ATM Encapsulation, ATM Protocol Compatibility 10-35
Table 10-5 PPPoE Parameters 10-45
Table 10-6 Service Categories 10-56
Table 10-7 Cell Rates Needed for Service Categories 10-57
Table 10-8 VCI Values 10-66
Table 10-9 VPI Values 10-69

Table 11-1 POS Monitoring Statistics 11-2
Table 12-1 LMI Parameters 12-3
Table 12-2 Frame Relay PVC Subinterface Parameters 12-10
Table 14-1 PPP Event Message Area Types 14-3
Table 14-2 PPP Event Message Severity Levels 14-3
Table 14-3 PPP Parameters 14-7
Table 15-1 Blade Combinations Allowed When Creating VRs 15-2
Table 16-1 NAT Access List Command Parameters 16-5
Table 16-2 Special Functions of the dstaddr Command 16-9
Table 16-3 ARP Command Parameters 16-16
Table 16-4 DHCP Relay Parameters 16-29
Table 16-5 Possible Values for IP Precedence Compatibility 16-44
Table 16-6 BGP-Community Parameters 16-49
Table 16-7 Export Policy Protocol Parameters 16-54
Table 16-8 BGP-Community Parameters (Import Policy) 16-63
Table 16-9 BGP Origin Attributes 16-65
Table 16-10 IP Address Routing Commands 16-69
Table 16-11 L2TP Parameters 16-82
Table 16-12 LAC Admin Status Parameters 16-90
Table 16-13 LNS Admin Status Parameters 16-92
Table 16-14 PPP Parameters for L2TP 16-93
Table 16-15 L2TP LAC Mode Parameters 16-100
Table 16-16 L2TP switch-to parameters 16-103
Table 16-17 VR Packet Filter Parameters 16-108
Table 16-18 Ping Parameters 16-112
Table 16-19 PPTP Server Parameters 16-115
Table 16-20 ds-byte-filter Parameters 16-119
Table 16-21 Possible Values for IP Precedence Compatibility 16-120
Table 16-22 Radius-Server Parameters 16-122
Table 16-23 Layer 2 Encapsulation Supported by Service Selection 16-130
Table 16-24 SNMP Server Parameters 16-145
Table 16-25 Syslogd Parameters 16-152
Table 16-26 Traceroute Parameters 16-162
Table 16-27 VR Types 16-164
Table 17-1 IPSec Transform Set Encryption and Authentication Types 17-3
Table 17-2 ISAKMP Parameter Flags 17-10
Table 17-3 ISAKMP Policy Options for Certificate and Pre-Shared Key 17-22
Table 17-4 ISAKMP Lifetime Settings 17-27
Table 17-5 IPSEC Dialup Set Options 17-33

Table 17-6 IPSEC ISAKMP Set Parameters 17-41
Table 17-7 IPSec-Manual Settings for Session Keys and Transform Sets 17-48
Table 17-8 IPSec Session Key Sizes (in bytes) 17-49
Table 18-1 IPSec Event Message Severity Levels 18-13
Table 18-2 FTP Functions. 18-22
Table 18-3 FTP Transactions 18-26
Table 18-4 HTTP-Proxy Features 18-37
Table 18-5 HTTP Functions 18-38
Table 18-6 HTTP Packet Headers 18-41
Table 18-7 HTTP Transactions. 18-43
Table 19-1 DLCI Route Parameters 19-16
Table 20-1 BGP Parameters 20-3
Table 20-2 BGP Neighbor Parameters. 20-12
Table 20-3 MPLS Metric-Style Parameters 20-34
Table 20-4 OSPF Area Parameters. 20-46
Table 20-5 RIP Event-Log Parameters 20-58
Table 21-1 VI Clear Command Parameters. 21-3
Table 21-2 VI Connection Type 21-7
Table 21-3 IPSec Event Message Areas 21-16
Table 21-4 IPSec Event Message Severity Levels 21-16
Table 21-5 Virtual Interface IP Parameters 21-26
Table 21-6 IS-IS Connection Parameters 21-37
Table 21-7 VI Packet Filter Parameters 21-53
Table 21-8 ds-byte-filter Parameters 21-58
Table 21-9 Possible Values for IP Precedence Compatibility 21-59
Table 21-10 VI Rate-limit Parameters. 21-61
Table 21-11 VI Connection Type 21-63

Preface

The *Cosine IPNOS Command Line Reference* provides descriptions, syntax, and examples of each command in the Cosine Internet Protocol Network Operating System (IPNOS™) software.

Section	Page
Audience	xxiii
Organization	xxiv
Conventions	xxv
Related Documentation	xxvi
Technical Assistance Center	xxvii

Audience

This publication is intended for network operations personnel who maintain high-speed, carrier-based networks. It is a standalone reference for experienced network administrators who are configuring and maintaining the IPSX Service Processing Switch (SPS).

Organization

This publication contains the following sections:

Chapter	Description
Chapter 1, “Command Usage and Structure”	Overview of command usage, command modes, command levels, and the CLI hierarchy.
Chapter 2, “Global Commands”	Descriptions, syntax, and examples of global commands.
Chapter 3, “Config Commands”	Descriptions, syntax, and examples of config commands for hardware and interfaces.
Chapter 4, “Control and Ethernet Blade Commands”	Descriptions, syntax, and examples of control blade commands.
Chapter 5, “Channelized DS3 Blade Commands”	Descriptions, syntax, and examples of channelized digital signal level 3 (DS3) blade commands.
Chapter 6, “Unchannelized DS3 Blade Commands”	Descriptions, syntax, and examples of unchannelized digital signal level 3 (DS3) blade commands.
Chapter 7, “Channelized E3 Blade Commands”	Descriptions, syntax, and examples of channelized E3 blade commands.
Chapter 8, “Unchannelized E3 Blade Commands”	Descriptions, syntax, and examples of unchannelized E3 blade commands.
Chapter 9, “OC-3c Commands”	Descriptions, syntax, and examples of optical carrier-3 (OC-3) blade commands.
Chapter 10, “ATM Commands”	Descriptions, syntax, and examples of Asynchronous Transfer Mode (ATM) commands.
Chapter 11, “POS Commands”	Descriptions, syntax, and examples of Packet over SONET (POS) commands.
Chapter 12, “Frame Relay Commands”	Descriptions, syntax, and examples of Frame Relay interface and PVC commands.
Chapter 13, “HDLC Commands”	Descriptions, syntax, and examples of high-level data link control (HDLC) commands.
Chapter 14, “PPP Commands”	Descriptions, syntax, and examples of Point-to-Point Protocol (PPP) commands.
Chapter 15, “VPN Commands”	Descriptions, syntax, and examples of virtual private network (VPN) commands.
Chapter 16, “VR Commands”	Descriptions, syntax, and examples of virtual router (VR) commands.

Chapter	Description
Chapter 17, “Crypto Commands”	Descriptions, syntax, and examples of Internet Protocol Security (IPSec) commands.
Chapter 18, “Firewall Commands”	Descriptions, syntax, and examples of firewall commands.
Chapter 19, “Frame Relay over IP Commands”	Descriptions, syntax, and examples of Frame Relay over IP commands.
Chapter 20, “Routing Commands”	Descriptions, syntax, and examples of routing commands.
Chapter 21, “VI Commands”	Descriptions, syntax, and examples of the virtual interface (VI) commands.
Appendix A, “SNMP Traps”	List of the Simple Network Management Protocol version 2 (SNMPv2) traps.
Appendix B, “Glossary”	Presents a glossary of CoSine-specific terminology.

Conventions

The following conventions are used in this publication:

Italic text is for emphasis and book titles.

Bold text identifies command names, menu names and options, items you click or enter, and keyboard keys.

The following conventions are used for command line syntax:

- Commands you enter are case insensitive, although some CLI information is displayed in upper and lowercase.
- `Courier font` identifies command syntax and examples.
- **`Courier bold font`** identifies commands or arguments you enter.
- Angle brackets < > indicate text or values you need to replace.
- Braces { } indicate required parameters or values, two or more separated by a pipe (|). You select one.
- Square brackets [ ] indicate optional parameters or values. You can select one, several, or all.

NOTE: Notes provide significant information about features, operations, or procedures.

CAUTION: Cautions identify important information about actions that could result in damage to or loss of data or equipment.

Related Documentation

For more information about CoSine Communications products, see the following documents:

- *CoSine IPSX 3500 Hardware Installation and Reference* — Describes how to set up the IPSX 3500, from unpacking through basic configuration. Includes information about the chassis, blades, and power supplies.
- *CoSine IPSX 9000 Hardware Installation and Reference* — Describes how to set up the IPSX 9000, from unpacking through basic configuration. Includes information about the chassis, blades, and power supplies.
- *CoSine IPNOS Command Line Reference* — Describes the syntax and parameters of each command in the Internet Protocol Network Operating System (IPNOS) command line interface (CLI), with examples. Also describes the command level hierarchy.
- *CoSine IPNOS CLI Configuration Guide* — Describes how to use the CLI to configure the IPSX blades; create and provision virtual private networks (VPNs), virtual routers (VRs), and virtual interfaces (VIs); and establish IPSec security and firewalls.
- *CoSine InVision SMS User's Guide* — Describes how to install and use the InVision SMS graphical user interface (GUI) to configure, provision, maintain, and troubleshoot the IPSX switch.
- *CoSine InGage CNM Administrator's Guide* — Describes InGage Customer Network Management (CNM) system and how enterprise network managers will use it to monitor their portion of Service Provider networks.

For the latest versions of product documentation, visit www.cosinecom.com.

Technical Assistance Center

The focal point for all CoSine software and hardware maintenance and support services is the CoSine Technical Assistance Center (TAC). The TAC is staffed by experienced networking and security professionals who provide technical support to customers for all CoSine products.

North America

1 877-8CoSine (1 877 826-7463) email: tac@cosinecom.com

Europe

+ 800 00 CoSine (00 26-7463) email: etac@cosinecom.com

Malaysia

+ 603-2055 8688 email: tac@cosinecom.com

Outside 800 Area Access

1 650-637-6800 email: tac@cosinecom.com

CHAPTER

1 Command Usage and Structure

This chapter describes the types of commands and how you use them.

Section	Page
Command Line Interface	1-1
CLI Access	1-2
InVision Service Management System	1-2
Command Structure	1-2
Config and Show	1-3
Global and Local	1-3
Physical and Virtual	1-3
Command Prompts	1-3
Command Shortcuts	1-4

Command Line Interface

Use the IPNOS™ command line interface (CLI) to configure and maintain the CoSine IPSX Service Processing Switch (SPS). For example, you can configure the following hardware and software features:

- IPSX chassis
- Blades such as control, DS3, E3, OC-3, and fast Ethernet
- Link layer protocols such as Frame Relay, Frame Relay over IP (FRoIP), High-level data link control (HDLC), Point-to-Point Protocol (PPP), Multi-link PPP (MLPPP), packet over SONET (POS), and Asynchronous Transfer Mode (ATM)

- Virtual private network (VPN), Virtual router (VR), and Virtual interface (VI)
- Layer 2 protocols such as Point-to-Point Tunneling Protocol (PPTP), Layer 2 Tunneling Protocol (L2TP)
- Layer 3 protocols such as firewall proxies, network address translation (NAT), packet filters, QoS, and IPSec
- Remote authentication dial-in user service (RADIUS) server
- Domain name server (DNS) and Windows Internet Navigation Server (WINS)
- Access lists and access control list (ACL)
- Routing policies such as Border Gateway Protocol (BGP), Intermediate System-to-Intermediate System (IS-IS), Multiprotocol Label Switching (MPLS), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP)

CLI Access

You can access the CLI by means of a console terminal physically connected to the IPSX switch, or through a terminal emulation program on a PC or UNIX workstation.

InVision Service Management System

The CoSine InVision™ Service Management System (SMS) is a graphical user interface–based alternative to the CLI. It is available for UNIX workstations or PCs running Microsoft Windows NT. Using the Invision SMS is the most practical way to set up and maintain hundreds of VPNs, VRs, and VIs.

Command Structure

The CLI is divided into different hierarchical levels. Each level represents a particular feature. Sometimes a feature has commands at more than one level. For example, the IPSec commands at the crypto level affect the entire virtual router, while the IPSec commands at the VI level affect only a single interface.

Config and Show

The config mode allows you to create, edit, delete, and view all IPNOS services. The show mode allows you to view your configuration status without accidentally changing the current setup. To enter a particular mode, type “config” or “show” at the topmost CLI level.

Global and Local

Global commands are available at any level. Local commands are available only at specific levels, also called a context. The context is the level in which you are currently working.

Physical and Virtual

Commands are divided by their usage into two groups: physical and virtual. Physical commands configure physical hardware such as the IPSX switch, blades, status panel, and the logical interfaces. Virtual commands configure VPN-related items such as VRs, VIs, routing, security, and firewall.

Command Prompts

The command prompt consists of the name of the IPSX switch being configured, followed by a colon and the current configuration level. All command prompt strings are terminated by a pound (#) sign. The prompt changes as you go further up the command hierarchy. Table 1-1 describes some example command prompts.

Table 1-1 IPNOS Command Prompts

Level	*Example Prompt*
Global	`IPNOS:#`
Show	`IPNOS: SHOW#`
Config	`IPNOS: CONFIG#`
Channelized Blade (Controller)	`IPNOS: CONFIG CONTROLLER T3 (2/1)#`
Control Blade (Ethernet Interface)	`IPNOS: CONFIG INTERFACE ETHERNET(1/1)#`
Unchannelized Blade (Serial Interface)	`IPNOS: CONFIG INTERFACE SERIAL(5/1)#`

Table 1-1 IPNOS Command Prompts (Continued)

Level	*Example Prompt*
Channelized Blade (Serial Interface)	`IPNOS: CONFIG INTERFACE SERIAL(2/1:1)#`
Frame Relay (Serial Interface)	`IPNOS: CONFIG INTERFACE SERIAL(4/1) FRAME-RELAY#`
Frame Relay PVC (Serial Interface)	`IPNOS: CONFIG INTERFACE SERIAL(4/1:0:1)#`
PPP (Serial Interface)	`IPNOS: CONFIG INTERFACE SERIAL(4/1) PPP#`
VPN	`IPNOS: CONFIG VPN(50)#`
VR	`IPNOS: CONFIG VPN(50) VR(192.128.42.24)#`
VI	`IPNOS: CONFIG VPN(50) VR(192.128.42.24) VI(1)#`
ACL	`IPNOS: CONFIG VPN(50) VR(192.128.42.24) ACL(1)#`
IPSec Transform Set	`IPNOS: CONFIG VPN(50) VR(192.128.42.24) TRANSFORM-SET(test)#`
ISAKMP Policy	`IPNOS: CONFIG VPN(50) VR(192.128.42.24) POLICY(test1 1)#`
Crypto Map (IPSec ISAKMP)	`IPNOS: CONFIG VPN(50) VR(192.128.42.24) MAP IPSEC-ISAKMP(map2 2)#`
Firewall (HTTP Proxy)	`IPNOS: CONFIG VPN(50) VR(192.128.42.24) HTTP-PROXY#`
Router RIP	`IPNOS: CONFIG VPN(50) VR(192.128.42.24) RIP#`
Router OSPF	`IPNOS: CONFIG VPN(50) VR(192.128.42.24) OSPF#`

Command Shortcuts

You can type an abbreviated form of any IPNOS command for the sake of speed. For example, instead of typing **interface serial 2/1**, you could type **int ser 2/1** and the command will still be recognized by the interpreter. You can abbreviate a command up to a unique character. For example, at the global command level, **cls** can be typed as “cl” and **config** can be typed as “co.” At the config command mode, **cls** can be typed as “cl,” **config** can be typed as “conf,” and **controller** can be typed as “cont.”

CHAPTER

2 Global Commands

Use the IPNOS™ global commands in any level. These commands affect the entire IPSX switch. This chapter describes the following commands:

Command	*Page*
?	2-2
abort	2-3
cls	2-4
commit	2-5
config	2-6
control-blade-failover	2-7
help	2-8
history	2-9
load	2-10
logout	2-11
reboot	2-12
save	2-14
show	2-15
sysname	2-16
sysuser	2-17
top	2-18
up	2-19
version	2-20

?

List the commands in the current level. Use to display command syntax.

Syntax	*Example*
?	IPNOS :# **?** Global: ? Command help ABORT Abort changes CLS Clear screen COMMIT Commit changes CONFIG Configure mode HELP Show help message LOAD Load configuration from file LOGOUT Logout user session REBOOT Reboot the system SAVE Save configuration to file SHOW Show mode SYSNAME Set the system name SYSUSER IPSX system users TOP Go to the top of the tree UP Up one branch VERSION Image version IPNOS :#
[COMMAND]?	IPNOS :# **save?** Global: SAVE IPNOS :#
[COMMAND] ?	IPNOS :# **save ?** SAVE [filename] IPNOS :#

Default Value

Display all commands at the current level

Command Level

global

Related Command

None

abort

Cancel changes at the current command level.

Syntax	*Example*

ABORT	`IPNOS : CONFIG VPN(2)# create vr 3.3.3.3`
	`VPN(2): Object successfully set`
	
	`IPNOS : CONFIG VPN(2) VR(3.3.3.3)# abort`
	`Abort configuration`
	
	`IPNOS : CONFIG VPN(2)#`

Default Value

None

Command Level

global

Related Command

None

cls

Clear the screen.

Syntax	*Example*

CLS	`IPNOS : CONFIG INTERFACE ATM(2/1)# dis atm`
	
	`ATM PARAMETERS`
	`IF-NAME : `
	`MAXVPI-BITS : 10 (0 VPI bits currently active)`
	`MAXVP-NUMBER : 1020 (0 VPCs currently configured)`
	`MAXVCI-BITS : 9 (0 VCI bits currently active)`
	`MAXVC-NUMBER : 500 (0 VCCs currently configured)`
	`ILMI-VPI : 0`
	`ILMI-VCI : 16`
	`ILMI VERSION : VERSION 4.0`
	`UNI TYPE : PRIVATE`
	`UNI VERSION : VERSION 4.0`
	`NEIGHBOR-IP-ADDRESS : 0.0.0.0`
	`NEIGHBOR-NAME : `
	`ADMIN STATUS : DISABLE`
	`OPER STATUS : DOWN`
	
	`IPNOS : CONFIG INTERFACE ATM(2/1)# cls`
	
	`IPNOS : CONFIG INTERFACE ATM(2/1)#`

Default Value

None

Command Level

global

Related Command

None

commit

Apply a configuration.

Syntax	*Example*
COMMIT	IPNOS : CONFIG# **create vpn 5** IPNOS : CONFIG VPN(5)# **commit** Commit configuration VPN(5): Object successfully set IPNOS : CONFIG#

Default Value

None

Command Level

global

Related Commands

load
save

config

Enter the configuration mode. See “Config Commands” in Chapter 3.

Syntax	*Example*
CONFIG	IPNOS : **config** IPNOS : CONFIG#
CONFIG	IPNOS : CONFIG VPN(2) VR(2.2.2.2)# **config** VR(2.2.2.2): Object successfully set VPN(2): Object successfully set IPNOS : CONFIG#

Default Value

None

Command Level

global

Related Command

None

control-blade-failover

Convert standby control blade to master and master control blade to standby. Commit and save your changes before using this command.

Syntax	*Example*

One control blade: `control-blade-failover`	`IPNOS: CONFIG#` **`control-blade-failover`**
	`About to do control blade failover. Are you sure? {YES | NO}:` **`yes`**
	`One Control Blade in the system DO NOT SHUTDOWN !!!`
	`CONTROL-BLADE-FAILOVER: Control blade failover failed`
	
	`IPNOS: CONFIG#`

Two control blades: `control-blade-failover`	`IPNOS: CONFIG#` **`control-blade-failover`**
	`About to do control blade failover. Are you sure? {YES | NO}:` **`yes`**
	`CONTROL-BLADE-FAILOVER: Control blade failover successfully`
	
	`IPNOS: CONFIG#`

Abort failover: `control-blade-failover`	`IPNOS: CONFIG#` **`control-blade-failover`** `About to do control blade failover. Are you sure? {YES | NO}:` **`no`** `IPNOS: CONFIG#`

Default Value

None

Command Level

global

Related Command

None

help

General instructions for using the **?** command.

Syntax	*Example*
HELP	IPNOS : CONFIG VPN(2)# **help** Help may be requested at any point by entering '?'. Two styles of help are provided: 1. Partial help (e.g. 'c?') will display all commands that start with 'c'. 2. Full help (e.g. 'iface ?') will display the valid argument for 'iface'. Entering '?' by itself will show all available commands in the current context.

Default Value

Shows the above message explaining the help system.

Command Level

global

Related Command

?

history

List the most recent commands to the screen.

Syntax	*Example*

`history`	`IPNOS: CONFIG VPN(50)# history`
	
	`1:dis`
	`2:vpn 50`
	`3:creare vr 1.1.1.1`
	`4:cre vr 1.1.1.1`
	`5:comm`
	`6:create vr 2.2.2.2 2`
	`7:comm`
	`8:history`
	
	`IPNOS: CONFIG VPN(50)#`

Default Value

None

Command Level

global

Related Command

None

load

Load from configuration file.

Syntax	*Example*
LOAD [<filename>]	IPNOS :# **load** loading from config.dat CONFIG vpn 1 VPN(1): Object successfully set

Default Value

None

Command Level

global

Related Commands

commit
save
configfile
bootfile1, bootfile2, bootfile3, bootfile4, bootfile5, bootfile6

logout

Exit the IPSX switch.

Syntax	*Example*

With no password set:
`LOGOUT`
	`IPNOS : CONFIG BLADE(1)# logout`
	
	`IPNOS :#`

With password set:
`LOGOUT`
	`IPNOS : CONFIG BLADE(1)# logout`
	
	`Login:`

Default Value

None

Command Level

global

Related Command

sysuser

reboot

Restart the IPSX switch.

Syntax	*Example*
REBOOT [NOW]	IPNOS : CONFIG VPN(1) VR(1.1.1.1)# **reboot** Unsaved configuration will be lost. Reboot anyway? {YES | NO}: **no** IPNOS : CONFIG VPN(1) VR(1.1.1.1)#
REBOOT [NOW]	IPNOS : CONFIG VPN(1) VR(1.1.1.1)# **reboot** Unsaved configuration will be lost. Reboot anyway? {YES | NO}: **yes** REBOOT: Reset successfully IPNOS : CONFIG VPN(1) VR(1.1.1.1)# CoSine Communications 1998 Memory Probe...Done Init Cache.....Done Low Mem Test...Pass Memory Test....Pass L2 Cache Test.. ... IPNOS :#
REBOOT [NOW]	IPNOS : CONFIG VPN(1) VR(1.1.1.1)# **reboot now** REBOOT: Reset successfully IPNOS : CONFIG VPN(1) VR(1.1.1.1)# CoSine Communications 1998 Memory Probe...Done Init Cache.....Done Low Mem Test...Pass ...

Default Value

None

Command Level

global

Related Commands

control-blade-failover
blade-stop-fs
reset-blade
stop-fs

save

Save to configuration file.

Syntax	*Example*
SAVE	IPNOS : CONFIG# **save** saving to config.dat IPNOS : CONFIG#
SAVE [<filename>]	IPNOS : CONFIG# **save ANY_FILENAME** saving to ANY_FILENAME IPNOS : CONFIG#
SAVE [<filename>]	IPNOS : CONFIG# **bootcfg** IPNOS : CONFIG BOOTCFG# **filename NEW_DEFAULT_FILENAME** IPNOS : CONFIG BOOTCFG# **commit** IPNOS : CONFIG# **save** saving to NEW_DEFAULT_FILENAME IPNOS : CONFIG#

Default Value

None

Command Level

global

Related Commands

commit
configfile
load

show

Enter show (read only) mode.

Syntax	*Example*

SHOW	`IPNOS:# show`
	`CHASSIS-TYPE  : IPSX 9000`
	`MIDPLANE-ID  : 00 00 00 00 00 00 00`
	`SYSCONTACT  :`
	`SYSLOCATION  :`
	
	`PACKET-RING:`
	`Ring Status  : PRIMARY ACTIVE`
	`Primary Endpoint Slot  : 0`
	`Protection Endpoint Slot : 0`
	`Primary CRC  : 0x0`
	`Protection CRC  : 0x0`
	
	`BLADE(s):`
	`Slot  : 1`
	`Type  : CONTROL BLADE`
	`State  : MASTER`
	
	`...`
	`IPNOS: SHOW#`

Default Value

None

Command Level

global special

Related Commands

config
display (in all levels)

sysname

Change the IPSX switch name.

Syntax	*Example*
`SYSNAME <name>`	`IPNOS :#` **`sysname CoSine`** `CoSine:#`

Default Value

IPNOS

Command Level

global

Related Commands

vr name

sysuser

Set user login name and password.

Syntax	*Example*

Create a new user account:
`SYSUSER <name>`
	`IPNOS : CONFIG BLADE(1)#` **`sysuser roland`**
	`Enter password for roland`
	`Password:`
	`Please re-type password for roland`
	`Password:`
	`SYSUSER roland: Object successfully set`

Change your user account:
`SYSUSER <name>`
	`IPNOS :#` **`sysuser sam`**
	`Current User's Password:`
	`Enter password for sam`
	`Password:`
	`Please re-type password for sam`
	`Password:`
	`SYSUSER sam: Object successfully set`

Default Value

None

Command Level

global

Related Commands

logout

top

Go to the top of the command tree. All changes are applied.

Syntax	*Example*
TOP	IPNOS : CONFIG VPN(2) VR(2.2.2.2)# **top** VR(2.2.2.2): Object successfully set VPN(2): Object successfully set IPNOS :#

Default Value

Takes you to the top-level prompt in the system.

Command Level

global

Related Commands

up

up

Go up one branch of the command tree. All changes are applied.

Syntax	*Example*

`UP`	`IPNOS : config vpn(2) vr(2.2.2.2) vi(2) # up`
	`IPNOS : config vpn(2) vr(2.2.2.2)# up`
	`IPNOS : config vpn(2)#`

Default Value

Moves you up one level in the system menu.

Command Level

global

Related Commands

top

version

Display current version of IPNOS.

Syntax	*Example*
VERSION [ALL]	IPNOS: CONFIG# **version** CoSine IPNOS 2.1 Copyright (c) 1998-2001 CoSine Communications, Inc. All rights Reserved. Cosine, CoSine logo and all CoSine product names are trademarks of CoSine Communication, Inc. Other brand and product names are trademarks of their respective holders. IPSX Ver 2.1.0 Released 2001.
VERSION [ALL]	IPNOS: CONFIG# **version all** CoSine IPNOS 2.1 Copyright (c) 1998-2001 CoSine Communications, Inc. All rights Reserved. Cosine, CoSine logo and all CoSine product names are trademarks of CoSine Communication, Inc. Other brand and product names are trademarks of their respective holders. IPSX Ver 2.1.0.90 BUILD-20010205010108 Released 2001. Linked on Feb 05 2001 at 01:15:57 Clearcase Label BUILD-20010205010108 set on Mon Feb 5 01:01:08 2001 Contact Info: 1200 Bridge Parkway Redwood City, CA 94065 U.S.A. Phone: (650) 637-4777 Fax: (650) 637-2411

Default Value

None

Command Level

global

Related Commands

None

CHAPTER

3 Config Commands

Use these IPNOS™ software commands to configure the IPSX Service Processing Switch, blades, status panel, and logical interfaces. This chapter describes the following commands:

Command	*Page*
blade	3-3
display	3-5
reset-blade	3-7
tftpget	3-8
tftpput	3-10
bootcfg	3-12
bootfile1, bootfile2, bootfile3, bootfile4, bootfile5, bootfile6	3-12
configfile	3-14
display	3-15
gateway	3-15
host	3-16
port	3-17
savefile	3-17
target	3-18
chassis	3-19
display	3-20
fan-tray	3-21
event-trap	3-25
power-supply	3-25

Command	Page
display	3-26
event-trap	3-27
status-panel	3-27
vsio	3-28
display	3-29
ipnos-diag	3-30
stop-fs	3-31
syscontact	3-32
syslocation	3-33
temp-sensor	3-34
controller	3-35
create	3-36
delete	3-37
display	3-38
filesystem	3-50
copy	3-51
delete	3-52
dir	3-54
rename	3-55
interface	3-57
system-up-time	3-59
time	3-60
verbose-load	3-62
vpn	3-63

blade

Go to the blade command level. The following commands are available.

- "blade-stop-fs" on page 3-3
- "display" on page 3-5
- "reset-blade" on page 3-7
- "tftpget" on page 3-8
- "tftpput" on page 3-10

Syntax	*Example*
`BLADE <slot-number>`	`IPNOS : CONFIG#` **`blade 1`** `IPNOS : CONFIG BLADE(1)#`

Default Value

None

Command Mode

config and show

Related Commands

None

blade-stop-fs

Stop the blade file system. Features will still run, but no changes can be saved.

Syntax	*Example*

`BLADE-STOP-FS <slot-number>`	`IPNOS: CONFIG BLADE(2)#` **`blade-stop-fs 2`**
	`2000-1-1,0:4:43.2: Stopping file system on slot [02]`
	`BLADE-STOP-FS 2: Shutdown successfully`
	
	`IPNOS: CONFIG BLADE(2)#`

Default Value

None

Command Level

blade

Related Command

reboot
reset-blade
stop-fs

display

Show blade or processor engine (PE) configuration.

Syntax	*Example*
DISPLAY	IPNOS: CONFIG BLADE(4)# **display** Slot : 4 Type : CHANNELIZED E3 BLADE State : OPERATIONAL IPSX Loader : Ver 2.0.0.83 BUILD-0008172116 Aug 17 2000 21:48:56 Model Number : 902-00046-01 CLEI Code : CLEI FPGA Version : HW2.0.0.14 Temperature Sensor 1 : 50.0 C Temperature Sensor 2 : 50.5 C Temperature Sensor 3 : 45.0 C Temperature Sensor 4 : 48.5 C ENGINE(s): PE No : 0 Engine Type : MAIN CPU ENGINE Engine State : OPERATIONAL Serial No : 0006 0051 0137 S/W Version : Ver 2.1.0.93 BUILD-20010208014126 PE No : 1 Engine Type : CRYPTO ENGINE Engine State : OPERATIONAL Serial No : 0006 0048 0174 S/W Version : Ver 2.1.0.93 BUILD-20010208014126 PE No : 2 Engine Type : PROCESSOR ENGINE Engine State : OPERATIONAL Serial No : 0006 0052 0378 S/W Version : Ver 2.1.0.93 BUILD-20010208014126 PE No : 3 Engine Type : CRYPTO ENGINE Engine State : OPERATIONAL Serial No : 0006 0048 0402 S/W Version : Ver 2.1.0.93 BUILD-20010208014126

Syntax	*Example*
DISPLAY ENGINE <engine-number>	IPNOS: CONFIG BLADE(4)# **display engine 2** PE No : 2 Engine Type : PROCESSOR ENGINE Engine State : OPERATIONAL Serial No : 0006 0052 0378 S/W Version : Ver 2.1.0.93 BUILD-20010208014126 Loader : CoSine PE#1-N BootLoader Ver 2.0.0.83 BUILD-0008172116 Aug 17 2000 21:20:48 Application : Ver 2.1.0.93 BUILD- 20010208014126 Feb 08 2001 01:49:55 Serial Number : 0006 0052 0378 Assembly Number : 890-00100-07 CPU Type : MIPS RM7000 CPU Version : 0x10 Data Cache Size : 16 KB Instruction Cache Size : 16 KB Secondary Cache Size : 256 KB Level 3 Cache Size : 1024 KB Memory Statistics Memory Size : 256 MB Allocs : 1376638 Alloc Failures : 0 Frees : 1376138 Bad Frees : 0 Bytes Allocated : 85878840 Bytes Freed : 77068336 Free Memory Current : 229 MB Avg over 30 secs : 200 MB Avg over 10 mins : 200 MB Largest Free Block : 229 MB CPU Utilization Current : 0% Avg over 30 secs : 0% Avg over 10 mins : 0%

Default Value

None

Command Level

blade

Related Commands

None

reset-blade

Reboot the current blade. This command will not reboot the control blade.

Syntax	*Example*
RESET-BLADE <slot-number>	IPNOS: CONFIG BLADE(2)# **reset-blade 2** 2000-1-1,0:7:15.1: Stopping file system on slot [02] RESET-BLADE 2: Reset successfully IPNOS: CONFIG BLADE(2)# 2000-1-1,0:7:21.9: Event Log: Blade in slot #2 is REMOVED 2000-1-1,0:7:26.6: Event Log: Blade in slot #2 is PRESENT 2000-1-1,0:7:58.8: Event Log: Blade in slot #2 is LOADING... 2000-1-1,0:7:58.8: Event Log: PE #0 in slot #2 is LOADING... 2000-1-1,0:7:58.9: oloader[slot 2 pe #0]: oloader requesting file d:\ipsx\2.1.0.90\ipnos.o3a 2000-1-1,0:0:4.2: Event Log: PE #0 in slot #2 is REBOOTING... 2000-1-1,0:0:22.2: ATM[2/1:0]: Declaring Link DOWN 2000-1-1,0:0:22.2: ATM[2/1:0]: Declaring Link DOWN 2000-1-1,0:0:22.3: OC3[2/1]: Loss-of-signal detect 2000-1-1,0:0:23.3: Event Log: PE #0 in slot #2 is OPERATIONAL 2000-1-1,0:0:24.4: Event Log: PE #2 in slot #2 is PRESENT 2000-1-1,0:0:24.4: Event Log: PE #2 in slot #2 is LOADING... 2000-1-1,0:0:38.3: Event Log: PE #2 in slot #2 is OPERATIONAL CONFIG 2000-1-1,0:0:53.5: Event Log: Blade in slot #2 is OPERATIONAL INTERFACE ATM 2/1 COMMIT Commit configuration INTERFACE ATM(2/1): Object successfully set INTERFACE ATM 2/1 COMMIT Commit configuration INTERFACE ATM(2/1): Object successfully set CONFIG IPNOS: CONFIG BLADE(2)#
Incorrect blade: RESET-BLADE <slot-number>	IPNOS: CONFIG BLADE(2)# **reset-blade 0** BLADE(0): Can NOT set object Blade Reset: Blade does not exist
Control blade: RESET-BLADE <slot-number>	IPNOS: CONFIG BLADE(2)# **reset-blade 1** BLADE(1): Can NOT set object Blade Reset: Single CB reset is not allowed for now, reboot system instead when needed

Default Value

None

Command Level

blade

Related Commands

blade-stop-fs
reboot
stop-fs

tftpget

Download a file to the blade using Trivial File Transfer Protocol (TFTP). On Windows TFTP servers, use double slashes to indicate directory slashes.

Syntax	*Example*

All servers:
TFTPGET
<host>
<remote-filename>
[<local-filename>]
	`IPNOS: CONFIG BLADE(1)# tftpget 172.20.12.236 sample.txt`
	
	`IPNOS: CONFIG BLADE(1)# up`
	
	`IPNOS: CONFIG# filesystem`
	
	`IPNOS: CONFIG FILESYSTEM# dir`
	`BITMAP.SYS 2560 Jan 1 00:00:00 2000`
	`FLIST.SYS 7168 Jan 1 00:00:00 2000`
	`bootcfg.0 380 Jan 1 00:03:00 2000`
	`ipnos.ctl 1801648 Dec 20 16:12:00 2000`
	`ipnos.crp 806704 Dec 20 16:14:00 2000`
	`passwd.sys 20 Jan 1 00:00:00 2000`
	`config.dat 647 Jan 1 00:04:00 2000`
	`timezone.sys 39 Dec 23 11:08:00 2000`
	`ipsx.dat 647 Jan 1 00:07:00 2000`
	`sample.txt 14 Jan 1 00:00:00 2000`
	
	`10 File(s) 2624512 bytes`
	`7730176 bytes free`

Syntax	Example
Windows servers: TFTPGET <host> <remote-filename> [<local-filename>]	IPNOS: CONFIG BLADE(1)# **tftpget 172.20.12.236 d:\\ipsx\\2.1.0.90\\sample.txt** IPNOS: CONFIG BLADE(1)# **up** IPNOS: CONFIG# **filesystem** IPNOS: CONFIG FILESYSTEM# **dir** BITMAP.SYS 2560 Jan 1 00:00:00 2000 FLIST.SYS 7168 Jan 1 00:00:00 2000 bootcfg.0 380 Jan 1 00:03:00 2000 ipnos.ctl 1801648 Dec 20 16:12:00 2000 ipnos.crp 806704 Dec 20 16:14:00 2000 passwd.sys 20 Jan 1 00:00:00 2000 config.dat 647 Jan 1 00:04:00 2000 timezone.sys 39 Dec 23 11:08:00 2000 ipsx.dat 647 Jan 1 00:07:00 2000 sample.txt 14 Jan 1 00:00:00 2000 10 File(s) 2624512 bytes 7730176 bytes free
Windows servers: TFTPGET <host> <remote-filename> [<local-filename>]	IPNOS: CONFIG BLADE(1)# **tftpget 172.20.12.236 d:\\ipsx\\2.1.0.90\\sample.txt newname.txt** IPNOS: CONFIG BLADE(1)# **up** IPNOS: CONFIG# **filesystem** IPNOS: CONFIG FILESYSTEM# **dir** BITMAP.SYS 2560 Jan 1 00:00:00 2000 FLIST.SYS 7168 Jan 1 00:00:00 2000 bootcfg.0 380 Jan 1 00:03:00 2000 ipnos.ctl 1801648 Dec 20 16:12:00 2000 ipnos.crp 806704 Dec 20 16:14:00 2000 passwd.sys 20 Jan 1 00:00:00 2000 config.dat 647 Jan 1 00:04:00 2000 timezone.sys 39 Dec 23 11:08:00 2000 ipsx.dat 647 Jan 1 00:07:00 2000 sample.txt 14 Jan 1 00:00:00 2000 newname.txt 14 Jan 1 00:06:00 2000 11 File(s) 2625024 bytes 7729664 bytes free

Default Value

None

Command Level

blade

Related Commands

tftpput
filesystem
bootfile1, bootfile2, bootfile3, bootfile4, bootfile5, bootfile6

tftpput

Upload a file to a TFTP server. On Windows TFTP servers, use double slashes to indicate directory slashes.

Syntax	*Example*
All servers: TFTPPUT <host> <local-filename> [<remote-filename>]	IPNOS: CONFIG BLADE(1)# **tftpput 172.20.12.236 sample.txt** IPNOS: CONFIG BLADE(1)#
All servers: TFTPPUT <host> <local-filename> [<remote-filename>]	IPNOS: CONFIG BLADE(1)# **tftpput 172.20.12.236 sample.txt newfile.txt** IPNOS: CONFIG BLADE(1)#
Windows servers: TFTPPUT <host> <local-filename> [<remote-filename>]	IPNOS: CONFIG BLADE(1)# **tftpput 172.20.12.236 sample.txt d:\\ipsx\\2.1.0.90\\newfile.txt** IPNOS: CONFIG BLADE(1)#
UNIX servers: TFTPPUT <host> <local-filename> [<remote-filename>]	IPNOS: CONFIG BLADE(1)# **tftpput 172.20.12.236 sample.txt /home/sam/newfile.txt** IPNOS: CONFIG BLADE(1)#

Default Value

None

Command Level

blade

Related Commands

bootfile1, bootfile2, bootfile3, bootfile4, bootfile5, bootfile6
tftpget

bootcfg

Go to the boot configuration level. The following commands are available.

- "bootfile1, bootfile2, bootfile3, bootfile4, bootfile5, bootfile6" on page 3-12
- "configfile" on page 3-14
- "display" on page 3-15
- "gateway" on page 3-15
- "host" on page 3-16
- "port" on page 3-17
- "savefile" on page 3-17
- "target" on page 3-18

Syntax	*Example*
`BOOTCFG`	`IPNOS : CONFIG#` **`bootcfg`** `IPNOS : CONFIG BOOTCFG(1)#`

Default Value

None

Command Mode

config and show

Related Commands

None

bootfile1, bootfile2, bootfile3, bootfile4, bootfile5, bootfile6

Set the location of boot image files. Include the filename prefix of the images. On Windows TFTP servers, use double slashes to indicate directory slashes.

NOTE: Load the latest boot images from the TFTP server, with **savefile** set to yes, before loading bootfiles from FLSH. Type FLSH in all caps.

Syntax	*Example*

Load from flash memory:
`BOOTFILE1`
`[FLSH][drive:]/path/name`
	`IPNOS: CONFIG BOOTCFG(1)#` **`bootfile1 FLSH/ipnos`**
	
	`IPNOS: CONFIG BOOTCFG(1)#` **`display bootfile1`**
	`BOOTFILE1 : FLSH/ipnos`
	
	`IPNOS: CONFIG BOOTCFG(1)#` **`commit`**
	`Commit configuration`
	`BOOTCFG(1): Object successfully set`

Load from Windows:
`BOOTFILE1`
`[FLSH][drive:]/path/name`
	`IPNOS: CONFIG BOOTCFG(1)#` **`bootfile1`**
	`d:\\ipsx\\rel2.1\\ipnos`
	
	`IPNOS: CONFIG BOOTCFG(1)#` **`display bootfile1`**
	`BOOTFILE1 : d:\ipsx\rel2.1\ipnos`
	
	`IPNOS: CONFIG BOOTCFG(1)#` **`commit`**
	`Commit configuration`
	`BOOTCFG(1): Object successfully set`

Load from UNIX:
`BOOTFILE1`
`[FLSH][drive:]/path/name`
	`IPNOS: CONFIG BOOTCFG(1)#` **`bootfile1`**
	`/usr/bin/ipnos/rel2.1/ipnos`
	
	`IPNOS: CONFIG BOOTCFG(1)#` **`display bootfile1`**
	`BOOTFILE1 : /usr/bin/ipnos/rel2.1/ipnos`
	
	`IPNOS: CONFIG BOOTCFG(1)#` **`commit`**
	`Commit configuration`
	`BOOTCFG(1): Object successfully set`

Clear path:
`NO BOOTFILE1`
	`IPNOS: CONFIG BOOTCFG(1)#` **`no bootfile1`**
	
	`IPNOS: CONFIG BOOTCFG(1)#` **`display bootfile1`**
	`BOOTFILE1 :`
	
	`IPNOS: CONFIG BOOTCFG(1)#` **`commit`**
	`Commit configuration`
	`BOOTCFG(1): Object successfully set`

Default Value

None

Command Level

bootcfg

Related Commands

configfile
host
savefile

configfile

Set the filename of the boot configuration file.

Syntax	*Example*

`CONFIGFILE <name>`	`IPNOS: CONFIG BOOTCFG(1)# configfile config.dat`
	
	`IPNOS: CONFIG BOOTCFG(1)# display configfile`
	`CONFIGFILE  : config.dat`
	
	`IPNOS: CONFIG BOOTCFG(1)# commit`
	`Commit configuration`
	`BOOTCFG(1): Object successfully set`

Default Value

config.dat

Command Level

bootcfg

Related Commands

bootfile1, bootfile2, bootfile3, bootfile4, bootfile5, bootfile6
load
host
save
target
verbose-load

display

The **display** command shows the current boot configuration.

Syntax	*Example*
DISPLAY	IPNOS: CONFIG BOOTCFG(1)# **display** BOOTCFG : 1 HOST : 172.20.12.236 PORT : 1 TARGET : 172.20.12.104|255.255.255.0 GATEWAY : 172.20.12.1 BOOTFILE1 : d:\ipsx\2.1.0.90\ipnos BOOTFILE2 : BOOTFILE3 : BOOTFILE4 : BOOTFILE5 : BOOTFILE6 : SAVEFILE : NO CONFIGFILE : config.dat

Default Value

None

Command Level

bootcfg

Related Commands

None

gateway

Set the default network gateway.

Syntax	*Example*
GATEWAY <address>	IPNOS : CONFIG BOOTCFG(1)# **gateway** 172.20.12.1 IPNOS: CONFIG BOOTCFG(1)# display gateway GATEWAY : 172.20.12.1

Default Value

None

Command Level

bootcfg

Related Commands

target
port
host

host

Set IP address of TFTP server.

Syntax	*Example*

`HOST <address>`	`IPNOS : CONFIG BOOTCFG(1)# host 172.20.12.236`
	
	`IPNOS : CONFIG BOOTCFG(1)# display host`
	`HOST : 172.20.12.236`

Default Value

None

Command Level

bootcfg

Related Commands

bootfile1, bootfile2, bootfile3, bootfile4, bootfile5, bootfile6
gateway
port
savefile
target

port

Set control blade port number for TFTP access.

Syntax	*Example*
PORT <port>	IPNOS : CONFIG BOOTCFG(1)# **port 1** IPNOS : CONFIG BOOTCFG(1)# **display port** PORTID : 1

Default Value

port 1

Command Level

bootcfg

Related Commands

gateway
target
host

savefile

Allow saving copies of boot image files to flash memory on all blades.

Syntax	*Example*
[NO] SAVEFILE	IPNOS: CONFIG BOOTCFG(1)# **savefile** IPNOS: CONFIG BOOTCFG(1)# **display save** SAVEFILE : YES
[NO] SAVEFILE	IPNOS: CONFIG BOOTCFG(1)# **no savefile** IPNOS: CONFIG BOOTCFG(1)# **display save** SAVEFILE : NO

Default Value

keep-image enabled

Command Level

bootcfg

Related Commands

bootfile1, bootfile2, bootfile3, bootfile4, bootfile5, bootfile6
load
save

target

Set IP address of IPSX switch.

Syntax	*Example*

`TARGET <address> <mask>`	`IPNOS : CONFIG BOOTCFG(1)# target 192.168.141.158 255.255.255.0`
	
	`IPNOS : CONFIG BOOTCFG(1)# display target`
	`TARGET : 192.168.141.158|255.255.255.0`

Default Value

None

Command Level

bootcfg

Related Commands

gateway
port
host

chassis

Go to the chassis command level. The following commands are available.

- "display" on page 3-20
- "fan-tray" on page 3-21
- "power-supply" on page 3-25
- "status-panel" on page 3-27
- "stop-fs" on page 3-31
- "syscontact" on page 3-32
- "syslocation" on page 3-33
- "temp-sensor" on page 3-34

Syntax	*Example*
`CHASSIS`	`IPNOS: CONFIG#` **`chassis`** `IPNOS: CONFIG CHASSIS#`

Default Value

None

Command Mode

config and show

Related Commands

None

display

Show IPSX chassis information.

Syntax	*Example*

IPSX 3500:
DISPLAY
	`IPNOS: CONFIG CHASSIS#` **`display`**
	`CHASSIS-TYPE : IPSX 3500`
	`MIDPLANE-ID : 01 FE CB E6 00 00 00`
	`SYSCONTACT : Rosa`
	`SYSLOCATION : Redwood City`
	
	`PACKET-RING:`
	`Ring Status : PRIMARY AND PROTECTION ACTIVE`
	`Primary Endpoint Slot : 4`
	`Protection Endpoint Slot : 1`
	`Primary CRC : 0x0`
	`Protection CRC : 0x0`
	
	`POWER-SUPPLY :`
	`EVENT-TRAP : ENABLE`
	`HW-REVISION : 0`
	`FW-MAJOR-REVISION : 3`
	`FW-MINOR-REVISION : 0`
	
	`STATUS-PANEL :`
	`HW-REVISION : 0`
	`FW-MAJOR-REVISION : 3`
	`FW-MINOR-REVISION : 0`
	
	`FAN-TRAY :`
	`EVENT-TRAP : ENABLE`
	`HW-REVISION : 0`
	`FW-MAJOR-REVISION : 3`
	`FW-MINOR-REVISION : 0`

Syntax	*Example*

IPSX 9000:
DISPLAY
	`IPNOS: CONFIG CHASSIS# display`
	`CHASSIS-TYPE : IPSX 9000`
	`MIDPLANE-ID : 01 F4 0A 68 07 00 00`
	`SYSCONTACT :`
	`SYSLOCATION :`
	
	`PACKET-RING:`
	`Ring Status : PRIMARY ACTIVE`
	`Primary Endpoint Slot : 0`
	`Protection Endpoint Slot : 0`
	`Primary CRC : 0xE1C`
	`Protection CRC : 0x0`
	
	`STATUS-PANEL :`
	`HW-REVISION : 4`
	`FW-MAJOR-REVISION : 2`
	`FW-MINOR-REVISION : 5`
	
	`FAN-TRAY : LOWER`
	`EVENT-TRAP : Invalid value`
	`HW-REVISION : 2`
	`FW-MAJOR-REVISION : 2`
	`FW-MINOR-REVISION : 8`
	
	`FAN-TRAY : UPPER`
	`EVENT-TRAP : Invalid value`
	`HW-REVISION : 3`
	`FW-MAJOR-REVISION : 2`
	`FW-MINOR-REVISION : 8`

Default Value

None

Command Level

chassis

Related Commands

None

fan-tray

Go to IPSX fan tray command level. The following commands are available.

- "display" on page 3-23
- "event-trap" on page 3-25

Syntax	*Example*
IPSX 3500: `FAN-TRAY`	`IPNOS: CONFIG CHASSIS#` **`fan-tray`** `IPNOS: CONFIG CHASSIS FAN-TRAY#`
IPSX 9000: `FAN-TRAY {LOWER | UPPER | FRONT | REAR}`	`IPNOS: CONFIG CHASSIS#` **`fan-tray lower`** `IPNOS: CONFIG CHASSIS FAN-TRAY(LOWER)#`
IPSX 9000: `FAN-TRAY {LOWER | UPPER | FRONT | REAR}`	`IPNOS: CONFIG CHASSIS#` **`fan-tray upper`** `IPNOS: CONFIG CHASSIS FAN-TRAY(UPPER)#`

Default Value

None

Command Level

chassis

Related Commands

None

display

Show fan tray information.

Syntax	*Example*
IPSX 3500: DISPLAY	IPNOS: CONFIG CHASSIS FAN-TRAY# **display** FAN-TRAY : EVENT-TRAP : ENABLE HW-REVISION : 0 FW-MAJOR-REVISION : 3 FW-MINOR-REVISION : 0 FAN(s): 1 2 TEMPERATURE-SENSOR(s): 1
IPSX 9000: DISPLAY	IPNOS: CONFIG CHASSIS FAN-TRAY(LOWER)# **display** FAN-TRAY : LOWER EVENT-TRAP : Invalid value HW-REVISION : 2 FW-MAJOR-REVISION : 2 FW-MINOR-REVISION : 8 FAN(s): 1 2 3 4 5 6 7 8 9 10 11 12 TEMPERATURE-SENSOR(s): 1 2 3 4 IPNOS: CONFIG CHASSIS FAN-TRAY(LOWER)#

Syntax	*Example*
IPSX 9000: DISPLAY	IPNOS: CONFIG CHASSIS FAN-TRAY(UPPER)# **display** FAN-TRAY : UPPER EVENT-TRAP : Invalid value HW-REVISION : 3 FW-MAJOR-REVISION : 2 FW-MINOR-REVISION : 8 FAN(s): 1 2 3 4 5 6 7 8 9 10 11 12 TEMPERATURE-SENSOR(s): 1 2 3 4 IPNOS: CONFIG CHASSIS FAN-TRAY(UPPER)
Fan detail: DISPLAY FAN <number>	IPNOS: CONFIG CHASSIS FAN-TRAY# **display fan 1** FAN-TRAY : FAN : 1 STATE : OK SPEED : 37
Fan temperature sensor: DISPLAY TEMP <number>	IPNOS: CONFIG CHASSIS FAN-TRAY(UPPER)# **display temp 1** TEMPERATURE-SENSOR : 1 STATE : OK TEMPERATURE : 31

Default Value

None

Command Level

fan-tray

Related Commands

None

event-trap

Enable SNMP trap messages from the intelligent fan tray.

Syntax	*Example*
EVENT-TRAP {ENABLE | DISABLE}	IPNOS: CONFIG CHASSIS FAN-TRAY# **event-trap enable** FAN-TRAY: Object successfully set IPNOS: CONFIG CHASSIS FAN-TRAY# **display event-trap** EVENT-TRAP : ENABLE
EVENT-TRAP {ENABLE | DISABLE}	IPNOS: CONFIG CHASSIS FAN-TRAY# **event-trap disable** FAN-TRAY: Object successfully set IPNOS: CONFIG CHASSIS FAN-TRAY# **display event-trap** EVENT-TRAP : DISABLE

Default Value

None

Command Level

fan-tray

Related Commands

None

power-supply

Go to power-supply command level. The following commands are available.

- "display" on page 3-26
- "event-trap" on page 3-27

Syntax	*Example*
POWER-SUPPLY	IPNOS: CONFIG CHASSIS# **power-supply** IPNOS: CONFIG CHASSIS POWER-SUPPLY#

Default Value

None

Command Level

chassis

Related Commands

None

display

Show power supply information.

Syntax	*Example*
IPSX 3500: DISPLAY	IPNOS: CONFIG CHASSIS POWER-SUPPLY# **display** EVENT-TRAP : ENABLE HW-REVISION : 0 FW-MAJOR-REVISION : 3 FW-MINOR-REVISION : 0 POWER-SUPPLY (s): 1 2
IPSX 9000: DISPLAY	IPNOS: CONFIG CHASSIS POWER-SUPPLY# **display** EVENT-TRAP : ENABLE HW-REVISION : 2 FW-MAJOR-REVISION : 6 FW-MINOR-REVISION : 2 POWER-SUPPLY (s): 1 2 3 4
Power Supply Detail: DISPLAY POWER <number>	IPNOS: CONFIG CHASSIS POWER-SUPPLY# **display power 1** POWER-SUPPLY : 1 INPUT : OK OUTPUT : OK

Default Value

None

Command Level

power-supply

Related Commands

None

event-trap

Enable SNMP trap messages from the intelligent power supply.

Syntax	*Example*
EVENT-TRAP {ENABLE | DISABLE}	IPNOS: CONFIG CHASSIS POWER-SUPPLY# **event-trap enable** POWER-SUPPLY: Object successfully set IPNOS: CONFIG CHASSIS POWER-SUPPLY# **commit** Commit configuration POWER-SUPPLY: Object successfully set
EVENT-TRAP {ENABLE | DISABLE}	IPNOS: CONFIG CHASSIS POWER-SUPPLY# **event-trap disable** POWER-SUPPLY: Object successfully set IPNOS: CONFIG CHASSIS POWER-SUPPLY# **commit** Commit configuration POWER-SUPPLY: Object successfully set

Default Value

None

Command Level

power-supply

Related Commands

None

status-panel

Go to status-panel command level. The following commands are available.

- "display" on page 3-28
- "vsio" on page 3-28

Syntax	*Example*
STATUS-PANEL	IPNOS: CONFIG CHASSIS# **status-panel** IPNOS: CONFIG CHASSIS STATUS-PANEL#

Default Value

None

Command Level

chassis

Related Commands

None

display

Show status panel information.

Syntax	*Example*

DISPLAY	`IPNOS: CONFIG CHASSIS STATUS-PANEL# display`
	`HW-REVISION : 0`
	`FW-MAJOR-REVISION : 3`
	`FW-MINOR-REVISION : 0`

Default Value

None

Command Level

status-panel

Related Commands

None

vsio

Go to Virtual Serial Input Output (VSIO) command level. The following commands are available.

- "display" on page 3-29
- "ipnos-diag" on page 3-30

Syntax	Example
VSIO	IPNOS: CONFIG CHASSIS STATUS-PANEL# **vsio** IPNOS: CONFIG CHASSIS STATUS-PANEL VSIO#

Default Value

None

Command Level

status-panel

Related Commands

None

display

Show status of modem port.

Syntax	Example
With the modem port disconnected: DISPLAY	IPNOS: CONFIG CHASSIS STATUS-PANEL VSIO# **display** There is nothing connected to the status panel modem port. IPNOS: CONFIG CHASSIS STATUS-PANEL VSIO#
With a the modem port connected: DISPLAY	IPNOS: CONFIG CHASSIS STATUS-PANEL VSIO# **display** The status panel modem port is connected to SYMTAB on blade 4, processor-ID 0 IPNOS: CONFIG CHASSIS STATUS-PANEL VSIO#

Default Value

None

Command Level

vsio

Related Commands

ipnos-diag

ipnos-diag

Connect modem port to a processor engine on a specific blade. Table 3-1 describes the parameters available.

Table 3-1 Parameters for ipnos-diag

Parameter	*Description*
Connect	Connect to a specific processor. Choose from blades 1–26 and processors 0–3.
Disconnect	Disconnect from the processor engine.
Who-connect	Show status of modem port.

Syntax	*Example*
IPNOS-DIAG CONNECT <blade> <processor-ID>	IPNOS: CONFIG CHASSIS STATUS-PANEL VSIO# **ipnos-diag connect 4 0** IPNOS: CONFIG CHASSIS STATUS-PANEL VSIO# **commit** Commit configuration VSIO: Status panel modem connected. IPNOS: CONFIG CHASSIS STATUS-PANEL# **vsio** IPNOS: CONFIG CHASSIS STATUS-PANEL VSIO# **display** The status panel modem port is connected to SYMTAB on blade 4, processor-ID 0.
IPNOS-DIAG DISCONNECT	IPNOS: CONFIG CHASSIS STATUS-PANEL VSIO# **ipnos-diag disconnect** IPNOS: CONFIG CHASSIS STATUS-PANEL VSIO# **commit** Commit configuration VSIO: Status panel modem port is disconnected. IPNOS: CONFIG CHASSIS STATUS-PANEL# **vsio** IPNOS: CONFIG CHASSIS STATUS-PANEL VSIO# **display** There is nothing connected to the status panel modem port.
With modem port connected: IPNOS-DIAG WHO-CONNECT	IPNOS: CONFIG CHASSIS STATUS-PANEL VSIO# **ipnos-diag who-connect** The status panel modem port is connected to SYMTAB on blade 4, processor-ID 0.
With modem port disconnected: IPNOS-DIAG WHO-CONNECT	IPNOS: CONFIG CHASSIS STATUS-PANEL VSIO# **ipnos-diag who-connect** There is nothing connected to the status panel modem port.

Default Value

None

Command Level

vsio

Related Commands

vsio display

stop-fs

Stop file systems on all blades. Features will still run, but no changes can be saved.

Syntax	*Example*
STOP-FS [NOW]	IPNOS: CONFIG CHASSIS# **stop-fs** Unsaved configuration will be lost. stop file-system anyway? {YES | NO}: **yes** 2000-1-1,0:4:51.3: Stopping file system on slot [04] 2000-1-1,0:4:52.3: Stopping file system on slot [02] STOP-FS: Chassis file-system is stopped successfully 2000-1-1,0:4:53.3: Stopping file system on slot [01] IPNOS: CONFIG CHASSIS#

Default Value

None

Command Level

chassis

Related Commands

blade-stop-fs
reboot
reset-blade

syscontact

Assign IPSX chassis' system administrator.

Syntax	*Example*
SYSCONTACT <contact-name>	IPNOS: CONFIG CHASSIS# **syscontact John Smith** CHASSIS: Object successfully set IPNOS: CONFIG CHASSIS# **display** CHASSIS-TYPE : IPSX 9000 MIDPLANE-ID : 00 00 00 00 00 00 00 SYSCONTACT : John Smith SYSLOCATION : PACKET-RING: Ring Status : PRIMARY ACTIVE Primary Endpoint Slot : 0 Protection Endpoint Slot : 0 Primary CRC : 0x0 Protection CRC : 0x0
[NO] SYSCONTACT	IPNOS: CONFIG CHASSIS# **no syscontact** CHASSIS: Object successfully set IPNOS: CONFIG CHASSIS# **display** CHASSIS-TYPE : IPSX 9000 MIDPLANE-ID : 00 00 00 00 00 00 00 SYSCONTACT : SYSLOCATION : PACKET-RING: Ring Status : PRIMARY ACTIVE Primary Endpoint Slot : 0 Protection Endpoint Slot : 0 Primary CRC : 0x0 Protection CRC : 0x0

Default Value

None

Command Level

chassis

Related Commands

syslocation

syslocation

Assign IPSX chassis' physical location.

Syntax	*Example*
SYSLOCATION <contact-name>	IPNOS: CONFIG CHASSIS# **syslocation 1200 Main Street, anytown, USA 12345-6789** CHASSIS: Object successfully set IPNOS: CONFIG CHASSIS# **display** CHASSIS-TYPE : IPSX 9000 MIDPLANE-ID : 00 00 00 00 00 00 00 SYSCONTACT : John Smith SYSLOCATION : 1200 Main Street, anytown, USA 12345-6789 PACKET-RING: Ring Status : PRIMARY ACTIVE Primary Endpoint Slot : 0 Protection Endpoint Slot : 0 Primary CRC : 0x0 Protection CRC : 0x0
[NO] SYSLOCATION	IPNOS: CONFIG CHASSIS# **no syslocation** CHASSIS: Object successfully set IPNOS: CONFIG CHASSIS# **display** CHASSIS-TYPE : IPSX 9000 MIDPLANE-ID : 00 00 00 00 00 00 00 SYSCONTACT : John Smith SYSLOCATION : PACKET-RING: Ring Status : PRIMARY ACTIVE Primary Endpoint Slot : 0 Protection Endpoint Slot : 0 Primary CRC : 0x0 Protection CRC : 0x0

Default Value

None

Command Level

chassis

Related Commands

syscontact

temp-sensor

Go to temperature sensor command level. The following command is available.

❖ "display" on page 3-34

Syntax	*Example*

IPSX 3500:
TEMP-SENSOR
	`IPNOS: CONFIG CHASSIS# temp-sensor`
	
	`IPNOS: CONFIG CHASSIS TEMP-SENSOR#`

Default Value

None

Command Level

chassis

Related Commands

None

display

Show temperature sensor information.

Syntax	*Example*

IPSX 3500
TEMP-SENSOR
	`IPNOS: CONFIG CHASSIS TEMP-SENSOR# display`
	`TEMP-SENSOR :`
	`STATE : OK`
	`TEMPERATURE : 22`

Default Value

None

Command Level

chassis

Related Commands

None

controller

Go to the DS3 or E3 command level. See “Channelized DS3 Blade Commands” in Chapter 5, and “Channelized E3 Blade Commands” in Chapter 7.

Syntax	*Example*
`CONTROLLER T3 {<slot>/<port>}`	`IPNOS : CONFIG # controller T3 2/1` `IPNOS : CONFIG CONTROLLER(2/1)#`
`CONTROLLER E3 {<slot>/<port>}`	`IPNOS : CONFIG # controller E3 4/1` `IPNOS : CONFIG CONTROLLER(4/1)#`

Default Value

None

Command Mode

config and show

Related Commands

interface

create

Create a VPN, VR, or VI. See “VPN Commands” in Chapter 15, “VR Commands” in Chapter 16, “VI Commands” in Chapter 21.

Syntax	*Example*
`CREATE {VPN | VR | VI} <id>`	`IPNOS : CONFIG#` **`create vpn 5`** `IPNOS : CONFIG VPN(5)#`

Default Value

None

Command Mode

config

Related Commands

delete
create vr
create vi

delete

Delete a VPN, VR, or VI. See “VPN Commands” in Chapter 15, “VR Commands” in Chapter 16, “VI Commands” in Chapter 21.

Syntax	*Example*
DELETE {VPN | VR | VI} <id>	IPNOS : CONFIG VPN(5) VR(1.1.1.1) VI(1)# **up** VI(1): Object successfully set IPNOS : CONFIG VPN(5) VR(1.1.1.1)# **delete vi 1** VR(1.1.1.1): Object successfully set VI 1: Object successfully deleted IPNOS : CONFIG VPN(5) VR(1.1.1.1)# **up** VR(1.1.1.1): Object successfully set IPNOS : CONFIG VPN(5)# **delete vr 1.1.1.1** VPN(5): Object successfully set VR 1.1.1.1: Object successfully deleted IPNOS : CONFIG VPN(5)# **up** VPN(5): Object successfully set IPNOS : CONFIG# **delete vpn 5** VPN 5: Object successfully deleted IPNOS : CONFIG#

Default Value

None

Command Mode

config

Related Commands

create

display

Use the **display** command to display the current IPNOS software configuration. You can also limit the display to only the items you want to see.

Syntax	*Example*
DISPLAY	

```
IPNOS: CONFIG# display
CHASSIS-TYPE    : IPSX 9000
MIDPLANE-ID     : 00 00 00 00 00 00 00
SYSCONTACT      :
SYSLOCATION     :

PACKET-RING:
Ring Status              : PRIMARY ACTIVE
Primary Endpoint Slot    : 0
Protection Endpoint Slot : 0
Primary CRC              : 0x0
Protection CRC           : 0x0

BLADE(s):
Slot                   : 1
Type                   : CONTROL BLADE
State                  : MASTER

Slot                   : 2
Type                   : OC-3 SMF BLADE (ATM)
State                  : OPERATIONAL

Slot                   : 4
Type                   : CHANNELIZED E3 BLADE
State                  : OPERATIONAL


VPN(s):
VPN ID    | VPN NAME         | ADMIN STATE
1000000   |                  | ENABLE

List is empty

CHANNELIZED E3(s):
4/1             4/2             4/3
```

Syntax	Example
DISPLAY (continued)	UNCHANNELIZED T3(s): List is empty UNCHANNELIZED E3(s): List is empty ETHERNET INTERFACE(s): 1/1 1/2 1/3 1/4 ATM INTERFACE(s): 2/1 POS INTERFACE(s): List is empty FRAME-RELAY PVC(s): List is empty ATM CONNECTION(s): List is empty ETHERNET VLAN(s): List is empty BUNDLE INTERFACE(s): List is empty BOOTCFG(s): 1 TIME : 2000-1-1,0:6:27.2,+161:0 SYSTEM-UP-TIME : 0 day(s), 2 hour(s), 15 minute(s), 34 second(s) VERBOSE-LOAD : ENABLE IPNOS: CONFIG#

Syntax	*Example*
DISPLAY BLADE	IPNOS: CONFIG# **display blade** BLADE(s): Slot : 1 Type : CONTROL BLADE State : MASTER Slot : 2 Type : OC-3 SMF BLADE (ATM) State : OPERATIONAL Slot : 4 Type : CHANNELIZED E3 BLADE State : OPERATIONAL
DISPLAY BOOTCFG	IPNOS: CONFIG# **display bootcfg** BOOTCFG(s): 1
DISPLAY CHASSIS	IPNOS: CONFIG# **display chassis** CHASSIS-TYPE : IPSX 9000 MIDPLANE-ID : 00 00 00 00 00 00 00 SYSCONTACT : SYSLOCATION : PACKET-RING: Ring Status : PRIMARY ACTIVE Primary Endpoint Slot : 0 Protection Endpoint Slot : 0 Primary CRC : 0x0 Protection CRC : 0x0
DISPLAY CONTROLLER	IPNOS: CONFIG# **display controller** List is empty CHANNELIZED E3(s): 4/1 4/2 4/3

Syntax	*Example*
DISPLAY DETAIL	IPNOS: CONFIG# **display detail** CHASSIS-TYPE : IPSX 9000 MIDPLANE-ID : 00 00 00 00 00 00 00 SYSCONTACT : SYSLOCATION : PACKET-RING: Ring Status : PRIMARY ACTIVE Primary Endpoint Slot : 0 Protection Endpoint Slot : 0 Primary CRC : 0x0 Protection CRC : 0x0 BLADE(s): SLOT : 1 TYPE : CONTROL BLADE MODEL : 902-00039-01 STATE : MASTER PE(s) : PE No : 0 Engine Type : MAIN CPU ENGINE Engine State : OPERATIONAL Serial No : 0006 0043 0748 F/W Version : HW2.0.0.14 S/W Version : Ver 2.1.0.90 BUILD-20010205010108 PE No : 1 Engine Type : CRYPTO ENGINE Engine State : OPERATIONAL Serial No : 0006 0044 0037 F/W Version : HW2.0.0.14 S/W Version : Ver 2.1.0.90 BUILD-20010205010108 SLOT : 2 TYPE : OC-3 SMF BLADE (ATM) MODEL : 902-xxxxxxxx STATE : OPERATIONAL PE(s) : PE No : 0 Engine Type : MAIN CPU ENGINE Engine State : OPERATIONAL Serial No : 0002 0014 0006 F/W Version : HW2.0.0.14 S/W Version : Ver 2.1.0.90 BUILD-20010205010108

Syntax	Example
DISPLAY DETAIL (continued)	

```
 PE No        : 2
        Engine Type  : PROCESSOR ENGINE
        Engine State : OPERATIONAL
        Serial No    : 0002 0024 0117
        F/W Version  : HW2.0.0.14
        S/W Version  : Ver 2.1.0.90 BUILD-20010205010108

SLOT    : 4
TYPE    : CHANNELIZED E3 BLADE
MODEL   : 902-00046-01
STATE   : OPERATIONAL
PE(s)   :

        PE No        : 0
        Engine Type  : MAIN CPU ENGINE
        Engine State : OPERATIONAL
        Serial No    : 0006 0051 0137
        F/W Version  : HW2.0.0.14
        S/W Version  : Ver 2.1.0.90 BUILD-20010205010108

        PE No        : 1
        Engine Type  : CRYPTO ENGINE
        Engine State : OPERATIONAL
        Serial No    : 0006 0048 0174
        F/W Version  : HW2.0.0.14
        S/W Version  : Ver 2.1.0.90 BUILD-20010205010108

        PE No        : 2
        Engine Type  : PROCESSOR ENGINE
        Engine State : OPERATIONAL
        Serial No    : 0006 0052 0378
        F/W Version  : HW2.0.0.14
        S/W Version  : Ver 2.1.0.90 BUILD-20010205010108

        PE No        : 3
        Engine Type  : CRYPTO ENGINE
        Engine State : OPERATIONAL
        Serial No    : 0006 0048 0402
        F/W Version  : HW2.0.0.14
        S/W Version  : Ver 2.1.0.90 BUILD-20010205010108
```

Syntax	Example
DISPLAY DETAIL (continued)	

```
VPN(s):

VPN 1000000

    VR 165.165.1.129

       VI ID| IP ADDRESS/MASK                | CONNECTED-TO
       1    | 165.165.1.129/255.255.255.255  | LOOPBACK
       99   | 1.1.1.2/255.255.255.0          | V
1000000/165.165.100.100/101

    VR 165.165.1.130

       VI ID| IP ADDRESS/MASK                | CONNECTED-TO
       1    | 165.165.1.130/255.255.255.255  | LOOPBACK
       99   | 1.1.2.2/255.255.255.0          | V
1000000/165.165.100.100/102

    VR 165.165.1.132

       VI ID| IP ADDRESS/MASK                | CONNECTED-TO
       1    | 165.165.1.132/255.255.255.255  | LOOPBACK
       99   | 1.1.4.2/255.255.255.0          | V
1000000/165.165.100.100/104

    VR 165.165.100.100

       VI ID| IP ADDRESS/MASK                | CONNECTED-TO
       1    | 165.165.100.100/255.255.255.255| LOOPBACK
       98   | 172.20.12.104/255.255.255.0    | L 1/1:0:0
       101  | 1.1.1.1/255.255.255.0          | V
1000000/165.165.1.129/99
       102  | 1.1.2.1/255.255.255.0          | V
1000000/165.165.1.130/99
       104  | 1.1.4.1/255.255.255.0          | V
1000000/165.165.1.132/99

CHANNELIZED T3(s):
List is empty

CHANNELIZED E3(s):

SLOT/PORT    : 4/1
ADMIN STATUS : ENABLE
OPER STATUS  : DOWN
CHANNEL(s)   :
List is empty
```

Syntax	*Example*
DISPLAY DETAIL (continued)	

```
SLOT/PORT    : 4/2
ADMIN STATUS : ENABLE
OPER STATUS  : DOWN
CHANNEL(s)   :
List is empty

SLOT/PORT    : 4/3
ADMIN STATUS : ENABLE
OPER STATUS  : DOWN
CHANNEL(s)   :
List is empty

UNCHANNELIZED T3(s):
List is empty

UNCHANNELIZED E3(s):
List is empty

ETHERNET INTERFACE(s):

        SLOT/PORT| SPEED    | ADMIN-STATUS| OPER-STATUS
        1/1      | 100 Mbps | ENABLE      | UP
        1/2      | 10 Mbps  | DISABLE     | DOWN
        1/3      | 10 Mbps  | DISABLE     | DOWN
        1/4      | 10 Mbps  | DISABLE     | DOWN

ATM INTERFACE(s):

        SLOT/PORT| ADMIN-STATUS| OPER-STATUS
        2/1      | DISABLE     | UP

POS INTERFACE(s):
List is empty

FRAME-RELAY PVC(s):
List is empty

ATM CONNECTION(s):
List is empty

ETHERNET VLAN(s):
List is empty

BUNDLE INTERFACE(s):
List is empty
```

Syntax	Example
DISPLAY DETAIL (continued)	BOOTCFG : 1 HOST : 172.20.12.236 PORT : 1 TARGET : 172.20.12.104|255.255.255.0 GATEWAY : 172.20.12.1 BOOTFILE1 : d:\ipsx\2.1.0.90\ipnos BOOTFILE2 : BOOTFILE3 : BOOTFILE4 : BOOTFILE5 : BOOTFILE6 : SAVEFILE : NO CONFIGFILE : TIME : 2000-1-1,0:1:59.2,+808335664:1550413934 SYSTEM-UP-TIME : 0 day(s), 2 hour(s), 19 minute(s), 6 second(s) VERBOSE-LOAD : ENABLE IPNOS: CONFIG#

Syntax	*Example*
DISPLAY INTERFACE	IPNOS: CONFIG# **display interface** UNCHANNELIZED T3(s): List is empty UNCHANNELIZED E3(s): List is empty ETHERNET INTERFACE(s): 1/1 1/2 1/3 1/4 ATM INTERFACE(s): 2/1 POS INTERFACE(s): List is empty FRAME-RELAY PVC(s): List is empty ATM CONNECTION(s): List is empty ETHERNET VLAN(s): List is empty BUNDLE INTERFACE(s): List is empty IPNOS: CONFIG#
DISPLAY SYSTEM-UP-TIME	IPNOS: CONFIG# **display system-up-time** SYSTEM-UP-TIME : 0 day(s), 2 hour(s), 29 minute(s), 28 second(s) IPNOS: CONFIG#
DISPLAY TIME	IPNOS: CONFIG# **display time** TIME : 2000-1-1,0:5:9.3,+-1911505980:1
DISPLAY VERBOSE-LOAD	IPNOS: CONFIG# **display verbose-load** VERBOSE-LOAD : ENABLE

Syntax	*Example*
DISPLAY VPN	IPNOS: CONFIG# **display vpn** VPN(s): VPN ID | VPN NAME | ADMIN STATE 1000000 | | ENABLE

Syntax	*Example*
DISPLAY BLADE DETAIL	IPNOS: CONFIG# **display blade detail** BLADE(s): SLOT : 1 TYPE : CONTROL BLADE MODEL : 902-xxxxxxxx STATE : MASTER PE(s) : PE No : 0 Engine Type : MAIN CPU ENGINE Engine State : OPERATIONAL Serial No : 0002 0010 0148 F/W Version : HW2.0.0.14 S/W Version : Ver 2.1.0.61 BUILD-20010108094211 SLOT : 2 TYPE : CHANNELIZED T3 BLADE MODEL : 902-xxxxxxxx STATE : OPERATIONAL PE(s) : PE No : 0 Engine Type : MAIN CPU ENGINE Engine State : OPERATIONAL Serial No : 0002-0006-0119 F/W Version : HW2.0.0.14 S/W Version : Ver 2.1.0.61 BUILD-20010108094211 PE No : 1 Engine Type : CRYPTO ENGINE Engine State : OPERATIONAL Serial No : 0002 0008 0138 F/W Version : HW2.0.0.14 S/W Version : Ver 2.1.0.61 BUILD-20010108094211 PE No : 2 Engine Type : PROCESSOR ENGINE Engine State : OPERATIONAL Serial No : 0002 0029 0053 F/W Version : HW2.0.0.14 S/W Version : Ver 2.1.0.61 BUILD-20010108094211

Syntax	Example
DISPLAY BLADE DETAIL (continued)	SLOT : 4 TYPE : OC-3 SMF BLADE (ATM) MODEL : 902-xxxxxxxx STATE : OPERATIONAL PE(s) : PE No : 0 Engine Type : MAIN CPU ENGINE Engine State : OPERATIONAL Serial No : 0002 0014 0028 F/W Version : HW2.0.0.14 S/W Version : Ver 2.1.0.61 BUILD-20010108094211 PE No : 2 Engine Type : PROCESSOR ENGINE Engine State : OPERATIONAL Serial No : 0002 0024 0059 F/W Version : HW2.0.0.14 S/W Version : Ver 2.1.0.61 BUILD-20010108094211

Default Value

None

Command Mode

config and show

Related Commands

None

filesystem

Go to the control blade filesystem command level. The following commands are available.

- "copy" on page 3-51
- "delete" on page 3-52
- "dir" on page 3-54
- "rename" on page 3-55
- "show" on page 3-56

Syntax	*Example*
`FILESYSTEM`	`IPNOS : CONFIG#` **`filesystem`** `IPNOS : CONFIG FILESYSTEM#`

Default Value

None

Command Mode

config and show

Related Commands

None

copy

Copy a file.

Syntax	*Example*
COPY <src-filename> <dst-filename>	IPNOS: CONFIG FILESYSTEM# **copy passwd.sys oldpasswd.sys** IPNOS: CONFIG FILESYSTEM# dir BITMAP.SYS 2560 Jan 1 00:00:00 2000 FLIST.SYS 7168 Jan 1 00:00:00 2000 bootcfg.0 380 Jan 1 00:03:00 2000 ipnos.ctl 1801648 Dec 20 16:12:00 2000 ipnos.crp 806704 Dec 20 16:14:00 2000 passwd.sys 20 Jan 1 00:00:00 2000 config.dat 647 Jan 1 00:04:00 2000 timezone.sys 39 Dec 23 11:08:00 2000 oldpasswd.sys 20 Jan 1 00:01:00 2000 9 File(s) 2623488 bytes 7731200 bytes free

Default Value

None

Command Mode

filesystem

Related Commands

None

delete

Delete a file. Use the asterisk * wildcard to remove multiple files at one time.

Syntax	*Example*
DELETE <filename>	IPNOS: CONFIG FILESYSTEM# **dir** BITMAP.SYS 2560 Jan 1 00:00:00 2000 FLIST.SYS 7168 Jan 1 00:00:00 2000 bootcfg.0 380 Jan 1 00:03:00 2000 ipnos.ctl 1801648 Dec 20 16:12:00 2000 ipnos.crp 806704 Dec 20 16:14:00 2000 passwd.sys 20 Jan 1 00:00:00 2000 config.dat 647 Jan 1 00:04:00 2000 timezone.sys 39 Dec 23 11:08:00 2000 oldpasswd.sys 20 Jan 1 00:01:00 2000 9 File(s) 2623488 bytes 7731200 bytes free IPNOS: CONFIG FILESYSTEM# **delete oldpasswd.sys** IPNOS: CONFIG FILESYSTEM# **dir** BITMAP.SYS 2560 Jan 1 00:00:00 2000 FLIST.SYS 7168 Jan 1 00:00:00 2000 bootcfg.0 380 Jan 1 00:03:00 2000 ipnos.ctl 1801648 Dec 20 16:12:00 2000 ipnos.crp 806704 Dec 20 16:14:00 2000 passwd.sys 20 Jan 1 00:00:00 2000 config.dat 647 Jan 1 00:04:00 2000 timezone.sys 39 Dec 23 11:08:00 2000 8 File(s) 2622976 bytes 7731712 bytes free

Syntax	*Example*
DELETE <filename>*	IPNOS: CONFIG FILESYSTEM# **dir** BITMAP.SYS 2560 Jan 1 00:00:00 2000 FLIST.SYS 7168 Jan 1 00:00:00 2000 bootcfg.0 380 Jan 1 00:03:00 2000 ipnos.ctl 1801648 Dec 20 16:12:00 2000 ipnos.crp 806704 Dec 20 16:14:00 2000 passwd.sys 20 Jan 1 00:00:00 2000 config.dat 647 Jan 1 00:04:00 2000 timezone.sys 39 Dec 23 11:08:00 2000 file1.txt 20 Jan 1 00:04:00 2000 file2.txt 20 Jan 1 00:04:00 2000 file3.txt 20 Jan 1 00:04:00 2000 11 File(s) 2624512 bytes 7730176 bytes free IPNOS: CONFIG FILESYSTEM# **del file*** deleting file1.txt deleting file2.txt deleting file3.txt 3 file(s) deleted IPNOS: CONFIG FILESYSTEM# **dir** BITMAP.SYS 2560 Jan 1 00:00:00 2000 FLIST.SYS 7168 Jan 1 00:00:00 2000 bootcfg.0 380 Jan 1 00:03:00 2000 ipnos.ctl 1801648 Dec 20 16:12:00 2000 ipnos.crp 806704 Dec 20 16:14:00 2000 passwd.sys 20 Jan 1 00:00:00 2000 config.dat 647 Jan 1 00:04:00 2000 timezone.sys 39 Dec 23 11:08:00 2000 8 File(s) 2622976 bytes 7731712 bytes free

Default Value

None

Command Mode

filesystem

Related Commands

None

dir

List the files. Table 3-2 describes typical files.

Table 3-2 IPNOS File Descriptions

File	*Description*
bitmap.sys	System file.
flist.sys	System file.
config.dat	IPNOS configuration file.
bootcfg.0	Boot image configuration file.
passwd.sys	Password file.
ipnos.*	Boot image file. The following is a list of images. • *ipnos.ctl*—Control blade • *ipnos.prc*—Processor engine card • *ipnos.crp*—Crypto engine card • *ipnos.d3c*—Channelized DS3 blade • *ipnos.d3u*—Unchannelized DS3 blade • *ipnos.e3c*—Channelized E3 blade • *ipnos.e3u*—Unchannelized E3 blade • *ipnos.o3p*—OC-3c blade for packet over SONET (POS) • *ipnos.o3a*—OC-3c blade for asynchronous transfer mode (ATM). (LED and laser versions) • *ipnos.pre*—Processor blade
timezone	Universal time code (UTC) file.

Syntax	*Example*

DIR	`IPNOS: CONFIG FILESYSTEM# dir`
	`BITMAP.SYS 2560 Jan 1 00:00:00 2000`
	`FLIST.SYS 7168 Jan 1 00:00:00 2000`
	`bootcfg.0 380 Jan 1 00:03:00 2000`
	`ipnos.ctl 1801648 Dec 20 16:12:00 2000`
	`ipnos.crp 806704 Dec 20 16:14:00 2000`
	`passwd.sys 20 Jan 1 00:00:00 2000`
	`config.dat 647 Jan 1 00:04:00 2000`
	`timezone.sys 39 Dec 23 11:08:00 2000`
	
	`8 File(s) 2622976 bytes`
	`7731712 bytes free`

Default Value

None

Command Mode

filesystem

Related Commands

None

rename

Rename a file.

Syntax	*Example*
RENAME <old-filename> <new-filename>	IPNOS: CONFIG FILESYSTEM# **dir** BITMAP.SYS 2560 Jan 1 00:00:00 2000 FLIST.SYS 7168 Jan 1 00:00:00 2000 bootcfg.0 380 Jan 1 00:03:00 2000 ipnos.ctl 1801648 Dec 20 16:12:00 2000 ipnos.crp 806704 Dec 20 16:14:00 2000 passwd.sys 20 Jan 1 00:00:00 2000 config.dat 647 Jan 1 00:04:00 2000 timezone.sys 39 Dec 23 11:08:00 2000 cosine.dat 647 Jan 1 00:07:00 2000 9 File(s) 2624000 bytes 7730688 bytes free IPNOS: CONFIG FILESYSTEM# **rename cosine.dat ipsx.dat** IPNOS: CONFIG FILESYSTEM# **dir** BITMAP.SYS 2560 Jan 1 00:00:00 2000 FLIST.SYS 7168 Jan 1 00:00:00 2000 bootcfg.0 380 Jan 1 00:03:00 2000 ipnos.ctl 1801648 Dec 20 16:12:00 2000 ipnos.crp 806704 Dec 20 16:14:00 2000 passwd.sys 20 Jan 1 00:00:00 2000 config.dat 647 Jan 1 00:04:00 2000 timezone.sys 39 Dec 23 11:08:00 2000 ipsx.dat 647 Jan 1 00:07:00 2000 9 File(s) 2624000 bytes 7730688 bytes free

Default Value

None

Command Mode

filesystem

Related Commands

None

show

Display contents of an ASCII text file.

Syntax	*Example*

`SHOW FILE <filename>`	`IPNOS: CONFIG FILESYSTEM#` **`show file timezone.sys`**
	`utc_dir:-`
	`utc_hours:8`
	`utc_mins:0`
	`DST:0`

Default Value

None

Command Mode

filesystem

Related Commands

None

interface

Go to the hardware or link layer interface command level. See the following chapters for commands available.

- Chapter 4, “Control and Ethernet Blade Commands”
- Chapter 5, “Channelized DS3 Blade Commands”
- Chapter 6, “Unchannelized DS3 Blade Commands”
- Chapter 7, “Channelized E3 Blade Commands”
- Chapter 8, “Unchannelized E3 Blade Commands”
- Chapter 9, “OC-3c Commands”
- Chapter 10, “ATM Commands”
- Chapter 11, “POS Commands”
- Chapter 12, “Frame Relay Commands”
- Chapter 14, “PPP Commands”

Table 3-3 explains the differences between interfaces.

Table 3-3 Interface Modes Compared

If you use this command...	*...you enter the interface level for...*
`INTERFACE ATM <slot>/<port>`	OC-3c ATM blade.
`[NO] INTERFACE ATM <slot>/<port>:<channel>:<subif>`	ATM private virtual circuit (PVC).
`[NO] INTERFACE BUNDLE <slot>/<port>`	Multi-Link PPP (MLPPP).
`INTERFACE ETHERNET <slot>/<port>`	Control and fast ethernet blades.
`[NO] INTERFACE ETHERNET <slot>/<port>:<channel>:<subif>`	Fast ethernet virtual local area network (VLAN) subinterface.
`INTERFACE POS <slot>/<port>`	OC-3c POS blade.
`[NO] INTERFACE POS {slot/port:channel:subif}`	POS subinterface.
`INTERFACE SERIAL <slot>/<port>`	Unchannelized DS3 or E3 blade.
`INTERFACE SERIAL <slot>/<port>:<channel>`	Channelized or unchannelized DS3 or E3 blade encapsulation. Unchannelized blades have channel “0.”
`[NO] INTERFACE SERIAL <slot>/<port>:<channel>:<subif>`	Frame Relay dedicated link connection identifier (DLCI).

Syntax	*Example*

OC-3c ATM blade configuration:
`INTERFACE ATM <slot>/<port>`
	`IPNOS : CONFIG# interface atm 2/1`
	
	`IPNOS : CONFIG INTERFACE ATM(2/1)#`

ATM PVC configuration:
`INTERFACE ATM <slot>/<port>:<channel>:<subif>`
	`IPNOS : CONFIG# interface atm 2/1:0:3`
	
	`IPNOS : CONFIG INTERFACE ATM(2/1:0:3)#`

MLPPP configuration:
`[NO] INTERFACE BUNDLE <slot>/<port>`
	`IPNOS: CONFIG# inter bundle 2/1`
	
	`IPNOS: CONFIG INTERFACE BUNDLE(2/1)#`

Control blade configuration:
`INTERFACE ETHERNET <slot>/<port>`
	`IPNOS : CONFIG# interface ethernet 4/1`
	
	`IPNOS : CONFIG INTERFACE ETHERNET(4/1)#`

Unchannelized DS3 blade configuration:
`INTERFACE SERIAL <slot>/<port>`
	`IPNOS : CONFIG# interface serial 2/1`
	
	`IPNOS : CONFIG INTERFACE SERIAL(2/1)#`

Channelized DS3 blade encapsulation:
`INTERFACE SERIAL <slot/port>}:<channel>`
	`IPNOS : CONFIG# interface serial 2/1:1`
	
	`IPNOS : CONFIG INTERFACE SERIAL(2/1:1)#`

Frame Relay PVC configuration:
`INTERFACE SERIAL <slot/port>:<channel>:<subif>`
	`IPNOS : CONFIG# interface serial 2/1:1:1`
	
	`IPNOS : CONFIG INTERFACE SERIAL(2/1:1:1)#`

Delete a subinterface configured for Frame Relay:
`[NO] INTERFACE SERIAL <slot>/<port>:<channel>:<subif>`
	`IPNOS : CONFIG# no interface serial 2/1:0:4`
	`SERIAL (2/1:0:4): Object successfully deleted`
	`  000485fb:  FR[2/1:0:4]: Declaring Circuit (DLCI 44) INACTIVE`

Default Value

None

Command Mode

config and show

Related Commands

controller

system-up-time

Display total time IPSX switch has been up and running.

Syntax	*Example*
SYSTEM-UP-TIME	IPNOS : CONFIG# **system-up-time** SYSTEM-UP-TIME : 6 day(s), 3 hour(s), 34 minute(s), 59 second(s) IPNOS : CONFIG#

Default Value

None

Command Mode

config and show command

Related Commands

None

time

Set system time, including difference from the universal time code (UTC). Table 3-4 describes time fields and available ranges.

Table 3-4 Time Fields

Syntax	*Description*	*Range*
YYYY	Year	1999–2098
MM	Month	1–12
DD	Day	1–31
HH	Hour	0–23
HH	Minutes	0–59
HH	Seconds	0–60 (use 60 for leap second)
ss	Deci-seconds	0–9 (default is 0)
+/-	Direction from UTC	"+" or "-"
HH	Hours from UTC	0–13
MM	Minutes from UTC	0–59

Syntax	*Example*

With timezone:
`TIME <YYYY-MM-DD,HH:MM:SS.ss>[<,+/-HH:MM>]`
	`IPNOS : CONFIG#` **`time 2000-02-08,14:58:30.00,-08.00`**
	`TIME: Object successfully set`
	
	`IPNOS : CONFIG#` **`display time`**
	
	`TIME : 2000-2-8,14:58:39.0,-8:0`

Without timezone:
`TIME <YYYY-MM-DD,HH:MM:SS.ss>`
	`IPNOS : CONFIG#` **`time 2000-02-08,14:53:30.00`**
	`No Timezone info; Re-enter if needed`
	`No Timezone info; Re-enter if needed`
	`TIME: Object successfully set`
	
	`IPNOS : CONFIG#` **`display time`**
	
	`TIME : 2000-2-8,14:55:35.0,+0:0`

Default Value

None

Command Mode

config command

Related Commands

None

verbose-load

Display contents of config.dat to the screen when loading.

Syntax	*Example*
VERBOSE-LOAD	IPNOS: CONFIG# **verbose-load** VERBOSE-LOAD: Object successfully set Load configuration output will be printed out to console IPNOS: CONFIG# **load** Loading configuration from config.dat CONFIG INTERFACE ETHERNET 1/1 DUPLEX HALF SPEED 100 STATUS ENABLE COMMIT ... VPN(1000000): Object successfully set TOP Done Loading configuration from config.dat
NO VERBOSE-LOAD	IPNOS: CONFIG# **no verbose-load** VERBOSE-LOAD: Object successfully set Load configuration output will be disabled IPNOS: CONFIG# **load** Loading configuration from config.dat Done Loading configuration from config.dat

Default Value

Enabled

Command Mode

config command

Related Command

load

vpn

Go to VPN command level. See "VPN Commands" in Chapter 15 for commands available.

Syntax	*Example*
[CREATE | DELETE] VPN <vpnId>	IPNOS : config# **vpn 50** IPNOS : config vpn(50)#
[CREATE | DELETE] VPN <vpnId>	IPNOS : config# **create vpn 50** IPNOS : config vpn(50)#
[CREATE | DELETE] VPN <vpnId>	IPNOS : config# **delete vpn 50** VPN 50: Object successfully deleted IPNOS : config#

Default Value

None

Command Mode

config

Related Commands

vr
vi

CHAPTER

4 Control and Ethernet Blade Commands

Use these commands to configure the control blade and the Ethernet blade. This chapter describes the following commands:

Command	*Page*
ethernet	4-2
autoneg	4-3
display	4-4
duplex	4-5
encapsulation	4-6
speed	4-7
status	4-8
vlanid	4-9
userpriority	4-10
vlanname	4-11

ethernet

Go to the control blade or fast Ethernet blade interface command level, or the fast Ethernet blade subinterface command level. The following commands are available.

- "autoneg" on page 4-3
- "display" on page 4-4
- "duplex" on page 4-5
- "encapsulation" on page 4-6
- "speed" on page 4-7
- "status" on page 4-8
- "vlanid" on page 4-9
- "userpriority" on page 4-10
- "vlanname" on page 4-11

Syntax	*Example*
Control and fast Ethernet blade interface: `INTERFACE ETHERNET {<slot>/<port>}`	`IPNOS : CONFIG#` **`interface ethernet 1/1`** `IPNOS : CONFIG INTERFACE ETHERNET(1/1)#`
Fast Ethernet blade subinterface: `[NO] INTERFACE ETHERNET {<slot>/<port>:<channel>:<subif>}`	`IPNOS : CONFIG#` **`interface ethernet 4/1:0:4`** `IPNOS : CONFIG INTERFACE ETHERNET(4/1:0:4)#`

Fast Ethernet blade subinterface:
`[NO] INTERFACE ETHERNET {<slot>/<port>:<channel>:<subif>}`
	`IPNOS : CONFIG#` **`no interface ethernet 4/1:0:4`**
	`ETHERNET (4/1:0:4): Object successfully deleted`
	`IPNOS : CONFIG#`

Default Value

None

Command Level

config

Related Commands

interface serial, interface atm

autoneg

Enable autonegotiation of the line speed and transmission type with a peer device. When enabled, you cannot manually change the duplex or the speed.

Syntax	*Example*
AUTONEG {ENABLE | DISABLE}	IPNOS: CONFIG INTERFACE ETHERNET(1/1)# **autoneg enable** IPNOS: CONFIG INTERFACE ETHERNET(1/1)# **display autoneg** AUTONEG : ENABLE IPNOS: CONFIG INTERFACE ETHERNET(1/1)# **comm** Commit configuration INTERFACE ETHERNET(1/1): Object successfully set
AUTONEG {ENABLE | DISABLE}	IPNOS: CONFIG INTERFACE ETHERNET(1/1)# **autoneg disable** IPNOS: CONFIG INTERFACE ETHERNET(1/1)# **display autoneg** AUTONEG : DISABLE IPNOS: CONFIG INTERFACE ETHERNET(1/1)# **comm** Commit configuration INTERFACE ETHERNET(1/1): Object successfully set

Default Value

enable

Command Level

ethernet

Related Commands

duplex
speed

display

Show current blade configuration.

Syntax	*Example*
Control blade and fast Ethernet blade: DISPLAY	IPNOS: CONFIG INTERFACE ETHERNET(1/1)# **display** SLOT : 1 PORT : 1 ENCAPSULATION : ETHERNET AUTONEG : ENABLE DUPLEX : HALF SPEED : 10 Mbps LINKTEST : 1 ADMIN-STATUS : ENABLE OPER-STATUS : UP ETHERNET VLAN(s): List is empty
Fast Ethernet blade subinterface: DISPLAY	IPNOS : CONFIG INTERFACE ETHERNET(4/1:0:1)# **display** SLOT : 4 PORT : 1 CHANNEL : 0 SUBIF : 1 VLANID : 0 VLANNAME : USERPRIORITY : 0 ADMIN-STATUS : ENABLE OPER-STATUS : DOWN

Default Value

None

Command Level

ethernet

Related Commands

serial display

duplex

Set data transmission direction. Choose full duplex (both directions simultaneously) or half duplex (one direction at a time). If autonegotiation is enabled, duplex is set automatically.

Syntax	*Example*
DUPLEX {HALF | FULL}	IPNOS: CONFIG INTERFACE ETHERNET(1/1)# **duplex half** IPNOS: CONFIG INTERFACE ETHERNET(1/1)# **display duplex** DUPLEX : HALF IPNOS: CONFIG INTERFACE ETHERNET(1/1)# **commit** Commit configuration INTERFACE ETHERNET(1/1): Object successfully set
DUPLEX {HALF | FULL}	IPNOS: CONFIG INTERFACE ETHERNET(1/1)# **duplex full** IPNOS: CONFIG INTERFACE ETHERNET(1/1)# **display duplex** DUPLEX : FULL IPNOS: CONFIG INTERFACE ETHERNET(1/1)# **commit** Commit configuration INTERFACE ETHERNET(1/1): Object successfully set

Default Value

half

Command Level

ethernet

Related Commands

autoneg
speed

encapsulation

Set interface encapsulation. Currently only Ethernet is supported.

Syntax	*Example*
ENCAPSULATION ETHERNET	IPNOS: CONFIG INTERFACE ETHERNET(1/1)# **encapsulation ethernet** IPNOS: CONFIG INTERFACE ETHERNET(1/1)# **display encapsulation** ENCAPSULATION : ETHERNET IPNOS: CONFIG INTERFACE ETHERNET(1/1)# **commit** Commit configuration INTERFACE ETHERNET(1/1): Object successfully set

Default Value

ethernet

Command Level

ethernet

Related Commands

autoneg
serial encapsulation

speed

Set bandwidth. Choose 10 Mbps or 100 Mbps. If autonegotiation is enabled, the speed is set automatically and this command cannot be used.

Syntax	*Example*

`SPEED {10 | 100}`	`IPNOS: CONFIG INTERFACE ETHERNET(1/1)# speed 10`
	
	`IPNOS: CONFIG INTERFACE ETHERNET(1/1)# display speed`
	`SPEED  : 10 Mbps`
	
	`IPNOS: CONFIG INTERFACE ETHERNET(1/1)# commit`
	`Commit configuration`
	`INTERFACE ETHERNET(1/1): Object successfully set`

`SPEED {10 | 100}`	`IPNOS: CONFIG INTERFACE ETHERNET(1/1)# speed 100`
	
	`IPNOS: CONFIG INTERFACE ETHERNET(1/1)# display speed`
	`SPEED  : 100 Mbps`
	
	`IPNOS: CONFIG INTERFACE ETHERNET(1/1)# commit`
	`Commit configuration`
	`INTERFACE ETHERNET(1/1): Object successfully set`

Default Value

100

Command Level

ethernet

Related Commands

duplex
autoneg

status

Enable administrative status of the interface or subinterface.

Syntax	*Example*
Control blade and fast Ethernet blade interface: STATUS {ENABLE | DISABLE}	IPNOS: CONFIG INTERFACE ETHERNET(1/1)# **status enable** IPNOS: CONFIG INTERFACE ETHERNET(1/1)# **display status** ADMIN-STATUS : ENABLE OPER-STATUS : DOWN IPNOS: CONFIG INTERFACE ETHERNET(1/1)# **commit** Commit configuration INTERFACE ETHERNET(1/1): Object successfully set
Control blade and fast Ethernet blade interface: STATUS {ENABLE | DISABLE}	IPNOS: CONFIG INTERFACE ETHERNET(1/1)# **status disable** IPNOS: CONFIG INTERFACE ETHERNET(1/1)# **display status** ADMIN-STATUS : DISABLE OPER-STATUS : DOWN IPNOS: CONFIG INTERFACE ETHERNET(1/1)# **commit** Commit configuration INTERFACE ETHERNET(1/1): Object successfully set
Fast Ethernet blade subinterface: STATUS {ENABLE | DISABLE}	IPNOS : CONFIG INTERFACE ETHERNET(4/1:0:1)# **status enable** IPNOS : CONFIG INTERFACE ETHERNET(4/1:0:1)# **display status** ADMIN-STATUS : ENABLE OPER-STATUS : UP
Fast Ethernet blade subinterface: STATUS {ENABLE | DISABLE}	IPNOS : CONFIG INTERFACE ETHERNET(4/1:0:1)# **status disable** IPNOS : CONFIG INTERFACE ETHERNET(4/1:0:1)# **display status** ADMIN-STATUS : DISABLE OPER-STATUS : DOWN

Default Value

enable

Command Level

ethernet

Related Commands

duplex
autoneg

vlanid

Set the virtual local area network (VLAN) identifier. Choose from 1 to 4094.

Syntax	*Example*

Fast Ethernet subinterface:
VLANID {1..4094}
	`IPNOS : CONFIG INTERFACE ETHERNET(4/1:0:1)# vlanid 400`
	
	`IPNOS : CONFIG INTERFACE ETHERNET(4/1:0:1)# display`
	`SLOT : 4`
	`PORT : 1`
	`CHANNEL : 0`
	`SUBIF : 1`
	`VLANID : 400`
	`VLANNAME :`
	`USERPRIORITY : 0`
	`ADMIN-STATUS : ENABLE`
	`OPER-STATUS : DOWN`

Default Value

None

Command Level

ethernet

Related Commands

userpriority
vlanname

userpriority

Set the forward priority bit for each VLAN. Choose 0 (lowest) to 7 (highest).

Syntax	*Example*
Fast Ethernet blade subinterface: USERPRIORITY {0..7}	IPNOS : CONFIG INTERFACE ETHERNET(4/1:0:1)# **userpriority 5** IPNOS : CONFIG INTERFACE ETHERNET(4/1:0:1)# **display** SLOT : 4 PORT : 1 CHANNEL : 0 SUBIF : 1 VLANID : 400 VLANNAME : USERPRIORITY : 5 ADMIN-STATUS : ENABLE OPER-STATUS : DOWN
Fast Ethernet blade subinterface: [NO] USERPRIORITY	IPNOS : CONFIG INTERFACE ETHERNET(4/1:0:1)# **no userpriority** IPNOS : CONFIG INTERFACE ETHERNET(4/1:0:1)# **display** SLOT : 4 PORT : 1 CHANNEL : 0 SUBIF : 1 VLANID : 400 VLANNAME : USERPRIORITY : 4 ADMIN-STATUS : ENABLE OPER-STATUS : DOWN

Default Value

0

Command Level

ethernet

Related Commands

vlanid
vlanname

vlanname

Set the name of the VLAN.

Syntax	*Example*
Fast Ethernet blade subinterface: VLANNAME <value>	IPNOS : CONFIG INTERFACE ETHERNET(4/1:0:1)# **vlanname vlan25** IPNOS : CONFIG INTERFACE ETHERNET(4/1:0:1)# **display** SLOT : 4 PORT : 1 CHANNEL : 0 SUBIF : 1 VLANID : 400 VLANNAME : vlan25 USERPRIORITY : 5 ADMIN-STATUS : ENABLE OPER-STATUS : DOWN

Default Value

None

Command Level

ethernet

Related Commands

vlanid
userpriority

CHAPTER

5 Channelized DS3 Blade Commands

Use these commands to configure a channelized DS3 blade. This chapter describes the following commands:

Command	*Page*
cablelength	5-2
channel	5-3
clock	5-5
display	5-6
framing	5-9
loopback	5-10
shutdown	5-12
t1	5-13
serial	5-17
display	5-18
encapsulation	5-20

You have to be in the **controller** level to access the blade interface commands. See “controller” on page 3-35.

cablelength

Set the length of cable directly attached to the TX and RX ports. The maximum distance is 450 feet.

Syntax	*Example*

`CABLELENGTH <length-in-feet>`	`IPNOS : CONFIG CONTROLLER T3(2/1)# cable CABLELENGTH length-in-feet: 200`
	
	`IPNOS : CONFIG CONTROLLER T3(2/1)# dis cable CABLELENGTH : 200`

Default Value

200 feet

Command Level

controller

Related Commands

interface serial cablelength (unchannelized DS3 blade)

channel

Create an HDLC channel. The channelized DS3 blade has two T3 ports. Each T3 port consists of 28 T1 links. Each T1 link has 24 DS0 timeslots. You can create one or more channels that allocate one or more non-contiguous timeslots in a T1. Channels cannot span across multiple T1 links. Create up to 128 channels on one T3 port. Table 5-1 describes the parameters.

Table 5-1 Channel Configuration Parameters

Parameter	*Description*
Channel ID	Set channel identifier. Choose from 1 to 128.
T1 ID	Set T1 identifier. Choose from 1 to 28.
Time slot	Assign one or more timeslots. Choose from 1 to 24 timeslots in the following manner: • Individual. For example, timeslot 5 • Contiguous. For example, timeslot 4–6 • Non-Contiguous. For example, timeslot 1–3, 6–10 • Combination. For example, timeslot 11–15, 20, 22–24
Speed	Set channel speed. Choose 56 kbps or 64 kbps.

Syntax	*Example*

For contiguous channels:
`CHANNEL <chanId> <t1Id> <time-slot> SPEED {56 | 64}`
	`IPNOS : CONFIG CONTROLLER T3(2/1)# channel 1 1 1-6 speed 64`
	`CONTROLLER T3(2/1): Object successfully set`
	`CHANNEL 1: Object successfully set`
	
	`IPNOS : CONFIG CONTROLLER T3(2/1)# display channel`
	
	`CHANNEL(s):`
	`CHANNEL | T1 | TIME-SLOT | SPEED`
	`1 | 1 | 1-6 | 64`

To delete a channel:
`[NO] CHANNEL <chanId>`
	`IPNOS : CONFIG CONTROLLER T3(2/1)# no channel 1`
	`CONTROLLER T3(2/1): Object successfully set`
	`CHANNEL 1: Object successfully deleted`
	
	`IPNOS : CONFIG CONTROLLER T3(2/1)# display channel`
	
	`CHANNEL(s):`
	`List is empty`

Syntax	*Example*
For non-contiguous channels: CHANNEL <chanId> <t1Id> <time-slot> SPEED {56 | 64}	IPNOS : CONFIG CONTROLLER T3(2/1)# **channel 4 3 13-16,22-24 speed 56** CONTROLLER T3(2/1): Object successfully set CHANNEL 4: Object successfully set IPNOS : CONFIG CONTROLLER T3(2/1)# **display channel** CHANNEL(s): CHANNEL | T1 | TIME-SLOT | SPEED 4 | 3 | 13-16, 22-24 | 56

Default Value

None

Command Level

controller

Related Commands

channel (channelized E3 blade)

clock

Set the source of the clock for internal or external (line).

Syntax	*Example*
CLOCK SOURCE {LINE | INTERNAL}	IPNOS : CONFIG CONTROLLER T3(2/1)# **clock source internal** IPNOS : CONFIG CONTROLLER T3(2/1)# **disp clock** CLOCK : INTERNAL
CLOCK SOURCE {LINE | INTERNAL}	IPNOS : CONFIG CONTROLLER T3(2/1)# **clock source line** IPNOS : CONFIG CONTROLLER T3(2/1)# **display clock** CLOCK : LINE

Default Value

internal

Command Level

controller

Related Commands

interface serial clock (unchannelized DS3 blade)
interface serial clock (unchannelized E3 blade)

display

View the current DS3 blade configuration. Table 5-2 describes performance monitoring parameters viewed using the **display tabular** command.

Table 5-2 Performance Monitoring Parameters

Parameter	*Description*
LCV	Line code violations
PCV	Path bit code violations
CCV	C-bit code violations
PES	Path bit errored seconds
PSES	Path bit severely errored seconds
LES	Line errored seconds
CES	C-bit errored seconds
CSES	C-bit severely errored seconds
LSES	Line severely errored seconds

Syntax	*Example*
DISPLAY	IPNOS : CONFIG CONTROLLER T3(2/1)# **display** SLOT : 2 PORT : 1 FRAMING : C-BIT CABLELENGTH : 200 INTERFACE : ENABLE TEST : 0 LOOPBACK : NONE CLOCK : LINE No Alarms detected. T1(s): 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 CHANNEL(s): CHANNEL | T1 | TIME-SLOT | SPEED 4 | 3 | 13-16, 22-24 | 56
DISPLAY TABULAR	Same as DISPLAY example above, plus: ... Current interval time elapsed: 9 min. and 37 sec. INTERVAL LCV PCV CCV PES PSES LES CES CSES LSES current 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 ... ... 80 0 0 0 1 1 0 1 1 0
DISPLAY [<argument>]	IPNOS : CONFIG CONTROLLER T3(2/1)# **display t1** T1(s): 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28

Syntax	*Example*
DISPLAY T1 <t1id>	IPNOS : CONFIG CONTROLLER T3(2/1)# **display t1 1** T1 : 1 FRAMING : ESF LINECODE : B8ZS CLOCK : INTERNAL YELLOW DETECTION: ENABLE YELLOW GENERATION: ENABLE FDL ANSI : ENABLE

Default Value

None

Command Level

controller

Related Commands

controller display (channelized E3 blade)
interface serial display (unchannelized DS3 blade)
interface serial display (unchannelized E3 blade)

framing

Set the DS3 framing options. Table 5-3 describes the parameters available.

Table 5-3 DS3 Framing Options

Option	*Description*
C-bit	Handle status information, connection alarms, and data link information.
M23	Tie all seven DS2 signals from the data stream into a full DS3.
Auto-Detect	Automatically detect framing options.

Syntax	*Example*
FRAMING {C-BIT | M23 | AUTO-DETECT}	IPNOS : CONFIG CONTROLLER T3(2/1)# **framing c-bit** IPNOS : CONFIG CONTROLLER T3(2/1)# **display framing** FRAMING : C-BIT
FRAMING {C-BIT | M23 | AUTO-DETECT}	IPNOS : CONFIG CONTROLLER T3(2/1)# **framing m23** IPNOS : CONFIG CONTROLLER T3(2/1)# **display framing** FRAMING : M23
FRAMING {C-BIT | M23 | AUTO-DETECT}	IPNOS : CONFIG CONTROLLER T3(2/1)# **framing auto-detect** IPNOS : CONFIG CONTROLLER T3(2/1)# **display framing** FRAMING :AUTO-DETECT

Default Value

C-bit

Command Level

controller

Related Commands

interface serial framing (unchannelized DS3 blade)
loopback

loopback

Set the local or remote port into a loopback state. Table 5-4 describes parameters available.

Table 5-4 Loopback Parameters

Parameter	*Description*
Local	Loop the router output data back toward the router at the T1 framer and send an alarm indicator signal (AIS) out toward the network.
Dual	Loop the router output data back toward the router at the T1 framer and send the payload data back toward the network at the T1 framer.
Network line	Loop the data back toward the network before the T1 framer. It sends a request to the far end for a network loopback, essentially performing a physical line test.
Network payload	Loop the payload data back toward the network at the T1 framer. This command divides the loopback signals into separate DS1/T1 payloads for all 28 T1s channels in the DS3 port.
Remote line	Send a repeating data link code word to the remote end, requesting that it enter into a network line loopback. This option can only be used when C-bit framing is used in the DS3.
Remote payload	Send a repeating data link code word to the remote end, requesting that it enter into a network payload loopback. This option can only be used when C-bit framing is used in the DS3.
No	Disable loopback.

Syntax	*Example*
LOOPBACK {LOCAL | DUAL | NETWORK {LINE | PAYLOAD} | REMOTE {LINE | PAYLOAD} }	IPNOS : CONFIG CONTROLLER T3(2/1)# **loopback local** IPNOS : CONFIG CONTROLLER T3(2/1)# **dis loopback** LOOPBACK : LOCAL(NEAR-END)
LOOPBACK {LOCAL | DUAL | NETWORK {LINE | PAYLOAD} | REMOTE {LINE | PAYLOAD} }	IPNOS : CONFIG CONTROLLER T3(2/1)# **loopback dual** IPNOS : CONFIG CONTROLLER T3(2/1)# **dis loopback** LOOPBACK : LOCAL(NEAR-END) LINE(NEAR-END)

Syntax	*Example*
LOOPBACK {LOCAL | DUAL | NETWORK {LINE | PAYLOAD} | REMOTE {LINE | PAYLOAD} }	IPNOS : CONFIG CONTROLLER T3(2/1)# **loopback network line** IPNOS : CONFIG CONTROLLER T3(2/1)# dis loopback LOOPBACK : LINE(NEAR-END)
LOOPBACK {LOCAL | DUAL | NETWORK {LINE | PAYLOAD} | REMOTE {LINE | PAYLOAD} }	IPNOS : CONFIG CONTROLLER T3(2/1)# **loopback network payload** IPNOS : CONFIG CONTROLLER T3(2/1)# **dis loopback** LOOPBACK : PAYLOAD(NEAR-END)
LOOPBACK {LOCAL | DUAL | NETWORK {LINE | PAYLOAD} | REMOTE {LINE | PAYLOAD} }	IPNOS : CONFIG CONTROLLER T3(2/1)# **loopback remote line** IPNOS : CONFIG CONTROLLER T3(2/1)# **dis loopback** LOOPBACK : PAYLOAD(NEAR-END) LINE(FAR-END)
LOOPBACK {LOCAL | DUAL | NETWORK {LINE | PAYLOAD} | REMOTE {LINE | PAYLOAD} }	IPNOS : CONFIG CONTROLLER T3(2/1)# **loopback remote payload** IPNOS : CONFIG CONTROLLER T3(2/1)# **dis loopback** LOOPBACK : PAYLOAD(NEAR-END) PAYLOAD(FAR-END)
NO LOOPBACK [REMOTE]	IPNOS : CONFIG CONTROLLER T3(2/1)# **no loopback remote** IPNOS : CONFIG CONTROLLER T3(2/1)# **dis loopback** LOOPBACK : PAYLOAD(NEAR-END)
NO LOOPBACK [REMOTE]	IPNOS : CONFIG CONTROLLER T3(2/1)# **no loopback** IPNOS : CONFIG CONTROLLER T3(2/1)# **dis loopback** LOOPBACK : NONE

Default Value

No loopback and no loopback remote.

Command Level

controller

Related Commands

interface serial loopback (channelized E3 blade)
interface serial loopback (unchannelized DS3 blade)
interface serial loopback (unchannelized E3 blade)

shutdown

Shut down a DS3 port.

Syntax	*Example*
SHUTDOWN	IPNOS : CONFIG CONTROLLER T3(2/1)# **shutdown** Change will take effect only after commit IPNOS : CONFIG CONTROLLER T3(2/1)# **display interface** INTERFACE : DISABLE
[NO] SHUTDOWN	IPNOS : CONFIG CONTROLLER T3(2/1)# **no shutdown** Change will take effect only after commit IPNOS : CONFIG CONTROLLER T3(2/1)# **display interface** INTERFACE : ENABLE

Default Value

None

Command Level

controller

Related Commands

commit
interface serial shutdown (unchannelized DS3 blade)
controller shutdown (channelized E3 blade)
interface serial shutdown (unchannelized E3 blade)

t1

Configure a specified T1 interface. Up to 28 T1 interfaces can be configured on a channelized DS3 blade. Table 5-5 shows the parameters available.

Table 5-5 T1 Parameters

Parameters	*Description*
Clock	Set clock source set to internal or external (line).
Framing	Set T1 framing format. The following formats are available: • *Superframe*—Combine twelve DS1 frames into a superframe for signal control. • *Extended Superframe*—Combine 24 DS1 frames for signal control.
Yellow Alarm	Detect or generate yellow alarm messages.

Syntax	*Example*
T1 <t1Id> CLOCK SOURCE {LINE | INTERNAL}	IPNOS : CONFIG CONTROLLER T3(2/1)# **t1 1 clock source line** CONTROLLER T3(2/1): Object successfully set T1 1: Object successfully set IPNOS : CONFIG CONTROLLER T3(2/1)# **display t1 1** T1 : 1 FRAMING : ESF LINECODE : B8ZS CLOCK : LINE YELLOW DETECTION: ENABLE YELLOW GENERATION: ENABLE FDL ANSI : ENABLE
T1 <t1Id> CLOCK SOURCE {LINE | INTERNAL}	IPNOS : CONFIG CONTROLLER T3(2/1)# **t1 1 clock source internal** CONTROLLER T3(2/1): Object successfully set T1 1: Object successfully set IPNOS : CONFIG CONTROLLER T3(2/1)# **display t1 1** T1 : 1 FRAMING : ESF LINECODE : B8ZS CLOCK : INTERNAL YELLOW DETECTION: ENABLE YELLOW GENERATION: ENABLE FDL ANSI : ENABLE

Syntax	*Example*
T1 <t1Id> FRAMING {SF | ESF}	IPNOS : CONFIG CONTROLLER T3(2/1)# **t1 1 framing sf** CONTROLLER T3(2/1): Object successfully set T1 1: Object successfully set IPNOS : CONFIG CONTROLLER T3(2/1)# **display t1 1** T1 : 1 FRAMING : SF LINECODE : B8ZS CLOCK : INTERNAL YELLOW DETECTION: ENABLE YELLOW GENERATION: ENABLE FDL ANSI : ENABLE
T1 <t1Id> FRAMING {SF | ESF}	IPNOS : CONFIG CONTROLLER T3(2/1)# **t1 1 framing esf** CONTROLLER T3(2/1): Object successfully set T1 1: Object successfully set IPNOS : CONFIG CONTROLLER T3(2/1)# **display t1 1** T1 : 1 FRAMING : ESF LINECODE : B8ZS CLOCK : INTERNAL YELLOW DETECTION: ENABLE YELLOW GENERATION: ENABLE FDL ANSI : ENABLE
[NO] T1 <t1Id> YELLOW {DETECTION | GENERATION}	IPNOS : CONFIG CONTROLLER T3(2/1)# **t1 1 yellow detection** CONTROLLER T3(2/1): Object successfully set T1 1: Object successfully set IPNOS : CONFIG CONTROLLER T3(2/1)# **display t1 1** T1 : 1 FRAMING : ESF LINECODE : B8ZS CLOCK : INTERNAL YELLOW DETECTION: ENABLE YELLOW GENERATION: ENABLE FDL ANSI : ENABLE

Syntax	Example
[NO] T1 <t1Id> YELLOW {DETECTION | GENERATION}	IPNOS : CONFIG CONTROLLER T3(2/1)# **no t1 1 yellow detection** CONTROLLER T3(2/1): Object successfully set T1 1: Object successfully set IPNOS : CONFIG CONTROLLER T3(2/1)# **display t1 1** T1 : 1 FRAMING : ESF LINECODE : B8ZS CLOCK : INTERNAL YELLOW DETECTION: DISABLE YELLOW GENERATION: ENABLE FDL ANSI : ENABLE
[NO] T1 <t1Id> YELLOW {DETECTION | GENERATION}	IPNOS : CONFIG CONTROLLER T3(2/1)# **t1 1 yellow generation** CONTROLLER T3(2/1): Object successfully set T1 1: Object successfully set IPNOS : CONFIG CONTROLLER T3(2/1)# **display t1 1** T1 : 1 FRAMING : ESF LINECODE : B8ZS CLOCK : INTERNAL YELLOW DETECTION: DISABLE YELLOW GENERATION: ENABLE FDL ANSI : ENABLE
[NO] T1 <t1Id> YELLOW {DETECTION | GENERATION}	IPNOS : CONFIG CONTROLLER T3(2/1)# **no t1 1 yellow generation** CONTROLLER T3(2/1): Object successfully set T1 1: Object successfully set IPNOS : CONFIG CONTROLLER T3(2/1)# **display t1 1** T1 : 1 FRAMING : ESF LINECODE : B8ZS CLOCK : INTERNAL YELLOW DETECTION: DISABLE YELLOW GENERATION: DISABLE FDL ANSI : ENABLE

Default Value

See Table 5-5

Command Level

controller

Related Commands

controller e1 (channelized E3 blade)
interface serial clock (unchannelized DS3 blade)
interface serial framing (unchannelized DS3 blade)
interface serial clock (unchannelized E3 blade)

serial

Go to the DS3 channel interface level. The following commands are available.

- "display" on page 5-18
- "encapsulation" on page 5-20

Syntax	*Example*

`INTERFACE SERIAL <slot>/<port>:<channel>`	`IPNOS : CONFIG#` **`interface serial 2/1:4`**
	
	`IPNOS : CONFIG INTERFACE SERIAL(2/1:4)#` **`dis`**
	`SLOT : 2`
	`PORT : 1`
	`CHANNEL : 4`
	`ENCAPSULATION : NONE`

Default Value

None

Command Level

Config

Related Commands

channel
interface ethernet (Control blade)
interface serial (channelized E3 blade)
interface serial (unchannelized DS3 blade)
interface serial (unchannelized E3 blade)

display

View current blade encapsulation.

Syntax	*Example*

With no encapsulation:
`DISPLAY`
	`IPNOS : CONFIG INTERFACE SERIAL(2/1:4)# display`
	`SLOT : 2`
	`PORT : 1`
	`CHANNEL : 4`
	`ENCAPSULATION : NONE`

With PPP encapsulation:
`DISPLAY`
	`IPNOS : CONFIG INTERFACE SERIAL(2/1:4)# display`
	`SLOT : 2`
	`PORT : 1`
	`CHANNEL : 4`
	`ENCAPSULATION : PPP`
	`AUTHENTICATION : NONE`
	`CONFIG-REQ-RETRANSMIT: 10`
	`TERM-REQ-RETRANSMIT: 2`
	`TIMEOUT : 5`
	`REAUTH : 0`
	`ADMIN-STATUS : ENABLE`
	`OPER-STATUS : DOWN`

Syntax	Example
With Frame Relay encapsulation: DISPLAY	IPNOS : CONFIG INTERFACE SERIAL(2/1:4)# **display** SLOT : 2 PORT : 1 CHANNEL : 4 ENCAPSULATION : FRAME-RELAY LMI-TYPE : ANSI LMI-T391DTE : 10 LMI-N391DTE : 5 LMI-N392DTE : 3 LMI-N393DTE : 4 INTF-TYPE : DTE LMI-N392DCE : 3 LMI-N393DCE : 4 LMI-T392DCE : 15 ADDRESS : Q922 LENGTH : TWO-OCTETS MAX-SUPPORTED-VC: 317 MULTICAST : NON-BROADCAST STATUS : RUNNING FRAME-RELAY PVC(s): List is empty
With HDLC encapsulation: DISPLAY	IPNOS : CONFIG INTERFACE SERIAL(2/1:4)# **display** SLOT : 2 PORT : 1 CHANNEL : 4 ENCAPSULATION : HDLC KEEPALIVE : 10 ADMIN-STATUS : ENABLE OPER-STATUS : DOWN

Default Value

None

Command Level

serial

Related Commands

interface serial display (channelized E3 blade)
interface serial display (unchannelized DS3 blade)
interface serial display (unchannelized E3 blade)
show

encapsulation

Encapsulate the channel and go to the PPP, Frame Relay, or HDLC command level. See the following chapters for commands.

- Chapter 12, “Frame Relay Commands”
- Chapter 13, “HDLC Commands”
- Chapter 14, “PPP Commands”

Syntax	*Example*
ENCAPSULATION {PPP | FRAME-RELAY | HDLC}	IPNOS : CONFIG INTERFACE SERIAL(2/1:4)# **encapsulation ppp** INTERFACE SERIAL(2/1:4): Object successfully set IPNOS : CONFIG INTERFACE SERIAL(2/1:4) PPP# **display** SLOT : 2 PORT : 1 CHANNEL : 4 AUTHENTICATION : NONE CONFIG-REQ-RETRANSMIT: 10 TERM-REQ-RETRANSMIT: 2 TIMEOUT : 5 REAUTH : 0 ADMIN-STATUS : ENABLE OPER-STATUS : DOWN IPNOS : CONFIG INTERFACE SERIAL(2/1:4) PPP#

Syntax	*Example*
ENCAPSULATION {PPP | FRAME-RELAY | HDLC}	IPNOS : CONFIG INTERFACE SERIAL(2/1:4)# **encapsulation frame-relay** INTERFACE SERIAL(2/1:4): Object successfully set IPNOS : CONFIG INTERFACE SERIAL(2/1:4) FRAME-RELAY# **display** SLOT : 2 PORT : 1 CHANNEL : 4 LMI-TYPE : ANSI LMI-T391DTE : 10 LMI-N391DTE : 5 LMI-N392DTE : 3 LMI-N393DTE : 4 INTF-TYPE : DTE LMI-N392DCE : 3 LMI-N393DCE : 4 LMI-T392DCE : 15 ADDRESS : Q922 LENGTH : TWO-OCTETS MAX-SUPPORTED-VC: 317 MULTICAST : NON-BROADCAST STATUS : RUNNING IPNOS : CONFIG INTERFACE SERIAL(2/1:4) FRAME-RELAY#
ENCAPSULATION {PPP | FRAME-RELAY | HDLC}	IPNOS : CONFIG INTERFACE SERIAL(2/1:4)# **encapsulation hdlc** INTERFACE SERIAL(2/1:4): Object successfully set IPNOS : CONFIG INTERFACE SERIAL(2/1:4) HDLC# **display** SLOT : 2 PORT : 1 CHANNEL : 4 KEEPALIVE : 10 ADMIN-STATUS : ENABLE OPER-STATUS : DOWN IPNOS : CONFIG INTERFACE SERIAL(2/1:4) HDLC#
NO ENCAPSULATION	IPNOS : CONFIG INTERFACE SERIAL(2/1:4)# **no encapsulation** IPNOS : CONFIG INTERFACE SERIAL(2/1:4)# **display** SLOT : 2 PORT : 1 CHANNEL : 4 ENCAPSULATION : NONE IPNOS : CONFIG INTERFACE SERIAL(2/1:4)#

Default Value

None

Command Level

serial

Related Commands

interface ethernet encapsulation (control blade)
interface serial encapsulation (channelized E3 blade)
interface serial encapsulation (unchannelized DS3 blade)
interface serial encapsulation (unchannelized E3 blade)

CHAPTER

6 Unchannelized DS3 Blade Commands

Use these commands to configure an unchannelized DS3 blade. This chapter describes the following commands:

Command	*Page*
serial	6-2
cablelength	6-2
clock	6-3
display	6-4
encapsulation	6-5
framing	6-6
loopback	6-7
shutdown	6-8

serial

Go to the interface command level. The following commands are available.

- "cablelength" on page 6-2
- "clock" on page 6-3
- "display" on page 6-4
- "encapsulation" on page 6-5
- "framing" on page 6-6
- "loopback" on page 6-7
- "shutdown" on page 6-8

Syntax	*Example*
`interface serial <slot/port>`	`IPNOS : CONFIG#` **`interface serial 2/1`** `IPNOS : CONFIG INTERFACE SERIAL(2/1)#`

Default Value

None

Command Level

config

Related Commands

interface ethernet (control blade)
interface serial (channelized E3 blade)
interface serial (channelized DS3 blade)
interface serial (unchannelized E3 blade)

cablelength

Set the length of cable attached to the TX and RX ports. The maximum distance is 450 feet.

Syntax	*Example*
`CABLELENGTH <length-in-feet>`	`IPNOS : CONFIG INTERFACE SERIAL(2/1)#` **`cablelength 200`**

Default Value

200 feet

Command Level

serial

Related Commands

interface cablelength (channelized DS3 blade)

clock

Set the source of the clock for internal or external (line).

Syntax	*Example*

`CLOCK SOURCE {INTERNAL | LINE}`	`IPNOS : CONFIG INTERFACE SERIAL(2/1)# clock source internal`
	`IPNOS : CONFIG INTERFACE SERIAL(2/1)# display clock`
	`CLOCK : INTERNAL`

`CLOCK SOURCE {INTERNAL | LINE}`	`IPNOS : CONFIG INTERFACE SERIAL(2/1)# clock source line`
	`IPNOS : CONFIG INTERFACE SERIAL(2/1)# display clock`
	`CLOCK : LINE`

Default Value

internal

Command Level

serial

Related Commands

interface clock (channelized DS3 blade)
interface serial clock (unchannelized E3 blade)

display

View the current DS3 blade configuration. Table 6-1 describes performance monitoring parameters viewed using the **display tabular** command.

Table 6-1 Performance Monitoring Parameters

Parameter	*Description*
LCV	Line code violations
PCV	Path bit code violations
CCV	C-bit code violations
PES	Path bit errored seconds
PSES	Path bit severely errored seconds
LES	Line errored seconds
CES	C-bit errored seconds
CSES	C-bit severely errored seconds
LSES	Line severely errored seconds

Syntax	*Example*
DISPLAY	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **display** SLOT : 2 PORT : 1 ENCAPSULATION : NONE FRAMING : C-BIT CABLELENGTH : 200 INTERFACE : ENABLE CLOCK : INTERNAL No Alarms detected.
DISPLAY TABULAR	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **display tabular** Current interval time elapsed: 9 min. and 37 sec. INTERVAL LCV PCV CCV PES PSES LES CES CSES LSES current 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 ... ... 80 0 0 0 1 1 0 1 1 0

Syntax	Example
DISPLAY <argument>	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **display framing** FRAMING : C-BIT

Default Value

None

Command Level

serial

Related Commands

interface serial display (channelized E3 blade)
controller display (channelized DS3 blade)
interface serial display (unchannelized E3 blade)
show

encapsulation

Encapsulate the channel and go to the PPP, Frame Relay, or HDLC command level. Each port is considered one channel. See the following chapters for commands.

- Chapter 12, “Frame Relay Commands”
- Chapter 13, “HDLC Commands”
- Chapter 14, “PPP Commands”

Syntax	Example
[NO] ENCAPSULATION {FRAME-RELAY | HDLC | PPP}	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **encapsulation frame-relay** IPNOS : CONFIG INTERFACE SERIAL(2/1) FRAME-RELAY#
[NO] ENCAPSULATION {FRAME-RELAY | HDLC | PPP}	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **encapsulation HDLC** IPNOS : CONFIG INTERFACE SERIAL(2/1) HDLC#
[NO] ENCAPSULATION {FRAME-RELAY | HDLC | PPP}	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **encapsulation PPP** IPNOS : CONFIG INTERFACE SERIAL(2/1) PPP#

Default Value

None

Command Level

serial

Related Commands

interface ethernet encapsulation (control blade)
interface serial encapsulation (channelized E3 blade)
interface serial encapsulation (channelized DS3 blade)
interface serial encapsulation (unchannelized E3 blade)

framing

Set the DS3 framing options. Table 6-2 describes the parameters available.

Table 6-2 DS3 Framing Options

Option	*Description*
C-bit	Handle status information, connection alarms, and data link information.
M23	Tie all seven DS2 signals from the data stream into a full DS3.
Auto-Detect	Automatically detect framing options.

Syntax	*Example*
FRAMING {C-BIT | M23 | AUTO-DETECT}	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **framing c-bit** IPNOS : CONFIG INTERFACE SERIAL(2/1)# **display framing** FRAMING : C-BIT
FRAMING {C-BIT | M23 | AUTO-DETECT}	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **framing m23** IPNOS : CONFIG INTERFACE SERIAL(2/1)# **display framing** FRAMING : M23
FRAMING {C-BIT | M23 | AUTO-DETECT}	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **framing auto-detect** IPNOS : CONFIG INTERFACE SERIAL(2/1)# **display framing** FRAMING : AUTO-DETECT

Default Value

c-bit

Command Level

serial

Related Commands

controller framing (channelized DS3 blade)

loopback

Set the local or remote port into a loopback state. Table 6-3 describes parameters available.

Table 6-3 Loopback Parameters

Parameter	*Description*
Local	Loop the router output data back toward the router at the T1 framer and send an alarm indicator signal (AIS) signal out toward the network.
Dual	Loop the router output data back toward the router at the T1 framer and send the payload data back toward the network at the T1 framer.
Network line	Loop the data back toward the network before the T1 framer. It sends a request to the far end for a network loopback, essentially performing a physical line test. The **line** option can only be used when C-bit framing is used in the DS3.
Remote line	Send a repeating data link code word to the remote end, requesting that it enter into a network line loopback.
No	Disable loopback.

Syntax	*Example*
`LOOPBACK {LOCAL | DUAL | NETWORK LINE | REMOTE LINE }`	`IPNOS : CONFIG INTERFACE SERIAL(2/1)#` **`loopback local`**
`LOOPBACK {LOCAL | DUAL | NETWORK LINE | REMOTE LINE }`	`IPNOS : CONFIG INTERFACE SERIAL(2/1)#` **`loopback dual`**
`LOOPBACK {LOCAL | DUAL | NETWORK LINE | REMOTE LINE }`	`IPNOS : CONFIG INTERFACE SERIAL(2/1)#` **`loopback network line`**

Syntax	*Example*
LOOPBACK {LOCAL | DUAL | NETWORK LINE | REMOTE LINE }	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **loopback remote line**
NO LOOPBACK	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **no loopback**

Default Value

No loopback and no loopback remote.

Command Level

controller

Related Commands

interface serial loopback (channelized E3 blade)
interface serial loopback (channelized DS3 blade)
interface serial loopback (unchannelized E3 blade)

shutdown

Shut down a DS3 port.

Syntax	*Example*
[NO] SHUTDOWN	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **shutdown** Change will take effect only after commit IPNOS : CONFIG INTERFACE SERIAL(2/1)# **display interface** INTERFACE : DISABLE
[NO] SHUTDOWN	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **no shutdown** Change will take effect only after commit IPNOS : CONFIG INTERFACE SERIAL(2/1)# **display interface** INTERFACE : ENABLE

Default Value

None

Command Level

serial

Related Commands

commit
controller shutdown (channelized DS3 blade)
controller shutdown (channelized E3 blade)
interface serial shutdown (unchannelized E3 blade)

CHAPTER

7 Channelized E3 Blade Commands

Use these commands to configure the channelized E3 blade. This chapter describes the following commands:

Command	*Page*
channel	7-2
display	7-4
e1	7-6
linecode	7-10
loopback	7-11
national bit	7-13
shutdown	7-14
serial	7-15
display	7-16
encapsulation	7-18

You have to be in the **controller** level to access the blade interface commands. See "controller" on page 3-35.

channel

Create an HDLC channel. The channelized E3 blade has three E3 ports. Each E3 port consists of 16 E1 links. Each E1 link has 31 timeslots. You can create one or more channels that allocate one or more non-contiguous timeslots in an E1. Channels cannot span multiple E1 links. Create up to 85 channels on ports 1 and 2, and 86 channels on port 3. Table 7-1 describes the parameters.

Table 7-1 Channel Configuration Settings

Setting	*Description*
Channel ID	Set channel identifier. Choose from the following. • 1–85 on port 1 • 1–85 on port 2 • 1–43 on port 3 (E1 links 1–8) • 44–86 on port 3 (E1 links 9–16)
E1 ID	Set E1 identifier. Choose from 1 to 16.
Time-slot	Assign one or more timeslots. Choose from 1 to 31 timeslots: • Individual. For example, timeslot 5 • Contiguous. For example, timeslot 4–6 • Non-Contiguous. For example, timeslot 1–3, 6–10 • Combination. For example, timeslot 11–15, 20, 24–31
Speed	Set channel speed. Choose 56 kbps or 64 kbps.

Syntax	*Example*
For contiguous channels: CHANNEL <channelid> <e1id> <time_slot> SPEED {56 | 64}	IPNOS : CONFIG CONTROLLER E3(4/1)# **channel 1 1 1-6 speed 64** CONTROLLER E3(4/1): Object successfully set CHANNEL 1: Object successfully set IPNOS : CONFIG CONTROLLER E3(4/1)# **display channel** CHANNEL(s): CHANNEL | E1 | TIME-SLOT | SPEED 1 | 1 | 1-6 | 64
For non-contiguous timeslots: CHANNEL <channelid> <e1id> <time_slot> SPEED {56 | 64}	IPNOS : CONFIG CONTROLLER E3(4/1)# **channel 4 3 13-16,22-31 speed 56** CONTROLLER E3(4/1): Object successfully set CHANNEL 4: Object successfully set IPNOS : CONFIG CONTROLLER E3(4/1)# **display channel** CHANNEL(s): CHANNEL | E1 | TIME-SLOT | SPEED 4 | 3 | 13-16, 22-31 | 56

Syntax	*Example*

To delete a channel:
`[NO] CHANNEL <channelid>`
	`IPNOS : CONFIG CONTROLLER E3(4/1)# no channel 1`
	`CONTROLLER E3(4/1): Object successfully set`
	`CHANNEL 1: Object successfully deleted`
	
	`IPNOS : CONFIG CONTROLLER E3(4/1)# display channel`
	
	`CHANNEL(s):`
	`List is empty`

Default Value

None

Command Level

controller

Related Commands

channel (channelized DS3 blade)

display

View the current E3 blade configuration. Table 7-2 displays the performance monitoring parameters that come up using the **display tabular** command.

Table 7-2 Performance Monitoring Parameters

Parameter	*Description*
LCV	Line code violations
LES	Line errored seconds
LSES	Line severely errored seconds

Syntax	*Example*
DISPLAY	IPNOS : CONFIG CONTROLLER E3(4/1)# **display** SLOT : 2 PORT : 1 CLOCK : INTERNAL FRAMING : G751 INTERFACE : ENABLE LOOPBACK : NONE LINECODE : HDB3 NATIONAL BIT : 0 Receiver has loss of signal. E1(s): 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 CHANNEL(s): CHANNEL | E1 | TIME-SLOT | SPEED 3 | 3 | 1-31 | 64
DISPLAY <argument>	IPNOS : CONFIG CONTROLLER E3(4/1)# **display channel** CHANNEL(s): CHANNEL | E1 | TIME-SLOT | SPEED 3 | 3 | 1-31 | 64

Syntax	*Example*
DISPLAY E1 <e1id>	IPNOS : CONFIG CONTROLLER E3(4/1)# **display e1 1** E1 : 1 FRAMING : CRC-MULTIFRAME-FEBE CLOCK : INTERNAL YELLOW DETECTION : ENABLE LINE-CONFIG : NO ALARMS LINE-STATUS : NO ALARMS NATIONAL BITS : 15 INTERNATIONAL BITS : 3
DISPLAY TABULAR	IPNOS : CONFIG CONTROLLER E3(4/1)# **display tabular** Current interval time elapsed: 5 min. and 58 sec. INTERVAL LCV LES LSES current 0 358 358 1 0 900 900 2 0 900 900 3 0 900 900 ... 87 0 900 900 88 0 900 900 89 32 900 900

Default Value

None

Command Level

controller

Related Commands

None

e1

Configure a specified E1 interface. Up to 16 E1 interfaces can be configured on a channelized E3 blade. Table 7-3 shows the parameters available.

Table 7-3 E1 Parameters

Parameter	*Description*
Framing	The E1 framing format. Refer to ITU Recommendations G.704, G.706. The following formats are available: • CRC4—An 8-bit Cyclic Redundancy Check-4 (CRC-4) multiframe with the first bit (the Si bit) set to 1. This is used in case digital paths cross national borders and encounter switches not using CRC-4. A multiframe consists of 16 E1 frames. • CRC4-Febe—CRC-4 multiframe with the Si bit set for Far End Block Error (FEBE) checking. This is used to check CRC-4 message block errors. • No-CRC4—CRC-4 multiframe alignment signal allocates all seven 680-bit M subframes within the 4760-bit multiframe to identify overhead bit positions. This is used to protect against false frame alignment. The Si bit is set to 1. • Unframed—No framing. When unframed, all E1 channels are available, but there is no protection against errors and frame misalignment.
Yellow Alarm	Detect yellow alarm messages. Default value is disabled.
National bits	Configures the E1 national reserve bit pattern. Choose from 0 to 31. The default value is 31.
International bits	Configures the E1 international reserve bit pattern. Choose from 0 to 3. The default value is 3.

Syntax	Example
E1 e1Id FRAMING {CRC4 | CRC4-FEBE | NO-CRC4 | UNFRAMED}	IPNOS: CONFIG CONTROLLER E3(3/1)# **e1 1 framing crc4** CONTROLLER E3(3/1): Object successfully set E1 1: Object successfully set IPNOS: CONFIG CONTROLLER E3(3/1)# **display e1 1** E1 : 1 FRAMING : CRC4 CLOCK : INTERNAL YELLOW DETECTION : ENABLE LINE-STATUS : AIS LOF NATIONAL BITS : 31 INTERNATIONAL BITS : 3
E1 e1Id FRAMING {CRC4 | CRC4-FEBE | NO-CRC4 | UNFRAMED}	IPNOS: CONFIG CONTROLLER E3(3/1)# **e1 1 framing crc4-febe** CONTROLLER E3(3/1): Object successfully set E1 1: Object successfully set IPNOS: CONFIG CONTROLLER E3(3/1)# **display e1 1** E1 : 1 FRAMING : CRC4-FEBE CLOCK : INTERNAL YELLOW DETECTION : ENABLE LINE-STATUS : AIS LOF NATIONAL BITS : 31 INTERNATIONAL BITS : 3
E1 e1Id FRAMING {CRC4 | CRC4-FEBE | NO-CRC4 | UNFRAMED}	IPNOS: CONFIG CONTROLLER E3(3/1)# **e1 1 framing no-crc4** CONTROLLER E3(3/1): Object successfully set E1 1: Object successfully set IPNOS: CONFIG CONTROLLER E3(3/1)# **display e1 1** E1 : 1 FRAMING : NO-CRC4 CLOCK : INTERNAL YELLOW DETECTION : ENABLE LINE-STATUS : AIS LOF NATIONAL BITS : 31 INTERNATIONAL BITS : 3

Syntax	*Example*
E1 e1Id FRAMING {CRC4 | CRC4-FEBE | NO-CRC4 | UNFRAMED}	IPNOS: CONFIG CONTROLLER E3(3/1)# **e1 1 framing unframed** CONTROLLER E3(3/1): Object successfully set E1 1: Object successfully set IPNOS: CONFIG CONTROLLER E3(3/1)# **display e1 1** E1 : 1 FRAMING : UNFRAMED CLOCK : INTERNAL YELLOW DETECTION : ENABLE LINE-STATUS : AIS LOF NATIONAL BITS : 31 INTERNATIONAL BITS : 3
[NO] E1 <e1Id> YELLOW DETECTION	IPNOS: CONFIG CONTROLLER E3(3/1)# **e1 2 yellow detect** CONTROLLER E3(3/1): Object successfully set E1 2: Object successfully set IPNOS: CONFIG CONTROLLER E3(3/1)# **display e1 2** E1 : 2 FRAMING : CRC4-FEBE CLOCK : INTERNAL YELLOW DETECTION : ENABLE LINE-STATUS : AIS LOF NATIONAL BITS : 31 INTERNATIONAL BITS : 3
[NO] E1 <e1Id> YELLOW DETECTION	IPNOS: CONFIG CONTROLLER E3(3/1)# **no e1 2 yellow detect** CONTROLLER E3(3/1): Object successfully set E1 2: Object successfully set IPNOS: CONFIG CONTROLLER E3(3/1)# **display e1 2** E1 : 2 FRAMING : CRC4-FEBE CLOCK : INTERNAL YELLOW DETECTION : DISABLE LINE-STATUS : AIS LOF NATIONAL BITS : 31 INTERNATIONAL BITS : 3

Syntax	Example
E1 <e1Id> NATIONAL BITS {1-31}	IPNOS: CONFIG CONTROLLER E3(3/1)# **e1 3 national bits 12** CONTROLLER E3(3/1): Object successfully set E1 3: Object successfully set IPNOS: CONFIG CONTROLLER E3(3/1)# **display e1 3** E1 : 3 FRAMING : CRC4-FEBE CLOCK : INTERNAL YELLOW DETECTION : ENABLE LINE-STATUS : AIS LOF NATIONAL BITS : 12 INTERNATIONAL BITS : 3
E1 <e1Id> INTERNATIONAL BITS {0-3}	IPNOS: CONFIG CONTROLLER E3(3/1)# **e1 3 international bits 2** CONTROLLER E3(3/1): Object successfully set E1 3: Object successfully set IPNOS: CONFIG CONTROLLER E3(3/1)# **display e1 3** E1 : 3 FRAMING : CRC4-FEBE CLOCK : INTERNAL YELLOW DETECTION : ENABLE LINE-STATUS : AIS LOF NATIONAL BITS : 12 INTERNATIONAL BITS : 2

Default Value

See Table 7-3

Command Level

controller

Related Commands

controller clock (channelized DS3 blade)
controller framing (channelized DS3 blade)
controller t1 (channelized DS3 blade)
interface serial clock (unchannelized DS3 blade)
interface serial framing (unchannelized DS3 blade)
interface serial clock (unchannelized E3 blade)
national bit (E3 national bit command)

linecode

Set the way data bits are represented on the physical communication line. Table 7-4 describes the parameters.

Table 7-4 Linecode Parameter

Parameter	*Description*
HDB3	High Density Bipolar 3
NRZ	Non-return to zero

Syntax	*Example*
LINECODE {HDB3 | NRZ}	IPNOS : CONFIG CONTROLLER E3(4/1)# **linecode hdb3** IPNOS : CONFIG CONTROLLER E3(4/1)# **display linecode** LINECODE : HDB3
LINECODE {HDB3 | NRZ}	IPNOS : CONFIG CONTROLLER E3(4/1)# **linecode nrz** IPNOS : CONFIG CONTROLLER E3(4/1)# **display linecode** LINECODE : NRZ
[NO] LINECODE	IPNOS : CONFIG CONTROLLER E3(4/1)# **no linecode** IPNOS : CONFIG CONTROLLER E3(4/1)# **display linecode** LINECODE : HDB3

Default Value

HDB3

Command Level

controller

Related Commands

commit

loopback

Set the local or remote port into a loopback state. Table 7-5 describes available loopback test options.

Table 7-5 Loopback Parameters

Parameter	*Description*
Local	Loop the router output data back toward the router at the E1 framer and send an alarm indicator signal (AIS) out toward the network.
Dual	Loop the router output data back toward the router at the E1 framer and send the payload data back toward the network at the E1 framer.
Network line	Loop the data back toward the network before the E1 framer. It sends a request to the far end for a network loopback, essentially performing a physical line test.
Network payload	Loop the payload data back toward the network at the E1 framer. This command divides the loopback signals into separate E1 payloads for all 28 E1s channels in the E3 port.
No	Disable loopback and allow normal traffic.

Syntax	*Example*
LOOPBACK {LOCAL | DUAL | NETWORK {LINE | PAYLOAD}}	IPNOS : CONFIG CONTROLLER E3(4/1)# **loopback local** IPNOS : CONFIG CONTROLLER E3(4/1)# **dis loopback** LOOPBACK : LOCAL(NEAR-END)
LOOPBACK {LOCAL | DUAL | NETWORK {LINE | PAYLOAD}}	IPNOS : CONFIG CONTROLLER E3(4/1)# **loopback dual** IPNOS : CONFIG CONTROLLER E3(4/1)# **dis loopback** LOOPBACK : LOCAL(NEAR-END) LINE(NEAR-END)
LOOPBACK {LOCAL | DUAL | NETWORK {LINE | PAYLOAD}}	IPNOS : CONFIG CONTROLLER E3(4/1)# **loopback network line** IPNOS : CONFIG CONTROLLER E3(4/1)# dis loopback LOOPBACK : LINE(NEAR-END)

Syntax	*Example*
LOOPBACK {LOCAL | DUAL | NETWORK {LINE | PAYLOAD}}	IPNOS : CONFIG CONTROLLER E3(4/1)# **loopback network payload** IPNOS : CONFIG CONTROLLER E3(4/1)# **dis loopback** LOOPBACK : PAYLOAD(NEAR-END)
[NO] LOOPBACK	IPNOS : CONFIG CONTROLLER E3(4/1)# **no loopback** IPNOS : CONFIG CONTROLLER E3(4/1)# **dis loopback** LOOPBACK : NONE

Default Value

No loopback

Command Level

controller

Related Commands

controller loopback (channelized DS3 blade)
interface serial loopback (unchannelized DS3 blade)
interface serial loopback (unchannelized E3 blade)

national bit

Enable or disable the E3 national bit setting. When enabled the G.751 frame setting is set to 12 bits.

Syntax	*Example*
`NATIONAL BIT {0 | 1}`	`IPNOS : CONFIG CONTROLLER E3(4/1)#` **`national bit 0`** `IPNOS : CONFIG CONTROLLER E3(4/1)#` **`display national`** `NATIONAL BIT  : 0`
`NATIONAL BIT {0 | 1}`	`IPNOS : CONFIG CONTROLLER E3(4/1)#` **`national bit 1`** `IPNOS : CONFIG CONTROLLER E3(4/1)#` **`display national`** `NATIONAL BIT  : 0`

Default Value

0 (disabled)

Command Level

controller

Related Commands

e1 (E1 national bits command)

shutdown

Shut down an E3 port.

Syntax	*Example*
[NO] SHUTDOWN	IPNOS : CONFIG CONTROLLER E3(4/1)# **shutdown** Change will take effect only after commit IPNOS : CONFIG CONTROLLER E3(4/1)# **display interface** INTERFACE : DISABLE
[NO] SHUTDOWN	IPNOS : CONFIG CONTROLLER E3(4/1)# **no shutdown** Change will take effect only after commit IPNOS : CONFIG CONTROLLER E3(4/1)# **display interface** INTERFACE : ENABLE

Default Value

None

Command Level

controller

Related Commands

commit
controller shutdown (DS3 channelized blade)
interface serial shutdown (DS3 unchannelized blade)
interface serial shutdown (E3 unchannelized blade)

serial

Go to the E3 channel interface level. The following commands are available.

- "display" on page 7-16
- "encapsulation" on page 7-18

Syntax	*Example*

`INTERFACE SERIAL <slot>/<port>:<channel>`	`IPNOS : CONFIG#` **`interface serial 4/1:4`**
	
	`IPNOS : CONFIG INTERFACE SERIAL(4/1:4)#` **`dis`**
	`SLOT : 2`
	`PORT : 1`
	`CHANNEL : 4`
	`ENCAPSULATION : NONE`

Default Value

None

Command Level

config command

Related Commands

interface ethernet (control blade)
interface serial (channelized DS3 blade)
interface serial (unchannelized DS3 blade)
interface serial (unchannelized E3 blade)

display

View current blade encapsulation.

Syntax	*Example*
With no encapsulation: DISPLAY	IPNOS : CONFIG INTERFACE SERIAL(4/1:4)# **display** SLOT : 2 PORT : 1 CHANNEL : 4 ENCAPSULATION : NONE MDL TRANSMIT: PATH : ENABLE IDLE-SIGNAL : ENABLE TEST-SIGNAL : DISABLE
With PPP encapsulation DISPLAY	IPNOS : CONFIG INTERFACE SERIAL(4/1:4)# **display** SLOT : 2 PORT : 1 CHANNEL : 4 ENCAPSULATION : PPP AUTHENTICATION : NONE CONFIG-REQ-RETRANSMIT: 10 TERM-REQ-RETRANSMIT: 2 TIMEOUT : 5 REAUTH : 0 ADMIN-STATUS : ENABLE OPER-STATUS : DOWN

Syntax	*Example*
With Frame Relay encapsulation: DISPLAY	IPNOS : CONFIG INTERFACE SERIAL(4/1:4)# **display** SLOT : 2 PORT : 1 CHANNEL : 4 ENCAPSULATION : FRAME-RELAY LMI-TYPE : ANSI LMI-E391DTE : 10 LMI-N391DTE : 5 LMI-N392DTE : 3 LMI-N393DTE : 4 INTF-TYPE : DTE LMI-N392DCE : 3 LMI-N393DCE : 4 LMI-E392DCE : 15 ADDRESS : Q922 LENGTH : TWO-OCTETS MAX-SUPPORTED-VC: 317 MULTICAST : NON-BROADCAST STATUS : RUNNING FRAME-RELAY PVC(s): List is empty
With HDLC encapsulation: DISPLAY	IPNOS : CONFIG INTERFACE SERIAL(4/1:4)# **display** SLOT : 2 PORT : 1 CHANNEL : 4 ENCAPSULATION : HDLC KEEPALIVE : 10 ADMIN-STATUS : ENABLE OPER-STATUS : DOWN

Default Value

None

Command Level

serial

Related Commands

interface serial display (channelized DS3 blade)
interface serial display (unchannelized DS3 blade)
interface serial display (unchannelized E3 blade)
show

encapsulation

Encapsulate the channel and go to the PPP, Frame Relay, or HDLC command level. See the following chapters for commands.

- Chapter 12, “Frame Relay Commands”
- Chapter 13, “HDLC Commands”
- Chapter 14, “PPP Commands”

Syntax	*Example*
ENCAPSULATION {FRAME-RELAY | HDLC | PPP}	IPNOS : CONFIG INTERFACE SERIAL(4/1:4)# **encapsulation ppp** INTERFACE SERIAL(4/1:4): Object successfully set IPNOS : CONFIG INTERFACE SERIAL(4/1:4) PPP# display SLOT : 2 PORT : 1 CHANNEL : 4 AUTHENTICATION : NONE CONFIG-REQ-RETRANSMIT: 10 TERM-REQ-RETRANSMIT: 2 TIMEOUT : 5 REAUTH : 0 ADMIN-STATUS : ENABLE OPER-STATUS : DOWN IPNOS : CONFIG INTERFACE SERIAL(4/1:4) PPP#

Syntax	*Example*
ENCAPSULATION {FRAME-RELAY | HDLC | PPP}	IPNOS : CONFIG INTERFACE SERIAL(4/1:4)# **encapsulation frame-relay** INTERFACE SERIAL(4/1:4): Object successfully set IPNOS : CONFIG INTERFACE SERIAL(4/1:4) FRAME-RELAY# **display** SLOT : 2 PORT : 1 CHANNEL : 4 LMI-TYPE : ANSI LMI-E391DTE : 10 LMI-N391DTE : 5 LMI-N392DTE : 3 LMI-N393DTE : 4 INTF-TYPE : DTE LMI-N392DCE : 3 LMI-N393DCE : 4 LMI-E392DCE : 15 ADDRESS : Q922 LENGTH : TWO-OCTETS MAX-SUPPORTED-VC: 317 MULTICAST : NON-BROADCAST STATUS : RUNNING IPNOS : CONFIG INTERFACE SERIAL(4/1:4) FRAME-RELAY#
ENCAPSULATION {FRAME-RELAY | HDLC | PPP}	IPNOS : CONFIG INTERFACE SERIAL(4/1:4)# **encapsulation hdlc** INTERFACE SERIAL(4/1:4): Object successfully set IPNOS : CONFIG INTERFACE SERIAL(4/1:4) HDLC# **display** SLOT : 2 PORT : 1 CHANNEL : 4 KEEPALIVE : 10 ADMIN-STATUS : ENABLE OPER-STATUS : DOWN IPNOS : CONFIG INTERFACE SERIAL(4/1:4) HDLC#
[NO] ENCAPSULATION	IPNOS : CONFIG INTERFACE SERIAL(4/1:4)# **no encapsulation** IPNOS : CONFIG INTERFACE SERIAL(4/1:4)# **display** SLOT : 2 PORT : 1 CHANNEL : 4 ENCAPSULATION : NONE IPNOS : CONFIG INTERFACE SERIAL(4/1:4)#

Default Value

None

Command Level

serial

Related Commands

interface ethernet encapsulation (control blade)
interface serial encapsulation (channelized DS3 blade)
interface serial encapsulation (unchannelized DS3 blade)
interface serial encapsulation (unchannelized E3 blade)

CHAPTER

8 Unchannelized E3 Blade Commands

Use these commands to configure the unchannelized E3 blade. This chapter describes the following commands:

Command	*Page*
serial	8-2
clock	8-2
display	8-3
encapsulation	8-4
loopback	8-7
shutdown	8-9

serial

Go to the interface command level. The following commands are available.

- "clock" on page 8-2
- "display" on page 8-3
- "encapsulation" on page 8-4
- "loopback" on page 8-7
- "shutdown" on page 8-9

Syntax	*Example*
`INTERFACE SERIAL <slot>/<port>`	`IPNOS : CONFIG#` **`interface serial 4/1`** `IPNOS : CONFIG INTERFACE SERIAL(2/1)#`

Default Value

None

Command Level

config

Related Commands

interface ethernet

clock

Set the source of the clock for internal or external (line).

Syntax	*Example*
`CLOCK SOURCE {INTERNAL | LINE}`	`IPNOS : CONFIG INTERFACE SERIAL(2/1)#` **`clock source internal`** `IPNOS : CONFIG INTERFACE SERIAL(2/1)#` **`display clock`** `CLOCK : INTERNAL`
`CLOCK SOURCE {INTERNAL | LINE}`	`IPNOS : CONFIG INTERFACE SERIAL(2/1)#` **`clock source line`** `IPNOS : CONFIG INTERFACE SERIAL(2/1)#` **`display clock`** `CLOCK : LINE`

Default Value

internal

Command Level

serial

Related Commands

controller clock (channelized DS3 blade)
interface serial clock (unchannelized DS3 blade)

display

View the current E3 blade configuration. Table 8-1 describes the performance monitoring parameters that come up using the **display tabular** command.

Table 8-1 Performance Monitoring Parameters

Parameter	*Description*
LCV	Line code violations
LES	Line errored seconds
LSES	Line severely errored seconds

Syntax	*Example*

DISPLAY	`IPNOS : CONFIG INTERFACE SERIAL(2/1)# display`
	`SLOT : 4`
	`PORT : 1`
	`ENCAPSULATION : NONE`
	
	`FRAMING : G751`
	`INTERFACE : ENABLE`
	`LOOPBACK : NONE`
	`CLOCK : INTERNAL`
	
	`No Alarm detected.`

Syntax	*Example*
DISPLAY [<argument>]	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **display framing** FRAMING : G751
DISPLAY TABULAR	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **display tabular** Current interval time elapsed: 3 min. and 16 sec. INTERVAL LCV LES LSES current 0 196 196 1 0 900 900 2 0 900 900 3 0 900 900 ... 18 0 900 900 19 0 900 900 20 0 900 900

Default Value

None

Command Level

serial

Related Commands

controller display (channelized DS3 blade)
interface serial display (unchannelized DS3 blade)

encapsulation

Encapsulate the channel and go to the PPP, Frame Relay, or HDLC command level. Each port is considered one channel. See the following chapters for commands.

- Chapter 12, “Frame Relay Commands”
- Chapter 13, “HDLC Commands”
- Chapter 14, “PPP Commands”

Syntax	*Example*
ENCAPSULATION {FRAME-RELAY | HDLC | PPP}	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **encap frame-relay** INTERFACE SERIAL(2/1): Object successfully set IPNOS : CONFIG INTERFACE SERIAL(2/1) FRAME-RELAY# **up** FRAME-RELAY: Object successfully set IPNOS : CONFIG INTERFACE SERIAL(2/1)# **display** SLOT : 2 PORT : 1 ENCAPSULATION : FRAME-RELAY LMI-TYPE : ANSI LMI-T391DTE : 10 LMI-N391DTE : 5 LMI-N392DTE : 3 LMI-N393DTE : 4 INTF-TYPE : DTE LMI-N392DCE : 3 LMI-N393DCE : 4 LMI-T392DCE : 15 ADDRESS : Q922 LENGTH : TWO-OCTETS MAX-SUPPORTED-VC: 317 MULTICAST : NON-BROADCAST STATUS : RUNNING FRAME-RELAY PVC(s): List is empty FRAMING : G751 INTERFACE : ENABLE LOOPBACK : NONE CLOCK : INTERNAL Receiver has loss of signal.

Syntax	*Example*
ENCAPSULATION {FRAME-RELAY | HDLC | PPP}	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **encap hdlc** INTERFACE SERIAL(2/1): Object successfully set IPNOS : CONFIG INTERFACE SERIAL(2/1) HDLC# **up** HDLC: Object successfully set IPNOS : CONFIG INTERFACE SERIAL(2/1)# **display** SLOT : 2 PORT : 1 ENCAPSULATION : HDLC KEEPALIVE : 10 ADMIN-STATUS : ENABLE OPER-STATUS : DOWN FRAMING : G751 INTERFACE : ENABLE LOOPBACK : NONE CLOCK : INTERNAL Receiver has loss of signal.
ENCAPSULATION {FRAME-RELAY | HDLC | PPP}	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **encap ppp** INTERFACE SERIAL(2/1): Object successfully set IPNOS : CONFIG INTERFACE SERIAL(2/1) PPP# **up** PPP: Object successfully set IPNOS : CONFIG INTERFACE SERIAL(2/1)# **display** SLOT : 2 PORT : 1 ENCAPSULATION : PPP AUTHENTICATION : NONE CONFIG-REQ-RETRANSMIT: 10 TERM-REQ-RETRANSMIT: 2 TIMEOUT : 5 REAUTH : 0 ADMIN-STATUS : ENABLE OPER-STATUS : DOWN FRAMING : G751 INTERFACE : ENABLE LOOPBACK : NONE CLOCK : INTERNAL Receiver has loss of signal.

Syntax	Example
[NO] ENCAPSULATION	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **no encap** IPNOS : CONFIG INTERFACE SERIAL(2/1)# **display** SLOT : 2 PORT : 1 ENCAPSULATION : NONE FRAMING : G751 INTERFACE : ENABLE LOOPBACK : NONE CLOCK : INTERNAL Receiver has loss of signal.

Default Value

None

Command Level

serial

Related Commands

interface ethernet encapsulation (control blade)
serial encapsulation (channelized DS3 blade)
interface serial encapsulation (unchannelized DS3 blade)

loopback

Set the local or remote port into a loopback state. Table 8-2 describes parameters available.

Table 8-2 Loopback Parameters

Parameter	Description
Local	Loops the router output data back toward the router at the E3 framer and sends an alarm indicator signal (AIS) signal out toward the network.
Dual	Loops the router output data back toward the router at the E3 framer and loops the payload data back toward the network at the E3 framer.

Table 8-2 Loopback Parameters (Continued)

Parameter	*Description*
Network line	Loops the data back toward the network before the E3 framer. It sends a request to the far end for a network loopback, essentially performing a physical line test. The **line** option can only be used when C-bit framing is used in the DS3.
No	Disable loopback.

Syntax	*Example*
LOOPBACK {LOCAL | DUAL | NETWORK LINE}	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **loopback local** IPNOS : CONFIG INTERFACE SERIAL(2/1)# **display loopback** LOOPBACK : LOCAL(NEAR-END)
LOOPBACK {LOCAL | DUAL | NETWORK LINE}	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **loopback dual** IPNOS : CONFIG INTERFACE SERIAL(2/1)# **display loopback** LOOPBACK : LOCAL(NEAR-END) LINE(NEAR-END)
LOOPBACK {LOCAL | DUAL | NETWORK LINE}	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **loopback network line** IPNOS : CONFIG INTERFACE SERIAL(2/1)# **display loopback** LOOPBACK : LINE(NEAR-END)
NO LOOPBACK	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **no loopback** IPNOS : CONFIG INTERFACE SERIAL(2/1)# **display loopback** LOOPBACK : NONE

Default Value

no loopback

Command Level

controller

Related Commands

controller loopback (channelized E3 blade)
controller loopback (channelized DS3 blade)
interface serial loopback (unchannelized DS3 blade)

shutdown

Shut down an E3 port.

Syntax	*Example*
[NO] SHUTDOWN	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **shutdown** Change will take effect only after commit IPNOS : CONFIG INTERFACE SERIAL(2/1)# **commit** Commit configuration INTERFACE SERIAL(2/1): Object successfully set IPNOS : CONFIG# **interface serial 2/1** IPNOS : CONFIG INTERFACE SERIAL(2/1)# **display interface** INTERFACE : DISABLE
[NO] SHUTDOWN	IPNOS : CONFIG INTERFACE SERIAL(2/1)# **no shutdown** Change will take effect only after commit IPNOS : CONFIG INTERFACE SERIAL(2/1)# **commit** Commit configuration INTERFACE SERIAL(2/1): Object successfully set IPNOS : CONFIG# **interface serial 2/1** IPNOS : CONFIG INTERFACE SERIAL(2/1)# **display interface** INTERFACE : ENABLE

Default Value

Shuts down the specific port indicated in your command prompt.

Command Level

serial

Related Commands

commit
controller shutdown (channelized DS3 blade)
interface serial shutdown (unchannelized DS3 blade)
controller shutdown (channelized E3 blade)

CHAPTER

9 OC-3c Commands

Use these commands to configure the Optical Carrier 3 concatenated (OC-3c) blade interface. You need to be in one of the following command levels.

"atm" on page 10-3

"pos" on page 11-2

This chapter describes the following command:

Command	*Page*
oc3	9-2

oc3

Configure the OC-3c blade physical interface. Table 9-1 describes OC-3c parameters.

Table 9-1 OC-3c Parameters

Parameter	*Description*
Alarm	Enable alarm reporting to the Network Management System (NMS). The default value is enabled.
Clock	Set the source of the clock for internal or external (line). If another node in the network is to be the master clock source, set the clock to line. The OC-3c blade supports operation as a master clock source for individual ports, not for the entire IPSX switch. The default value is local.
CRC	Set the Cyclic Redundancy Check (CRC) type. The values are 16 or 32 bit. The default value is 16. **NOTE:** This parameter is available only on OC-3c blades running POS.
Enable	Enable OC-3c port. The default value is enabled.
Framing	Set the framing to Synchronous Optical Network (SONET) STS-3c or Synchronous Digital Hierarchy (SDH) STM-1. SDH is the European version of SONET. The default value is SONET.
J0	Set the section trace J0 byte of the SONET frame. The default J0 value is 01xCC for most networks. Set the J0 value to 0x01 for connecting to a Japanese ATM network.
Loop	Set the loopback on the OC-3c interface. The default value is none.
Max-packet-size	Set the maximum size of the POS packet. The default value is 1792 bytes. **NOTE:** This parameter is available only on OC-3c blades running POS.
Path-trace	Set the label of the local path terminating equipment (PTE). This label is a string that is sent to the other end of the SONET connection to check the physical path. It is expected to match the remote-path-trace label of the Remote PTE. A mismatch alarm is raised if the labels do not match. The label in SONET uses a 64-byte string, while SDH uses a 16-byte string. It is located in the J1 byte of the SONET frame. The label can hold alphanumeric characters, spaces, carriage returns and line feeds. The default value is a null string.

Table 9-1 OC-3c Parameters (Continued)

Parameter	*Description*
Remote-path-trace	Set the label expected from the remote PTE. This label matches the string that is received from the other end of the SONET connection to check the physical path. A mismatch alarm is raised if the labels do not match. The label in SONET uses a 64-byte string, while SDH uses a 16-byte string. It is located in the J1 byte of the SONET frame. The label can hold alphanumeric characters, spaces, carriage returns and line feeds. The default value is a null string.
SCRAMBLE	Enable data scrambling. Scrambling spreads the transmission load evenly across the stream by randomizing the data in the payload. By default, scrambling is enabled.

Syntax	*Example*
OC3 ALARM	IPNOS : CONFIG INTERFACE ATM(1/1)# **oc3 alarm** OC-3(1/1): Object successfully set IPNOS : CONFIG INTERFACE ATM(1/1)# **display oc3** OC-3 PARAMETERS FRAMING : SONET CLOCK : LINE J0 : 0xCC SCRAMBLE : ENABLE ALARM : ENABLE PATH-TRACE : REMOTE-PATH-TRACE : LOOP : NONE ADMIN STATUS : ENABLE OPER STATUS : UP - VALID INTERVALS : 12 INVALID INTERVALS : 0 ELAPSED TIME(sec) : 615

Syntax	*Example*
NO OC3 ALARM	IPNOS : CONFIG INTERFACE ATM(1/1)# **no oc3 alarm** OC-3(1/1): Object successfully set IPNOS : CONFIG INTERFACE ATM(1/1)# **display oc3** OC-3 PARAMETERS FRAMING : SONET CLOCK : LINE J0 : 0xCC SCRAMBLE : ENABLE ALARM : DISABLE PATH-TRACE : REMOTE-PATH-TRACE : LOOP : NONE ADMIN STATUS : ENABLE OPER STATUS : UP - VALID INTERVALS : 12 INVALID INTERVALS : 0 ELAPSED TIME(sec) : 615
OC3 CLOCK SOURCE {LINE | INTERNAL}	IPNOS : CONFIG INTERFACE ATM(1/1)# **oc3 clock source line** OC-3(1/1): Object successfully set IPNOS : CONFIG INTERFACE ATM(1/1)# **display oc3** OC-3 PARAMETERS FRAMING : SONET CLOCK : LINE J0 : 0xCC SCRAMBLE : ENABLE ALARM : ENABLE PATH-TRACE : REMOTE-PATH-TRACE : LOOP : NONE ADMIN STATUS : ENABLE OPER STATUS : UP - VALID INTERVALS : 12 INVALID INTERVALS : 0 ELAPSED TIME(sec) : 615

Syntax	Example
OC3 CLOCK SOURCE {LINE | INTERNAL}	IPNOS : CONFIG INTERFACE ATM(1/1)# **oc3 clock source internal** OC-3(1/1): Object successfully set OC-3 PARAMETERS FRAMING : SONET CLOCK : INTERNAL J0 : 0xCC SCRAMBLE : ENABLE ALARM : ENABLE PATH-TRACE : REMOTE-PATH-TRACE : LOOP : NONE ADMIN STATUS : ENABLE OPER STATUS : UP - VALID INTERVALS : 12 INVALID INTERVALS : 0 ELAPSED TIME(sec) : 615
CRC check (POS only): OC3 CRC {16 | 32}	IPNOS : CONFIG INTERFACE POS(4/1)# **oc3 crc 16** OC-3(4/1): Object successfully set IPNOS : CONFIG INTERFACE POS(4/1)# **display oc3** ENCAPSULATION : FR OC-3 PARAMETERS FRAMING : SONET CLOCK : INTERNAL CRC : 16 MAX-PACKET-SIZE : 1792 J0 : 0xCC SCRAMBLE : ENABLE ALARM : ENABLE PATH-TRACE : REMOTE-PATH-TRACE : LOOP : NONE ADMIN STATUS : ENABLE OPER STATUS : UP - VALID INTERVALS : 3 INVALID INTERVALS : 1 ELAPSED TIME(sec) : 36

Syntax	*Example*

CRC check (POS only):
`OC3 CRC {16 | 32}`
	`IPNOS : CONFIG INTERFACE POS(4/1)# oc3 crc 32`
	`OC-3(4/1): Object successfully set`
	
	`IPNOS : CONFIG INTERFACE POS(4/1)# display oc3`
	`ENCAPSULATION : FR`
	
	`OC-3 PARAMETERS`
	`FRAMING : SONET`
	`CLOCK : INTERNAL`
	`CRC : 32`
	`MAX-PACKET-SIZE : 1792`
	`J0 : 0xCC`
	`SCRAMBLE : ENABLE`
	`ALARM : ENABLE`
	`PATH-TRACE :`
	`REMOTE-PATH-TRACE :`
	`LOOP : NONE`
	`ADMIN STATUS : ENABLE`
	`OPER STATUS : UP -`
	
	`VALID INTERVALS : 3`
	`INVALID INTERVALS : 1`
	`ELAPSED TIME(sec) : 106`

`OC3 ENABLE`	`IPNOS : CONFIG INTERFACE ATM(1/1)# oc3 enable`
	`OC-3(1/1): Object successfully set`
	
	`IPNOS : CONFIG INTERFACE ATM(1/1)# display oc3`
	
	`OC-3 PARAMETERS`
	`FRAMING : SDH`
	`CLOCK : NONE`
	`J0 : 0xCC`
	`SCRAMBLE : ENABLE`
	`ALARM : ENABLE`
	`PATH-TRACE :`
	`REMOTE-PATH-TRACE :`
	`LOOP : NONE`
	`ADMIN STATUS : ENABLE`
	`OPER STATUS : UP -`
	
	`VALID INTERVALS : 12`
	`INVALID INTERVALS : 0`
	`ELAPSED TIME(sec) : 615`

Syntax	Example
NO OC3 ENABLE	IPNOS : CONFIG INTERFACE ATM(1/1)# **no oc3 enable** OC-3(1/1): Object successfully set IPNOS : CONFIG INTERFACE ATM(1/1)# **display oc3** OC-3 PARAMETERS FRAMING : SONET CLOCK : NONE J0 : 0xCC SCRAMBLE : ENABLE ALARM : ENABLE PATH-TRACE : REMOTE-PATH-TRACE : LOOP : NONE ADMIN STATUS : DISABLE OPER STATUS : DOWN - LOS_S VALID INTERVALS : 12 INVALID INTERVALS : 0 ELAPSED TIME(sec) : 615
OC3 FRAMING {SDH | SONET}	IPNOS : CONFIG INTERFACE ATM(1/1)# **oc3 framing sdh** OC-3(1/1): Object successfully set IPNOS : CONFIG INTERFACE ATM(1/1)# **display oc3** OC-3 PARAMETERS FRAMING : SDH CLOCK : NONE J0 : 0xCC SCRAMBLE : ENABLE ALARM : ENABLE PATH-TRACE : REMOTE-PATH-TRACE : LOOP : NONE ADMIN STATUS : ENABLE OPER STATUS : UP - VALID INTERVALS : 12 INVALID INTERVALS : 0 ELAPSED TIME(sec) : 615

Syntax	*Example*
OC3 FRAMING {SDH | SONET}	IPNOS : CONFIG INTERFACE ATM(1/1)# **oc3 framing sonet** OC-3(1/1): Object successfully set IPNOS : CONFIG INTERFACE ATM(1/1)# **display oc3** OC-3 PARAMETERS FRAMING : SONET CLOCK : NONE J0 : 0xCC SCRAMBLE : ENABLE ALARM : ENABLE PATH-TRACE : REMOTE-PATH-TRACE : LOOP : NONE ADMIN STATUS : ENABLE OPER STATUS : UP - VALID INTERVALS : 12 INVALID INTERVALS : 0 ELAPSED TIME(sec) : 615
OC3 J0 {01 | CC}	IPNOS : CONFIG INTERFACE ATM(1/1)# **oc3 j0 01** OC-3(1/1): Object successfully set OC-3 PARAMETERS FRAMING : SONET CLOCK : NONE J0 : 0x01 SCRAMBLE : ENABLE ALARM : ENABLE PATH-TRACE : REMOTE-PATH-TRACE : LOOP : NONE ADMIN STATUS : ENABLE OPER STATUS : UP - VALID INTERVALS : 12 INVALID INTERVALS : 0 ELAPSED TIME(sec) : 615

Syntax	Example
OC3 J0 {01 | CC}	IPNOS : CONFIG INTERFACE ATM(1/1)# **oc3 j0 cc** OC-3(1/1): Object successfully set IPNOS : CONFIG INTERFACE ATM(1/1)# **display oc3** OC-3 PARAMETERS FRAMING : SONET CLOCK : NONE J0 : 0xCC SCRAMBLE : ENABLE ALARM : ENABLE PATH-TRACE : REMOTE-PATH-TRACE : LOOP : NONE ADMIN STATUS : ENABLE OPER STATUS : UP - VALID INTERVALS : 12 INVALID INTERVALS : 0 ELAPSED TIME(sec) : 615
OC3 LOOP INTERNAL	IPNOS : CONFIG INTERFACE ATM(1/1)# **oc3 loop internal** OC-3(1/1): Object successfully set IPNOS : CONFIG INTERFACE ATM(1/1)# **display oc3** OC-3 PARAMETERS FRAMING : SONET CLOCK : INTERNAL J0 : 0xCC SCRAMBLE : ENABLE ALARM : ENABLE PATH-TRACE : REMOTE-PATH-TRACE : LOOP : INTERNAL ADMIN STATUS : ENABLE OPER STATUS : UP - VALID INTERVALS : 12 INVALID INTERVALS : 0 ELAPSED TIME(sec) : 615

Syntax	*Example*
OC3 LOOP LINE	IPNOS : CONFIG INTERFACE ATM(1/1)# **oc3 loop line** OC-3(1/1): Object successfully set IPNOS : CONFIG INTERFACE ATM(1/1)# **display oc3** OC-3 PARAMETERS FRAMING : SONET CLOCK : INTERNAL J0 : 0xCC SCRAMBLE : ENABLE ALARM : ENABLE PATH-TRACE : REMOTE-PATH-TRACE : LOOP : LINE ADMIN STATUS : ENABLE OPER STATUS : UP - VALID INTERVALS : 12 INVALID INTERVALS : 0 ELAPSED TIME(sec) : 615
NO OC3 LOOP	IPNOS : CONFIG INTERFACE ATM(1/1)# **no oc3 loop** OC-3(1/1): Object successfully set IPNOS : CONFIG INTERFACE ATM(1/1)# **display oc3** OC-3 PARAMETERS FRAMING : SONET CLOCK : INTERNAL J0 : 0xCC SCRAMBLE : ENABLE ALARM : ENABLE PATH-TRACE : REMOTE-PATH-TRACE : LOOP : NONE ADMIN STATUS : ENABLE OPER STATUS : UP - VALID INTERVALS : 12 INVALID INTERVALS : 0 ELAPSED TIME(sec) : 615

Syntax	Example
POS only: max-packet-size {4..1792}	IPNOS : CONFIG INTERFACE POS(4/1)# **oc3 max-packet-size 512** OC-3(4/1): Object successfully set IPNOS : CONFIG INTERFACE POS(4/1)# **display** SLOT : 4 PORT : 1 ENCAPSULATION : FR OC-3 PARAMETERS FRAMING : SONET CLOCK : INTERNAL CRC : 16 MAX-PACKET-SIZE : 512 J0 : 0xCC SCRAMBLE : ENABLE ALARM : ENABLE PATH-TRACE : REMOTE-PATH-TRACE : LOOP : NONE ADMIN STATUS : ENABLE OPER STATUS : UP - VALID INTERVALS : 4 INVALID INTERVALS : 1 ELAPSED TIME(sec) : 275
[NO] OC3 PATH-TRACE <string>	IPNOS : CONFIG INTERFACE ATM(1/1)# **oc3 path-trace NYC Sw5 P4** OC-3(1/1): Object successfully set IPNOS : CONFIG INTERFACE ATM(1/1)# display oc3 OC-3 PARAMETERS FRAMING : SONET CLOCK : INTERNAL J0 : 0xCC SCRAMBLE : ENABLE ALARM : ENABLE PATH-TRACE : NYC Sw5 P4 REMOTE-PATH-TRACE : LOOP : NONE ADMIN STATUS : ENABLE OPER STATUS : UP - VALID INTERVALS : 12 INVALID INTERVALS : 0 ELAPSED TIME(sec) : 615

Syntax	*Example*
[NO] OC3 PATH-TRACE	IPNOS : CONFIG INTERFACE ATM(1/1)# **no oc3 path-trace** OC-3(1/1): Object successfully set IPNOS : CONFIG INTERFACE ATM(1/1)# display oc3 OC-3 PARAMETERS FRAMING : SONET CLOCK : INTERNAL J0 : 0xCC SCRAMBLE : ENABLE ALARM : ENABLE PATH-TRACE : REMOTE-PATH-TRACE : LOOP : NONE ADMIN STATUS : ENABLE OPER STATUS : UP - VALID INTERVALS : 12 INVALID INTERVALS : 0 ELAPSED TIME(sec) : 618
[NO] OC3 REMOTE-PATH-TRACE <string>	IPNOS : CONFIG INTERFACE ATM(1/1)# **oc3 remote-path-trace Den Sw3 P4** OC-3(1/1): Object successfully set IPNOS : CONFIG INTERFACE ATM(1/1)# **display oc3** OC-3 PARAMETERS FRAMING : SONET CLOCK : INTERNAL J0 : 0xCC SCRAMBLE : ENABLE ALARM : ENABLE PATH-TRACE : NYC Sw5 P4 REMOTE-PATH-TRACE : Den Sw3 P4 LOOP : NONE ADMIN STATUS : ENABLE OPER STATUS : UP - VALID INTERVALS : 12 INVALID INTERVALS : 0 ELAPSED TIME(sec) : 620

Syntax	Example
[NO] OC3 REMOTE-PATH-TRACE	IPNOS : CONFIG INTERFACE ATM(1/1)# **no oc3 remote-path-trace** OC-3(1/1): Object successfully set IPNOS : CONFIG INTERFACE ATM(1/1)# **display oc3** OC-3 PARAMETERS FRAMING : SONET CLOCK : INTERNAL J0 : 0xCC SCRAMBLE : ENABLE ALARM : ENABLE PATH-TRACE : NYC Sw5 P4 REMOTE-PATH-TRACE : Den Sw3 P4 LOOP : NONE ADMIN STATUS : ENABLE OPER STATUS : UP - VALID INTERVALS : 12 INVALID INTERVALS : 0 ELAPSED TIME(sec) : 620
OC3 SCRAMBLE	IPNOS : CONFIG INTERFACE ATM(1/1)# **oc3 scramble** OC-3(1/1): Object successfully set IPNOS : CONFIG INTERFACE ATM(1/1)# **display oc3** OC-3 PARAMETERS FRAMING : SONET CLOCK : INTERNAL J0 : 0xCC SCRAMBLE : ENABLE ALARM : ENABLE PATH-TRACE : REMOTE-PATH-TRACE : LOOP : NONE ADMIN STATUS : ENABLE OPER STATUS : UP - VALID INTERVALS : 12 INVALID INTERVALS : 0 ELAPSED TIME(sec) : 630

Syntax	*Example*
NO OC3 SCRAMBLE	IPNOS : CONFIG INTERFACE ATM(1/1)# **no oc3 scramble** OC-3(1/1): Object successfully set IPNOS : CONFIG INTERFACE ATM(1/1)# **display oc3** OC-3 PARAMETERS FRAMING : SONET CLOCK : INTERNAL J0 : 0xCC SCRAMBLE : DISABLE ALARM : ENABLE PATH-TRACE : REMOTE-PATH-TRACE : LOOP : NONE ADMIN STATUS : ENABLE OPER STATUS : UP - VALID INTERVALS : 12 INVALID INTERVALS : 0 ELAPSED TIME(sec) : 615

Default Value

See Table 9-1

Command Level

atm and pos interface

Related Commands

atm interface
pos interface

CHAPTER

10 ATM Commands

Use these commands to configure Asynchronous Transfer Mode (ATM). This chapter describes the following commands:

Command	*Page*
atm	10-3
atm	10-4
display	10-17
atm	10-23
arp	10-24
arp-age	10-28
arp-max-entries	10-29
display	10-32
enable	10-33
encapsulation	10-34
icr	10-36
ip-addr	10-37
mbs	10-38
mcr	10-40
oam-loopback	10-41
pcr	10-41
pppoe	10-42
display	10-43
pppoe	10-45
protocol	10-50

Command	Page
receive-size	10-54
scr	10-55
service-category	10-56
transmit-size	10-65
vci	10-66
vpi	10-69

atm

Go to the ATM interface command level. The following commands are available.

- "atm" on page 10-4 (ATM interface)
- "display" on page 10-17
- "oc3" on page 9-2 (OC-3c blade physical interface)

Syntax	*Example*
Configure OC-3c and ATM interfaces: `INTERFACE ATM <slot>/<port>`	`IPNOS : CONFIG#` **`interface atm 2/1`** `IPNOS : CONFIG INTERFACE ATM(2/1)#`

Default Value

None

Command Level

interface command

Related Commands

oc3 interface
atm interface
atm subinterface

atm

Configure asynchronous transfer mode (ATM) interface. Table 10-1 describes the parameters available.

Table 10-1 ATM Parameters

Parameter	*Description*
Enable	Enable administrative status. The default value is disabled.
If-name	Assign interface name. The default value is none.
ILMI-enable	Activate the integrated local management interface (ILMI). The default value is disabled.
ILMI-VCI	Specify a virtual channel to carry ILMI traffic. Channels 0–31 are recommended. Channel 16 is the default.
ILMI-VPI	Specify a virtual path to carry ILMI traffic. The ILMI-VPI is always set to 0.
Maxvc-number	Set maximum virtual channel connections (VCCs). The default value is 16384 VCCs per VPC. Determine the max-VCC number by the VCI address range: • 14 bits address = 16384 VCCs per VPC • 13 bit address = 8192 VCCs per VPC • 12 bits address = 4096 VCCs per VPC • 6 bits address = 64 VCCs per VPC
Maxvci-bits	Set the 14-bit address range for virtual channels. VCI bits are closely linked to VPI bits, and scale within a total combined limit of 14 bits. Currently only 6, 12, 13, and 14 VCI bits are supported; these support 0, 1, 2, and 8 VPI bits, respectively. The default value is 14 bits. The address ranges limit the number of virtual path connections (VPCs) and virtual channel connections (VCCs), which scale together in the following manner: • 14 VCI bits, 0 VPI bits • 13 VCI bits, 1 VPI bit • 12 VCI bits, 2 VPI bits • 6 VCI bits, 8 VPI bits
Maxvp-number	Set the maximum virtual path connections (VPCs). The default value is 0 bits. The maximum number of VPCs are determined by the current setting of the 8-bit VPI address range: • 0 bits = 1 VPC • 1 bit = 2 VPCs • 2 bits = 4 VPCs • 8 bits = 256 VPCs

Table 10-1 ATM Parameters (Continued)

Parameter	*Description*
Maxvpi-bits	Set the 8-bit address range for virtual paths. VPI bits and VCI bits are linked together and scale within a total combined limit of 14 bits. Currently only 0, 1, 2, and 8 VPI bits are supported; these support 14, 13, 12 and 6 VCI bits respectively. The default value is 0 bits. The address ranges limit the number of VPCs and VCCs, which scale together in the following manner: • 0 VPI bits, 14 VCI bits • 1 VPI bit, 13 VCI bits • 2 VPI bits, 12 VCI bits • 8 VPI bits, 6 VCI bits
Neighbor-ip-address	Assign the IP address of neighbor ATM interface. The default value is 0.0.0.0.
Neighbor-name	Assign the name of neighbor ATM interface. The default value is none.

Syntax	*Example*

ATM ENABLE	`IPNOS : CONFIG INTERFACE ATM(1/1)# atm enable`
	
	`IPNOS : CONFIG INTERFACE ATM(1/1)# display atm`
	
	`ATM PARAMETERS`
	`IF-NAME : `
	`MAXVPI-BITS : 0 (0 VPI bits currently active)`
	`MAXVP-NUMBER : 0 (0 VPCs currently configured)`
	`MAXVCI-BITS : 12 (12 VCI bits currently active)`
	`MAXVC-NUMBER : 4096 (0 VCCs currently configured)`
	`ILMI-VPI : 0`
	`ILMI-VCI : 16`
	`ILMI VERSION : VERSION 4.0`
	`ILMI ADMIN STATUS : DISABLE`
	`ILMI OPER STATUS : DOWN`
	`NEIGHBOR-IP-ADDRESS : 0.0.0.0`
	`NEIGHBOR-NAME : `
	`LINK ADMIN STATUS : ENABLE`
	`LINK OPER STATUS : UP`

Syntax	*Example*
NO ATM ENABLE	IPNOS : CONFIG INTERFACE ATM(1/1)# **no atm enable** IPNOS : CONFIG INTERFACE ATM(1/1)# **display atm** ATM PARAMETERS IF-NAME : MAXVPI-BITS : 0 (0 VPI bits currently active) MAXVP-NUMBER : 0 (0 VPCs currently configured) MAXVCI-BITS : 12 (12 VCI bits currently active) MAXVC-NUMBER : 4096 (0 VCCs currently configured) ILMI-VPI : 0 ILMI-VCI : 16 ILMI VERSION : VERSION 4.0 ILMI ADMIN STATUS : DISABLE ILMI OPER STATUS : DOWN NEIGHBOR-IP-ADDRESS : 0.0.0.0 NEIGHBOR-NAME : LINK ADMIN STATUS : DISABLE LINK OPER STATUS : UP
ATM IF-NAME <string>	IPNOS : CONFIG INTERFACE ATM(1/1)# **atm if-name Acme1** IPNOS : CONFIG INTERFACE ATM(1/1)# **display atm** SLOT : 1 PORT : 1 ATM PARAMETERS IF-NAME : Acme1 MAXVPI-BITS : 0 (0 VPI bits currently active) MAXVP-NUMBER : 256 (0 VPCs currently configured) MAXVCI-BITS : 6 (12 VCI bits currently active) MAXVC-NUMBER : 4096 (0 VCCs currently configured) ILMI-VPI : 16 ILMI-VCI : 30 ILMI VERSION : VERSION 4.0 ILMI ADMIN STATUS : DISABLE ILMI OPER STATUS : DOWN NEIGHBOR-IP-ADDRESS : 0.0.0.0 NEIGHBOR-NAME : LINK ADMIN STATUS : DISABLE LINK OPER STATUS : UP

Syntax	Example
[NO] ATM IF-NAME	IPNOS : CONFIG INTERFACE ATM(1/1)# **no atm if-name** IPNOS : CONFIG INTERFACE ATM(1/1)# **display atm** ATM PARAMETERS IF-NAME : MAXVPI-BITS : 0 (0 VPI bits currently active) MAXVP-NUMBER : 256 (0 VPCs currently configured) MAXVCI-BITS : 6 (12 VCI bits currently active) MAXVC-NUMBER : 4096 (0 VCCs currently configured) ILMI-VPI : 16 ILMI-VCI : 30 ILMI VERSION : VERSION 4.0 ILMI ADMIN STATUS : DISABLE ILMI OPER STATUS : DOWN NEIGHBOR-IP-ADDRESS : 0.0.0.0 NEIGHBOR-NAME : LINK ADMIN STATUS : DISABLE LINK OPER STATUS : UP
ATM ILMI-ENABLE	IPNOS : CONFIG INTERFACE ATM(1/1)# **atm ilmi-enable** IPNOS : CONFIG INTERFACE ATM(1/1)# **display atm** ATM PARAMETERS IF-NAME : MAXVPI-BITS : 0 (0 VPI bits currently active) MAXVP-NUMBER : 256 (0 VPCs currently configured) MAXVCI-BITS : 6 (10 VCI bits currently active) MAXVC-NUMBER : 4096 (0 VCCs currently configured) ILMI-VPI : 16 ILMI-VCI : 30 ILMI VERSION : VERSION 4.0 ILMI ADMIN STATUS : ENABLE ILMI OPER STATUS : DOWN NEIGHBOR-IP-ADDRESS : 0.0.0.0 NEIGHBOR-NAME : LINK ADMIN STATUS : DISABLE LINK OPER STATUS : UP

Syntax	*Example*
[NO] ATM ILMI-ENABLE	IPNOS : CONFIG INTERFACE ATM(1/1)# **no atm ilmi-enable** IPNOS : CONFIG INTERFACE ATM(1/1)# **display atm** ATM PARAMETERS IF-NAME : MAXVPI-BITS : 0 (0 VPI bits currently active) MAXVP-NUMBER : 256 (0 VPCs currently configured) MAXVCI-BITS : 6 (10 VCI bits currently active) MAXVC-NUMBER : 4096 (0 VCCs currently configured) ILMI-VPI : 16 ILMI-VCI : 30 ILMI VERSION : VERSION 4.0 ILMI ADMIN STATUS : DISABLE ILMI OPER STATUS : DOWN NEIGHBOR-IP-ADDRESS : 0.0.0.0 NEIGHBOR-NAME : LINK ADMIN STATUS : DISABLE LINK OPER STATUS : UP
ATM ILMI-VCI <vciid>	IPNOS : CONFIG INTERFACE ATM(1/1)# **atm ilmi-vci 30** IPNOS : CONFIG INTERFACE ATM(1/1)# display atm ATM PARAMETERS IF-NAME : MAXVPI-BITS : 0 (0 VPI bits currently active) MAXVP-NUMBER : 0 (0 VPCs currently configured) MAXVCI-BITS : 12 (12 VCI bits currently active) MAXVC-NUMBER : 4096 (0 VCCs currently configured) ILMI-VPI : 0 ILMI-VCI : 30 ILMI VERSION : VERSION 4.0 ILMI ADMIN STATUS : DISABLE ILMI OPER STATUS : DOWN NEIGHBOR-IP-ADDRESS : 0.0.0.0 NEIGHBOR-NAME : LINK ADMIN STATUS : DISABLE LINK OPER STATUS : UP

Syntax	Example
[NO] ATM ILMI-VCI	IPNOS : CONFIG INTERFACE ATM(1/1)# **no atm ilmi-vci** IPNOS : CONFIG INTERFACE ATM(1/1)# **display atm** ATM PARAMETERS IF-NAME : MAXVPI-BITS : 0 (0 VPI bits currently active) MAXVP-NUMBER : 256 (0 VPCs currently configured) MAXVCI-BITS : 6 (10 VCI bits currently active) MAXVC-NUMBER : 4096 (0 VCCs currently configured) ILMI-VPI : 16 ILMI-VCI : 16 ILMI VERSION : VERSION 4.0 ILMI ADMIN STATUS : DISABLE ILMI OPER STATUS : DOWN NEIGHBOR-IP-ADDRESS : 0.0.0.0 NEIGHBOR-NAME : LINK ADMIN STATUS : DISABLE LINK OPER STATUS : UP
ATM ILMI-VPI <vpiid>	IPNOS : CONFIG INTERFACE ATM(1/1)# **atm ilmi-vpi 16** IPNOS : CONFIG INTERFACE ATM(1/1)# **display atm** ATM PARAMETERS IF-NAME : MAXVPI-BITS : 0 (0 VPI bits currently active) MAXVP-NUMBER : 256 (0 VPCs currently configured) MAXVCI-BITS : 6 (10 VCI bits currently active) MAXVC-NUMBER : 4096 (0 VCCs currently configured) ILMI-VPI : 16 ILMI-VCI : 16 ILMI VERSION : VERSION 4.0 ILMI ADMIN STATUS : DISABLE ILMI OPER STATUS : DOWN NEIGHBOR-IP-ADDRESS : 0.0.0.0 NEIGHBOR-NAME : LINK ADMIN STATUS : DISABLE LINK OPER STATUS : UP

Syntax	*Example*
[NO] ATM ILMI-VPI	IPNOS : CONFIG INTERFACE ATM(1/1)# **no atm ilmi-vpi** IPNOS : CONFIG INTERFACE ATM(1/1)# **display atm** ATM PARAMETERS IF-NAME : MAXVPI-BITS : 0 (0 VPI bits currently active) MAXVP-NUMBER : 256 (0 VPCs currently configured) MAXVCI-BITS : 6 (10 VCI bits currently active) MAXVC-NUMBER : 4096 (0 VCCs currently configured) ILMI-VPI : 0 ILMI-VCI : 16 ILMI VERSION : VERSION 4.0 ILMI ADMIN STATUS : DISABLE ILMI OPER STATUS : DOWN NEIGHBOR-IP-ADDRESS : 0.0.0.0 NEIGHBOR-NAME : LINK ADMIN STATUS : DISABLE LINK OPER STATUS : UP
ATM MAXVC-NUMBER <number>	IPNOS : CONFIG INTERFACE ATM(1/1)# **atm maxvc-number 4096** IPNOS : CONFIG INTERFACE ATM(1/1)# **display atm** ATM PARAMETERS IF-NAME : MAXVPI-BITS : 0 (0 VPI bits currently active) MAXVP-NUMBER : 256 (0 VPCs currently configured) MAXVCI-BITS : 6 (10 VCI bits currently active) MAXVC-NUMBER : 4096 (0 VCCs currently configured) ILMI-VPI : 0 ILMI-VCI : 16 ILMI VERSION : VERSION 4.0 ILMI ADMIN STATUS : DISABLE ILMI OPER STATUS : DOWN NEIGHBOR-IP-ADDRESS : 0.0.0.0 NEIGHBOR-NAME : LINK ADMIN STATUS : DISABLE LINK OPER STATUS : UP

Syntax	Example
[NO] ATM MAXVC-NUMBER	IPNOS : CONFIG INTERFACE ATM(1/1)# **no atm maxvc-number** IPNOS : CONFIG INTERFACE ATM(1/1)# **display atm** ATM PARAMETERS IF-NAME : MAXVPI-BITS : 0 (0 VPI bits currently active) MAXVP-NUMBER : 256 (0 VPCs currently configured) MAXVCI-BITS : 6 (10 VCI bits currently active) MAXVC-NUMBER : 16384 (0 VCCs currently configured) ILMI-VPI : 0 ILMI-VCI : 16 ILMI VERSION : VERSION 4.0 ILMI ADMIN STATUS : DISABLE ILMI OPER STATUS : DOWN NEIGHBOR-IP-ADDRESS : 0.0.0.0 NEIGHBOR-NAME : LINK ADMIN STATUS : DISABLE LINK OPER STATUS : UP
ATM MAXVCI-BITS <number>	IPNOS : CONFIG INTERFACE ATM(1/1)# **atm maxvci-bits 6** IPNOS : CONFIG INTERFACE ATM(1/1)# **display atm** ATM PARAMETERS IF-NAME : MAXVPI-BITS : 0 (0 VPI bits currently active) MAXVP-NUMBER : 256 (0 VPCs currently configured) MAXVCI-BITS : 6 (10 VCI bits currently active) MAXVC-NUMBER : 4096 (0 VCCs currently configured) ILMI-VPI : 0 ILMI-VCI : 16 ILMI VERSION : VERSION 4.0 ILMI ADMIN STATUS : DISABLE ILMI OPER STATUS : DOWN NEIGHBOR-IP-ADDRESS : 0.0.0.0 NEIGHBOR-NAME : LINK ADMIN STATUS : DISABLE LINK OPER STATUS : UP

Syntax	*Example*
[NO] ATM MAXVCI-BITS	IPNOS : CONFIG INTERFACE ATM(1/1)# **no atm maxvci-bits** IPNOS : CONFIG INTERFACE ATM(1/1)# **display atm** ATM PARAMETERS IF-NAME : MAXVPI-BITS : 0 (0 VPI bits currently active) MAXVP-NUMBER : 256 (0 VPCs currently configured) MAXVCI-BITS : 12 (10 VCI bits currently active) MAXVC-NUMBER : 4096 (0 VCCs currently configured) ILMI-VPI : 0 ILMI-VCI : 16 ILMI VERSION : VERSION 4.0 ILMI ADMIN STATUS : DISABLE ILMI OPER STATUS : DOWN NEIGHBOR-IP-ADDRESS : 0.0.0.0 NEIGHBOR-NAME : LINK ADMIN STATUS : DISABLE LINK OPER STATUS : UP
ATM MAXVP-NUMBER <number>	IPNOS : CONFIG INTERFACE ATM(1/1)# **atm maxvp-number 256** IPNOS : CONFIG INTERFACE ATM(1/1)# **display atm** ATM PARAMETERS IF-NAME : MAXVPI-BITS : 0 (0 VPI bits currently active) MAXVP-NUMBER : 256 (0 VPCs currently configured) MAXVCI-BITS : 12 (10 VCI bits currently active) MAXVC-NUMBER : 4096 (0 VCCs currently configured) ILMI-VPI : 0 ILMI-VCI : 16 ILMI VERSION : VERSION 4.0 ILMI ADMIN STATUS : DISABLE ILMI OPER STATUS : DOWN NEIGHBOR-IP-ADDRESS : 0.0.0.0 NEIGHBOR-NAME :

Syntax	Example
[NO] ATM MAXVP-NUMBER	IPNOS : CONFIG INTERFACE ATM(1/1)# **no atm maxvp-number** IPNOS : CONFIG INTERFACE ATM(1/1)# **display atm** ATM PARAMETERS IF-NAME : MAXVPI-BITS : 0 (0 VPI bits currently active) MAXVP-NUMBER : 0 (0 VPCs currently configured) MAXVCI-BITS : 12 (10 VCI bits currently active) MAXVC-NUMBER : 4096 (0 VCCs currently configured) ILMI-VPI : 0 ILMI-VCI : 16 ILMI VERSION : VERSION 4.0 ILMI ADMIN STATUS : DISABLE ILMI OPER STATUS : DOWN NEIGHBOR-IP-ADDRESS : 0.0.0.0 NEIGHBOR-NAME : LINK ADMIN STATUS : DISABLE LINK OPER STATUS : UP
ATM MAXVPI-BITS <number>	IPNOS : CONFIG INTERFACE ATM(1/1)# **atm maxvpi-bits 1** IPNOS : CONFIG INTERFACE ATM(1/1)# **display atm** ATM PARAMETERS IF-NAME : MAXVPI-BITS : 1 (0 VPI bits currently active) MAXVP-NUMBER : 0 (0 VPCs currently configured) MAXVCI-BITS : 12 (10 VCI bits currently active) MAXVC-NUMBER : 4096 (0 VCCs currently configured) ILMI-VPI : 0 ILMI-VCI : 16 ILMI VERSION : VERSION 4.0 ILMI ADMIN STATUS : DISABLE ILMI OPER STATUS : DOWN NEIGHBOR-IP-ADDRESS : 0.0.0.0 NEIGHBOR-NAME : LINK ADMIN STATUS : DISABLE LINK OPER STATUS : UP

Syntax	*Example*
[NO] ATM MAXVPI-BITS	IPNOS : CONFIG INTERFACE ATM(1/1)# **no atm maxvpi-bits** IPNOS : CONFIG INTERFACE ATM(1/1)# **display atm** ATM PARAMETERS IF-NAME : MAXVPI-BITS : 0 (0 VPI bits currently active) MAXVP-NUMBER : 0 (0 VPCs currently configured) MAXVCI-BITS : 12 (10 VCI bits currently active) MAXVC-NUMBER : 4096 (0 VCCs currently configured) ILMI-VPI : 0 ILMI-VCI : 16 ILMI VERSION : VERSION 4.0 ILMI ADMIN STATUS : DISABLE ILMI OPER STATUS : DOWN NEIGHBOR-IP-ADDRESS : 0.0.0.0 NEIGHBOR-NAME : LINK ADMIN STATUS : DISABLE LINK OPER STATUS : UP
ATM NEIGHBOR-IP-ADDRESS <ip-address>	IPNOS : CONFIG INTERFACE ATM(1/1)# **atm neighbor-ip-address 192.168.142.96** IPNOS : CONFIG INTERFACE ATM(1/1)# **display atm** ATM PARAMETERS IF-NAME : MAXVPI-BITS : 0 (0 VPI bits currently active) MAXVP-NUMBER : 0 (0 VPCs currently configured) MAXVCI-BITS : 12 (10 VCI bits currently active) MAXVC-NUMBER : 4096 (0 VCCs currently configured) ILMI-VPI : 0 ILMI-VCI : 16 ILMI VERSION : VERSION 4.0 ILMI ADMIN STATUS : DISABLE ILMI OPER STATUS : DOWN NEIGHBOR-IP-ADDRESS : 192.168.142.96 NEIGHBOR-NAME : LINK ADMIN STATUS : DISABLE LINK OPER STATUS : UP

Syntax	Example
[NO] ATM NEIGHBOR-IP-ADDRESS	IPNOS : CONFIG INTERFACE ATM(1/1)# **no atm neighbor-ip-address** IPNOS : CONFIG INTERFACE ATM(1/1)# **display atm** ATM PARAMETERS IF-NAME : MAXVPI-BITS : 0 (0 VPI bits currently active) MAXVP-NUMBER : 0 (0 VPCs currently configured) MAXVCI-BITS : 12 (10 VCI bits currently active) MAXVC-NUMBER : 4096 (0 VCCs currently configured) ILMI-VPI : 0 ILMI-VCI : 16 ILMI VERSION : VERSION 4.0 ILMI ADMIN STATUS : DISABLE ILMI OPER STATUS : DOWN NEIGHBOR-IP-ADDRESS : 0.0.0.0 NEIGHBOR-NAME : LINK ADMIN STATUS : DISABLE LINK OPER STATUS : UP
ATM NEIGHBOR-NAME <string>	IPNOS : CONFIG INTERFACE ATM(1/1)# **atm neighbor-name acmenet1** IPNOS : CONFIG INTERFACE ATM(1/1)# **display atm** ATM PARAMETERS IF-NAME : MAXVPI-BITS : 0 (0 VPI bits currently active) MAXVP-NUMBER : 0 (0 VPCs currently configured) MAXVCI-BITS : 12 (10 VCI bits currently active) MAXVC-NUMBER : 4096 (0 VCCs currently configured) ILMI-VPI : 0 ILMI-VCI : 16 ILMI VERSION : VERSION 4.0 ILMI ADMIN STATUS : DISABLE ILMI OPER STATUS : DOWN NEIGHBOR-IP-ADDRESS : 0.0.0.0 NEIGHBOR-NAME : acmenet1 LINK ADMIN STATUS : DISABLE LINK OPER STATUS : UP

Syntax	*Example*
[NO] ATM NEIGHBOR-NAME	IPNOS : CONFIG INTERFACE ATM(1/1)# **no atm neighbor-name** IPNOS : CONFIG INTERFACE ATM(1/1)# **display atm** ATM PARAMETERS IF-NAME : MAXVPI-BITS : 0 (0 VPI bits currently active) MAXVP-NUMBER : 0 (0 VPCs currently configured) MAXVCI-BITS : 12 (10 VCI bits currently active) MAXVC-NUMBER : 4096 (0 VCCs currently configured) ILMI-VPI : 0 ILMI-VCI : 16 ILMI VERSION : VERSION 4.0 ILMI ADMIN STATUS : DISABLE ILMI OPER STATUS : DOWN NEIGHBOR-IP-ADDRESS : 0.0.0.0 NEIGHBOR-NAME : LINK ADMIN STATUS : DISABLE LINK OPER STATUS : UP

Default Value

None

Command Level

atm interface

Related Command

None

display

View the current ATM interface and OC-3c interface configuration. See "oc3" on page 9-2 for OC-3 **display** information. Table 10-2 describes performance monitoring parameters using the **display stats sum** command.

Table 10-2 Performance Monitoring Parameters

Parameter	*Description*
CV_s	Section Code Violation
ES_s	Section Errored Seconds
SES_s	Section Severely Errored Seconds
SEFS_s	Section Severely Framed Errored Seconds
CV_l	Line Code Violations
ES_l	Line Errored Seconds
SES_l	Line Severely Errored Seconds
UAS_l	Line Unavailable Seconds
FC_l	Line Failure Counts
CV_p	Path Code Violations
ES_p	Path Errored Seconds
SES_p	Path Severely Errored Seconds
UAS_p	Path Unavailable Seconds
FC_p	Path Failure Counts

Syntax	*Example*

ATM Interface:
DISPLAY
	`IPNOS : CONFIG INTERFACE ATM(1/1)# display`
	`SLOT : 1`
	`PORT : 1`
	`ATM PARAMETERS`
	`IF-NAME :`
	`MAXVPI-BITS : 0 (0 VPI bits currently active)`
	`MAXVP-NUMBER : 0 (0 VPCs currently configured)`
	`MAXVCI-BITS : 12 (12 VCI bits currently active)`
	`MAXVC-NUMBER : 4096 (0 VCCs currently configured)`
	`ILMI-VPI : 0`
	`ILMI-VCI : 16`
	`ILMI VERSION : VERSION 4.0`
	`ILMI ADMIN STATUS : DISABLE`
	`ILMI OPER STATUS : DOWN`
	`NEIGHBOR-IP-ADDRESS : 0.0.0.0`
	`NEIGHBOR-NAME :`
	`LINK ADMIN STATUS : DISABLE`
	`LINK OPER STATUS : UP`
	
	`OC-3 PARAMETERS`
	`FRAMING : SONET`
	`CLOCK : INTERNAL`
	`J0 : 0xCC`
	`SCRAMBLE : ENABLE`
	`ALARM : ENABLE`
	`PATH-TRACE :`
	`REMOTE-PATH-TRACE :`
	`LOOP : NONE`
	`ADMIN STATUS : ENABLE`
	`OPER STATUS : UP -`
	
	`VALID INTERVALS : 12`
	`INVALID INTERVALS : 0`
	`ELAPSED TIME(sec) : 615`

Syntax	Example
ATM interface: DISPLAY <argument>	IPNOS : CONFIG INTERFACE ATM(2/1)# **display atm** ATM PARAMETERS IF-NAME : atm1 MAXVPI-BITS : 10 (0 VPI bits currently active) MAXVP-NUMBER : 1020 (0 VPCs currently configured) MAXVCI-BITS : 9 (0 VCI bits currently active) MAXVC-NUMBER : 500 (0 VCCs currently configured) ILMI-VPI : 0 ILMI-VCI : 16 ILMI VERSION : VERSION 4.0 UNI TYPE : PRIVATE UNI VERSION : VERSION 4.0 NEIGHBOR-IP-ADDRESS : 192.168.142.96 NEIGHBOR-NAME : fred1 ADMIN STATUS : ENABLE OPER STATUS : DOWN
OC3 interface: DISPLAY OC3	IPNOS : CONFIG INTERFACE POS(4/1)# **display oc3** ENCAPSULATION : NONE OC-3 PARAMETERS FRAMING : SONET CLOCK : INTERNAL J0 : 0xCC SCRAMBLE : ENABLE ALARM : ENABLE PATH-TRACE : REMOTE-PATH-TRACE : LOOP : NONE ADMIN STATUS : ENABLE OPER STATUS : UP - VALID INTERVALS : 12 INVALID INTERVALS : 0 ELAPSED TIME(sec) : 615

Syntax	*Example*

Display current OC-3 interval stats on OC-3c ATM blade:
`DISPLAY STATS [0 | 1..32 | SUM]`
	`IPNOS : CONFIG INTERFACE ATM(1/1)#` **`display stats 0`**
	
	`OC-3 STATISTICS`
	`Current Interval - elapsed time(min:sec) 10:3`
	
	`Param. Near End`
	`Section:`
	`CV_s 0`
	`ES_s 595`
	`SES_s 595`
	`SEFS_s 595`
	
	`Line:`
	`CV_l 0`
	`ES_l 0`
	`SES_l 0`
	`UAS_l 595`
	`FC_l 0`
	
	`Path:`
	`CV_p 0`
	`ES_p 0`
	`SES_p 0`
	`UAS_p 595`
	`FC_p 0`

Syntax	Example
Display previous OC-3 interval stats on OC-3c ATM blade: DISPLAY STATS [0 | 1..32 | SUM]	IPNOS : CONFIG INTERFACE ATM(1/1)# **display stats 12** OC-3 STATISTICS Valid Interval #12 into the past Param. Near End Section: CV_s 0 ES_s 892 SES_s 892 SEFS_s 892 Line: CV_l 0 ES_l 0 SES_l 0 UAS_l 892 FC_l 0 Path: CV_p 0 ES_p 0 SES_p 0 UAS_p 892 FC_p 0
Display OC-3 interval summary on OC-3c ATM blade: DISPLAY STATS [0 | 1..32 | SUM]	IPNOS : CONFIG INTERFACE ATM(1/1)# **display stats sum** OC-3 STATISTICS Param. Cur. Intv Prev. Intv Cur. Day Prev. Day Section: CV_s 0 0 0 0 ES_s 658 892 39920 85663 SES_s 658 892 39920 85663 SEFS_s 658 892 39920 85663 Line: CV_l 0 0 0 0 ES_l 0 0 0 0 SES_l 0 0 0 0 UAS_l 658 892 39920 85663 FC_l 0 0 0 0 Path: CV_p 0 0 0 0 ES_p 0 0 0 0 SES_p 0 0 0 0 UAS_p 658 892 39920 85663 FC_p 0 0 0 0

Default Value

None

Command Level

atm and pos interfaces

Related Commands

None

atm

Go to the ATM subinterface command level.

NOTE: Disable the ATM subinterface before changing any parameters. The exceptions are **ARP**, **ARP-age, ip-addr**, and **OAM-loopback** commands.

The following commands are available.

- "arp" on page 10-24
- "arp-age" on page 10-28
- "arp-max-entries" on page 10-29
- "clear" on page 10-30
- "display" on page 10-32
- "enable" on page 10-33
- "encapsulation" on page 10-34
- "icr" on page 10-36
- "ip-addr" on page 10-37
- "mbs" on page 10-38
- "mcr" on page 10-40
- "oam-loopback" on page 10-41
- "pcr" on page 10-41
- "pppoe" on page 10-42
- "protocol" on page 10-50
- "receive-size" on page 10-54
- "scr" on page 10-55
- "service-category" on page 10-56
- "transmit-size" on page 10-65
- "vci" on page 10-66
- "vpi" on page 10-69

Syntax	*Example*

Create/configure ATM subinterface:
`INTERFACE ATM <slot>/<port>:<channel>:<subinterface>`
	`IPNOS : CONFIG#` **`int atm 1/1:0:40`**
	
	`IPNOS : CONFIG INTERFACE ATM(1/1:0:40)#` **`enable`**
	
	`IPNOS : CONFIG INTERFACE ATM(1/1:0:40)#` **`comm`**
	`Commit configuration`
	`INTERFACE ATM(1/1:0:40): Object successfully set`
	
	`IPNOS : CONFIG#`

Delete ATM subinterface:
`NO INTERFACE ATM <slot>/<port>:<channel>:<subinterface>`
	`IPNOS : CONFIG#` **`no interface atm 1/1:0:40`**
	`ATM (1/1:0:40): Object successfully deleted`
	
	`IPNOS : CONFIG#`

Default Value

None

Command Level

interface

Related Commands

oc3 interface
atm interface

arp

Create an entry in the virtual channel ARP cache. For static ARP, you must assign an IP address to a unique hardware Media Access Control (MAC) address. Dynamic ARP is generated automatically by the ATM bridge. You cannot create a dynamic ARP entry, only delete it.

NOTE: Modifying the ARP cache is only effective when bridging encapsulation is set on the subinterface. See “encapsulation” on page 10-34.

Syntax	*Example*

Create a static ARP entry:
`ARP STATIC <ip-address> <hw-address>`
	`IPNOS: CONFIG INTERFACE ATM(2/1:0:100)# arp static 10.20.20.20 23:00:88:88:99:00`
	`INTERFACE ATM(2/1:0:100): Object successfully set`
	`STATIC ARP [2/1:0:100] : 10.20.20.20 Object successfully set`
	
	`IPNOS: CONFIG INTERFACE ATM(2/1:0:100)# display arp`
	`ARP General Information::`
	`MAC-ADDR : 00:d0:7d:08:00:64`
	`ARP-AGE : 12`
	`ARP-MAX-ENTRIES : 20`
	
	`IP Address(es):`
	`IPv4 Addresses | Type`
	`0.0.0.0 PROXY`
	`10.10.10.10 PROXY`
	
	`STATIC ARP(s):`
	`IP Address | HW Address`
	`10.20.20.20 23:00:88:88:99:00`
	
	`DYNAMIC ARP(s):`
	`IP Address | HW Address`
	`50.50.50.50 12:23:34:05:11:89`

Syntax	*Example*
Delete a static ARP entry: NO ARP STATIC <ip-address>	IPNOS: CONFIG INTERFACE ATM(2/1:0:100)# **no arp static 10.20.20.20** STATIC ARP [2/1:0:100] : 10.20.20.20 Object successfully deleted IPNOS: CONFIG INTERFACE ATM(2/1:0:100)# **display arp** ARP General Information:: MAC-ADDR : 00:d0:7d:08:00:64 ARP-AGE : 12 ARP-MAX-ENTRIES : 20 IP Address(es): IPv4 Addresses | Type 0.0.0.0 PROXY 10.10.10.10 PROXY STATIC ARP(s): List is empty DYNAMIC ARP(s): IP Address | HW Address 50.50.50.50 12:23:34:05:11:89

Syntax	*Example*

Delete a dynamic ARP entry:
`NO ARP DYNAMIC <ip-address>`
	`IPNOS: CONFIG INTERFACE ATM(2/1:0:100)# no arp dynamic 50.50.50.50`
	`DYNAMIC ARP [2/1:0:100] : 50.50.50.50 Object successfully deleted`
	
	`IPNOS: CONFIG INTERFACE ATM(2/1:0:100)# display arp`
	`ARP General Information::`
	`MAC-ADDR : 00:d0:7d:08:00:64`
	`ARP-AGE : 12`
	`ARP-MAX-ENTRIES : 20`
	
	`IP Address(es):`
	`IPv4 Addresses | Type`
	`0.0.0.0 PROXY`
	`10.10.10.10 PROXY`
	
	`STATIC ARP(s):`
	`List is empty`
	
	`DYNAMIC ARP(s):`
	`List is empty`

Default Value

None

Command Level

atm subinterface

Related Commands

clear
ATM subinterface encapsulation

arp-age

Set the aging time for an entry in the virtual channel Address Resolution Protocol (ARP) cache. Choose from 6 to 60 minutes.

NOTE: Modifying the ARP age is only effective when bridging encapsulation is set on the subinterface. See "encapsulation" on page 10-34.

Syntax	*Example*
ARP-AGE <number>	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **arp-age 30** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display arp-age** ARP-AGE : 30
NO ARP-AGE	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# no **arp-age** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display arp-age** ARP-AGE : 15

Default Value

15 minutes

Command Level

atm subinterface

Related Commands

encapsulation

arp-max-entries

Set the maximum size of the virtual channel ARP cache. Choose from 8 to 64 entries.

NOTE: You cannot change this value when the admin status is enabled and protocol is set to IP.

Modifying the ARP cache size is only effective when bridging encapsulation is set on the subinterface. See "encapsulation" on page 10-34.

Syntax	*Example*
ARP-MAX-ENTRIES <number>	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **arp-max-entries 50** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display arp-max-entries** ARP-MAX-ENTRIES : 50
NO ARP-MAX-ENTRIES	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **no arp-max-entries** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display arp-max-entries** ARP-MAX-ENTRIES : 8

Default Value

8 entries

Command Level

atm subinterface

Related Commands

encapsulation

clear

Clear the ARP cache. Choose a single cache or clear the entire cache list.

NOTE: Clearing the ARP cache is only effective when bridging encapsulation is set on the subinterface. See "encapsulation" on page 10-34.

Syntax	*Example*

Delete a static ARP entry:
`CLEAR ARP STATIC [ip-address]`
	`IPNOS: CONFIG INTERFACE ATM(2/1:0:100)#` **`clear arp static 10.20.20.20`**
	`STATIC ARP [2/1:0:100] : 10.20.20.20 Object successfully deleted`
	
	`IPNOS: CONFIG INTERFACE ATM(2/1:0:100)#` **`display arp`**
	`ARP General Information::`
	`MAC-ADDR : 00:d0:7d:08:00:64`
	`ARP-AGE : 12`
	`ARP-MAX-ENTRIES : 20`
	
	`IP Address(es):`
	`IPv4 Addresses | Type`
	`0.0.0.0 PROXY`
	`10.10.10.10 PROXY`
	
	`STATIC ARP(s):`
	`List is empty`
	
	`DYNAMIC ARP(s):`
	`IP Address | HW Address`
	`50.50.50.50 12:23:34:05:11:89`

Syntax	*Example*

Clear entire dynamic ARP cache:
`CLEAR ARP DYNAMIC [ip-address]`
	`IPNOS: CONFIG INTERFACE ATM(2/1:0:100)#` **`clear arp dynamic`**
	`DYNAMIC ARP [2/1:0:100] : 50.50.50.50 Object successfully deleted`
	
	`IPNOS: CONFIG INTERFACE ATM(2/1:0:100)#` **`display arp`**
	`ARP General Information::`
	`MAC-ADDR : 00:d0:7d:08:00:64`
	`ARP-AGE : 12`
	`ARP-MAX-ENTRIES : 20`
	
	`IP Address(es):`
	`IPv4 Addresses | Type`
	`0.0.0.0 PROXY`
	`10.10.10.10 PROXY`
	
	`STATIC ARP(s):`
	`List is empty`
	
	`DYNAMIC ARP(s):`
	`List is empty`

Default Value

None

Command Level

atm subinterface

Related Commands

arp
encapsulation

display

View the current ATM subinterface.

Syntax	*Example*

DISPLAY	`IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# display`
	`SLOT : 1`
	`PORT : 1`
	`CHANNEL : 0`
	`SUBIF : 40`
	`VPI : 0`
	`VCI : 0`
	`SERVICE-CATEGORY: UBR`
	
	`CONNECTION-KIND : PVC`
	`PVC-TYPE : PVCC`
	`PROTOCOL : NONE`
	`PCR : 175`
	`TRANSMIT SDU SIZE : 1792`
	`RECEIVE SDU SIZE : 1792`
	`OAM-LOOPBACK : DISABLE`
	`ENCAPSULATION : LLC-ROUTED-ENCAPSULATION`
	`ADMIN-STATUS : DISABLE`
	`OPER-STATUS : DOWN`
	
	`ARP General Information:`
	`MAC-ADDR : 02:02:00:00:00:64`
	`ARP-AGE : 15`
	`ARP-MAX-ENTRIES : 8`
	
	`IP Address(es):`
	`IPv4 Addresses | Type`
	`0.0.0.0 PROXY`
	`10.10.10.10 PROXY`
	
	`STATIC ARP(s):`
	`List is empty`
	
	`DYNAMIC ARP(s):`
	`List is empty`

Default Value

None

Command Level

atm subinterface

Related Commands

show

enable

Enable the subinterface. Disable the subinterface before changing any parameters.

Syntax	*Example*
[NO] ENABLE	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **enable** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display admin-status** ADMIN-STATUS : ENABLE OPER-STATUS : DOWN
[NO] ENABLE	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **no enable** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display admin-status** ADMIN-STATUS : DISABLE OPER-STATUS : DOWN

Default Value

disabled

Command Level

atm subinterface

Related Commands

enable (ATM interface)

encapsulation

Encapsulate the subinterface for ISO or non-ISO routed or bridged protocols. See RFC 1483 and RFC 2684 for details. Table 10-3 describes the parameters.

Table 10-3 ATM subinterface Encapsulation Parameters

Parameter	*Description*
LLC-routed-encapsulation	Allow multiplexing of multiple protocols over a single ATM virtual connection (VC). The protocol type of each ATM cell is identified by a Logical Link Control (LLC) header. Two types are supported: • Unacknowledged connectionless mode cells get an 8-byte header. • Connection-mode cells get a 4-byte header.
LLC-bridged-encapsulation	Same as LLC-routed-encapsulation, but provides egress and ingress to and from an IP over ATM bridge. IPNOS provides services over the bridge, but is not a bridge in itself. A hardware MAC header is added to outgoing packets from an ARP cache. Refer to “arp” on page 10-24. A proxy IP address for the destination ATM PVC can be configured if you are creating a proxy ARP. Refer to “ip-addr” on page 10-37 for more information.
VC-mux-routed-encapsulation	Each ATM VC carries ATM cells of exactly one protocol type. When multiple protocols need to be transported, there is a separate VC for each.
VC-mux-bridged-8023-encapsulation	Same as VC-mux-routed-encapsulation, but provides egress and ingress to and from an IP over ATM bridge. IPNOS provides services over the bridge, but is not a bridge in itself. A hardware MAC header is added to outgoing packets from an ARP cache. Refer to “arp” on page 10-24. A proxy IP address for the destination ATM PVC can be configured if you are creating a proxy ARP. Refer to “ip-addr” on page 10-37 for more information.

NOTE: IPNOS IP over ATM bridging currently only supports 802.3 Ethernet framing format. For ingress, IPNOS supports protocol IDs 0x00-01 and 0x00-7. For egress, IPNOS supports protocol ID 0x00-7. Other bridge framing formats, such as Token Ring and FDDI, are currently not supported.

Table 10-4 describes the compatibility between ATM encapsulation and ATM protocols.

Table 10-4 ATM Encapsulation, ATM Protocol Compatibility

ATM Protocol	*ATM Encapsulation*			
	LLC-Routed	*LLC-Bridged*	*VC-Multiplex-Routed*	*VC-Multiplex-Bridged-802.3*
None	Yes	Yes	Yes	Yes
PPP	Yes	No	Yes	No
PPPoE	No	Yes	No	Yes
IP	Yes	Yes	Yes	Yes

Syntax	*Example*
ENCAPSULATION {LLC-ROUTED-ENCAPSULATION | LLC-BRIDGED-ENCAPSULATION | VC-MUX-ROUTED-ENCAPSULATION | VC-MUX-BRIDGED-8023-ENCAPSULATION}	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **encapsulation llc-routed-encapsulation** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display encapsulation** ENCAPSULATION : LLC-ROUTED-ENCAPSULATION
ENCAPSULATION {LLC-ROUTED- ENCAPSULATION | LLC-BRIDGED-ENCAPSULATION | VC-MUX-ROUTED-ENCAPSULATION | VC-MUX-BRIDGED-8023-ENCAPSULATION}	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **encapsulation llc-bridged-encapsulation** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display encapsulation** ENCAPSULATION : LLC-BRIDGED-ENCAPSULATION
ENCAPSULATION {LLC-ROUTED- ENCAPSULATION | LLC-BRIDGED-ENCAPSULATION | VC-MUX-ROUTED-ENCAPSULATION | VC-MUX-BRIDGED-8023-ENCAPSULATION}	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **encapsulation vc-mux-routed-encapsulation** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display encapsulation** ENCAPSULATION : VC-MUX-ROUTED-ENCAPSULATION

Syntax	*Example*
ENCAPSULATION {LLC-ROUTED- ENCAPSULATION | LLC-BRIDGED-ENCAPSULATION | VC-MUX-ROUTED-ENCAPSULATION | VC-MUX-BRIDGED-8023-ENCAPSULATION}	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **encapsulation vc-mux-bridged-8023-encapsulation** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display encapsulation** ENCAPSULATION : VC-MUX-BRIDGED-ENCAPSULATION
NO ENCAPSULATION	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **no encapsulation** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display encapsulation** ENCAPSULATION : LLC-ROUTED-ENCAPSULATION

Default Value

LLC-routed-encapsulation

Command Level

atm subinterface

Related Commands

arp-age
arp
arp-max-entries
ip-addr
protocol

icr

Set the initial cell rate (ICR). The ICR is the rate, in cells per second, at which the VR should send cells initially and after an idle period. Choose from 1 cell per second to the peak cell rate (PCR). Use this command when the Quality of Service is set to available bit rate (ABR). See “service-category” on page 10-56.

Syntax	Example
ICR <number>	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **icr 25** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display icr** ICR : 25
NO ICR	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **no icr** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display icr** ICR : 175

Default Value

175 cells per second

Command Level

atm subinterface

Related Commands

mcr
pcr
service-category

ip-addr

Create a proxy-ARP IP address. This command is optional for IP over ATM bridging. Non-proxy ARP IP addresses may be generated by the connection. For true proxy ARP functionality, use 0.0.0.0.

NOTE: Modfying the proxy-ARP address is only effective when bridging encapsulation is set on the subinterface. See “encapsulation” on page 10-34.

Syntax	*Example*
IP-ADDR <ip-address>	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **ip-addr 10.10.10.10** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display ip** IP Address(es): IPv4 Addresses | Type 0.0.0.0 PROXY 10.10.10.10 PROXY
NO IP-ADDR <ip-address>	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **no ip-addr 10.10.10.10** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display ip** IP Address(es): IPv4 Addresses | Type 0.0.0.0 PROXY

Default Value

None

Command Level

atm subinterface

Related Commands

encapsulation

mbs

Set the maximum burst size (MBS). The MBS is the maximum burst transfer limit at the agreed-upon peak rate. The value is cells per second. Use this command when the Quality of Service is set to real-time variable bit rate (RT-VBR) or non-real-time variable bit rate (NRT-VBR). See “service-category” on page 10-56.

Syntax	Example
MBS <number>	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **mbs 100** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display mbs** MBS : 100
NO MBS	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **no mbs** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display mbs** MBS : 175

Default Value

175 cells per second

Command Level

atm subinterface

Related Commands

pcr
service-category

mcr

Set the minimum cell rate (MCR). The MCR is the minimum transmission rate for the PVC. The value is cells per second. Use this command when the Quality of Service is set to available bit rate (ABR). See "service-category" on page 10-56.

Syntax	*Example*
MCR <number>	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **mcr 25** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display mcr** MCR : 25
NO MCR	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **no mcr** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display mcr** MCR : 175

Default Value

175 cells per second

Command Level

atm subinterface

Related Commands

icr
pcr
service-category

oam-loopback

Enable the operations administration and maintenance (OAM) loopback. When enabled, **oam-loopback** pings the network from end to end four times.

Syntax	*Example*
OAM-LOOPBACK	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **oam-loopback** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display oam** OAM-LOOPBACK : ENABLE
[NO] OAM-LOOPBACK	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **no oam-loopback** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display oam** OAM-LOOPBACK : DISABLE

Default Value

disabled

Command Level

atm subinterface

Related Commands

None

pcr

Set the peak cell rate (PCR). The PCR is the highest transfer rate allowable for the customer connection. The value is cells per second. Choose from 1 to 353,257 cells per second, or the maximum blade wire speed. Use this command with all Quality of Service settings. See “service-category” on page 10-56.

Syntax	*Example*
PCR <number>	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **pcr 200** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display pcr** PCR : 200

Syntax	Example
NO PCR	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **no pcr** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display pcr** PCR : 175

Default Value

175 cells per second

Command Level

atm subinterface

Related Commands

mcr
scr
service-category

pppoe

Go to the PPP over Ethernet (PPPoE) command level. Use this feature to enable digital subscriber line (DSL) services over the network via RFC 1483 Ethernet bridged frames, and terminate at the IPSX switch or a remote broadband server. See RFC 2516.

Before running this command, set the ATM subinterface **encapsulation** to a bridged format, and set the **protocol** to PPPoE. Refer to "encapsulation" on page 10-34 and "protocol" on page 10-50. The following commands are available at this level.

- "display" on page 10-43
- "pppoe" on page 10-45
- "session" on page 10-48

Syntax	Example
PPPOE	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)#**protocol pppoe** IPNOS : CONFIG INTERFACE ATM(1/1:0:40) PPPoE#

Default Value

None

Command Level

atm subinterface

Related Commands

encapsulation
protocol

display

View the current configuration of the PPPoE access concentrator.

Syntax	*Example*
DISPLAY	

```
IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# display
SLOT                     : 2
PORT                     : 1
CHANNEL                  : 0
SUBIF                    : 85

PPPOE ACCESS CONCENTRATOR PARAMETERS
NAME                     :
MAX-TOTAL-SESSIONS       : 8
MAX-SESSIONS-PER-HOST    : 2
CURRENT-TOTAL-SESSIONS   : 0
CURRENT-ACTIVE-SESSIONS  : 0
PORT-MAC-ADDRESS         : 01:00:00:00:00:55
SERVICE-ADVERTISING      : ENABLE
PPP-AUTHENTICATION       : CHAPMD5
SERVICE-TEMPLATE         :
ADMIN-STATUS             : DISABLE
OPER-STATUS              : DOWN

SESSION(s):
ID        | USERNAME        | SERVICE-NAME    | UP-TIME
152       | customer1       | service1        | 136
176       | customer2       | service2        | 89
203       | customer3       | service3        | 94

STATISTICS:
PADI Received            : 0
PADI Rejected            : 0
PADR Received            : 0
PADR Rejected            : 0
PADO Transmitted         : 0
PADS Transmitted         : 0
PADT Received            : 0
PADT Transmitted         : 0
AC System Errors         : 0
Malformed Packets        : 0
Service Name Errors      : 0
Generic Error Received   : 0
Generic Error Transmitted: 0
```

Default Value

None

Command Level

PPPoE access concentrator

Related Commands

None

pppoe

Configure the PPPoE access concentrator. Table 10-5 describes the parameters.

Table 10-5 PPPoE Parameters

Parameter	*Description*
Enable	Enable or disable the PPPoE access concentrator. Use the **no** option to reset the command to its default value. The default value is disabled.
Max-sessions-per-host	Set the maximum number of PPP sessions per host. Choose from 1 to 8. Use the **no** option to reset the command to its default value. The default value is 2 sessions.
Max-total-sessions	Set the total number of PPP sessions for the access concentrator. Choose from 1 to 64. Use the **no** option to reset the command to its default value. The default value is 8 sessions.
Name	Name of the access concentrator.
PPP-authentication	Set the PPP authentication type. Choose from none, PAP, CHAPMD5, or MSCHAPV1. Use the **no** option to reset the command to its default value. The default value is CHAPMD5.
Service-advertising	Enable advertising of service profiles in the PPPoE Active Discovery Offer (PADO) packets. Use the **no** option to reset the command to its default value. The default value is disabled.
Service-template	Set the name of the service template. This is an optional feature to give users multiple ISP options, and have their PPP packets forwarded to one of a selection of LACs and LNSs. See “service-selection-template” on page 16-136. Use the **no** option to reset the command to its default value. The default value is empty.

Syntax	*Example*
PPPOE ENABLE	IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **pppoe enable** IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **display admin** ADMIN-STATUS : ENABLE
NO PPPOE ENABLE	IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **no pppoe enable** IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **display admin** ADMIN-STATUS : DISABLE
PPPOE MAX-SESSIONS-PER-HOST <number>	IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **pppoe max-sessions-per-host 5** IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **display max-sessions-per-host** MAX-SESSIONS-PER-HOST : 5
NO PPPOE MAX-SESSIONS-PER-HOST	IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **no pppoe max-sessions-per-host** IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **display max-sessions-per-host** MAX-SESSIONS-PER-HOST : 2
PPPOE MAX-TOTAL-SESSIONS <number>	IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **pppoe max-total-sessions 20** IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **display max-total-sessions** MAX-TOTAL-SESSIONS : 20
NO PPPOE MAX-TOTAL-SESSIONS	IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **no pppoe max-total-sessions** IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **display max-total-sessions** MAX-TOTAL-SESSIONS : 8
PPPOE NAME <string>	IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **pppoe name belmont_ac** IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **display name** NAME : belmont_ac

Syntax	*Example*
NO PPPOE NAME	IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **no pppoe name** IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **display name** NAME :
PPPOE PPP-AUTHENTICATION {CHAPMD5 | MSCHAPV1 | PAP | NONE}	IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **pppoe ppp-authentication PAP** IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **display ppp-authentication** PPP-AUTHENTICATION : PAP
NO PPPOE PPP-AUTHENTICATION	IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **no pppoe ppp-authentication** IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **display ppp-authentication** PPP-AUTHENTICATION : CHAPMD5
PPPOE SERVICE-ADVERTISING	IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **pppoe service-advertising** IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **display service-advertising** SERVICE-ADVERTISING : ENABLE
NO PPPOE SERVICE-ADVERTISING	IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **no pppoe service-advertising** IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **display service-advertising** SERVICE-ADVERTISING : DISABLE
PPPOE SERVICE-TEMPLATE <name>	IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **pppoe service-template template_1** IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **display service-template** SERVICE-TEMPLATE : template_1
NO PPPOE SERVICE-TEMPLATE	IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **no pppoe service-template** IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **display service-template** SERVICE-TEMPLATE :

Default Value

None

Command Level

PPPoE access concentrator

Related Commands

None

session

Delete sessions based on their session ID or their username.

Syntax	*Example*
[NO] SESSION <session-id>	IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **display session** SESSION(s): ID | USERNAME | SERVICE-NAME | UP-TIME 152 | customer1 | service1 | 136 176 | customer2 | service2 | 89 203 | customer3 | service3 | 94 IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **session 152** IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **display session** SESSION(s): ID | USERNAME | SERVICE-NAME | UP-TIME 176 | customer2 | service2 | 89 203 | customer3 | service3 | 94
[NO] SESSION <user-name>	IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **display session** SESSION(s): ID | USERNAME | SERVICE-NAME | UP-TIME 176 | customer2 | service2 | 89 203 | customer3 | service3 | 94 IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **session customer2** IPSX: CONFIG INTERFACE ATM(2/1:0:40) PPPoE# **display session** SESSION(s): ID | USERNAME | SERVICE-NAME | UP-TIME 203 | customer3 | service3 | 94

Default Value

None

Command Level

PPPoE access concentrator

Related Commands

None

protocol

Enable ATM network protocol. Choose from IP, PPP, and PPP over Ethernet. Disable the subinterface before changing the protocol. If the current protocol is IP, then disable the VI connection to the subinterface before changing the protocol. Changing the protocol will automatically disconnect any PPP connections to the ATM subinterface. Use the **no** option to reset the command to its default value.

NOTE: The protocol is not automatically changed to IP when you run the **no protocol** command. You also need to attach a VI to the ATM subinterface to change the protocol to IP.

Syntax	*Example*
PROTOCOL {PPP | PPPOE}	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **no enable** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **protocol ppp** INTERFACE ATM(1/1:0:40): Object successfully set IPNOS : CONFIG INTERFACE ATM(1/1:0:40) PPP# **ppp enable** IPNOS : CONFIG INTERFACE ATM(1/1:0:40) PPP# **commit** Commit configuration PPP: Object successfully set IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **enable** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display** SLOT : 1 PORT : 1 CHANNEL : 0 SUBIF : 40 VPI : 0 VCI : 0 SERVICE-CATEGORY: UBR CONNECTION-KIND : PVC PVC-TYPE : PVCC PROTOCOL : PPP PCR : 175 TRANSMIT SDU SIZE : 1792 RECEIVE SDU SIZE : 1792 OAM-LOOPBACK : DISABLE ENCAPSULATION : LLC-ROUTED-ENCAPSULATION ADMIN-STATUS : ENABLE OPER-STATUS : DOWN ARP General Information: MAC-ADDR : 02:02:00:00:00:64 ARP-AGE : 15 ARP-MAX-ENTRIES : 32 IP Address(es): IPv4 Addresses | Type 0.0.0.0 PROXY 10.10.10.10 PROXY STATIC ARP(s): List is empty DYNAMIC ARP(s): List is empty

Syntax	*Example*
NO PROTOCOL	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **no enable** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **no protocol** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **commit** Commit configuration INTERFACE ATM(1/1:0:40): Object successfully set IPNOS : CONFIG# **interface atm 1/1:0:40** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display** SLOT : 1 PORT : 1 CHANNEL : 0 SUBIF : 40 VPI : 0 VCI : 0 SERVICE-CATEGORY: UBR CONNECTION-KIND : PVC PVC-TYPE : PVCC PROTOCOL : NONE PCR : 175 TRANSMIT SDU SIZE : 1792 RECEIVE SDU SIZE : 1792 OAM-LOOPBACK : DISABLE ENCAPSULATION : LCC-ROUTED-ENCAPSULATION ADMIN-STATUS : DISABLE OPER-STATUS : DOWN ARP General Information: MAC-ADDR : 02:02:00:00:00:64 ARP-AGE : 15 ARP-MAX-ENTRIES : 32 IP Address(es): IPv4 Addresses | Type 0.0.0.0 PROXY 10.10.10.10 PROXY STATIC ARP(s): List is empty DYNAMIC ARP(s): List is empty

Default Value

None

Command Level

atm subinterface

Related Commands

encapsulation

receive-size

Set the receive size of the service data units (SDUs). The range is from 0 to 9188 bytes, but currently up to 1792 bytes are supported.

Syntax	*Example*
RECEIVE-SIZE <number>	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **receive-size 512** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display receive-size** RECEIVE SDU SIZE : 512
NO RECEIVE-SIZE	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **no receive-size** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display receive-size** RECEIVE SDU SIZE : 1792

Default Value

1792 bytes

Command Level

atm subinterface

Related Commands

transmit-size

scr

Set the sustainable cell rate (SCR) in cells/sec. The SCR is the average rate of an ATM connection over time. Choose from 1 cell/sec to the peak cell rate (PCR). Use this command when the Quality of Service is set to real-time variable bit rate (RT-VBR) or non-real-time variable bit rate (NRT-VBR). See "service-category" on page 10-56.

Syntax	*Example*
SCR <number>	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **scr 75** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display scr** SCR : 75
NO SCR	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **no scr** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display scr** SCR : 175

Default Value

175 cells per second

Command Level

atm subinterface

Related Commands

pcr
service-category

service-category

Set the Quality of Service category for a specific ATM PVC. Use this command to set the level of throughput and delay for a virtual channel. Each virtual channel can have a different Quality of Service, depending on the needs of the customer and the type of data being transmitted over the PVC.

The Quality of Service available is based on the ATM adaptation layer (AAL), which is analogous to the data link layer of the OSI reference model. There are three AAL layers available.

- **AAL1** is a connection-oriented service, suitable for handling circuit-emulation applications such as voice and video conferencing.
- **AAL3/4** supports both connection-oriented and connectionless data. It is designed for network service providers and is closely aligned with the switched multimegabit data service (SMDS).
- **AAL5** is the primary AAL for data, and supports both connection-oriented and connectionless data. It is used to transfer most non-SMDS data, such as IP over ATM and LAN emulation (LANE).

Table 10-6 shows the service categories available.

Table 10-6 Service Categories

Parameter	*Definition*
ABR	**Available bit rate**. ABR allows the subinterface to grab bandwidth, depending on network resource availability. You can set a minimum cell rate (MCR), which is the bandwidth guaranteed by the network for that connection. You can also set the initial cell rate (ICR) as your starting point, and the peak cell rate (PCR) to enable additional bandwidth when required. The ABR functions under AAL1 and is the next most common Quality of Service setting to the default UBR.
CBR	**Constant bit rate**. ATM cells must transmit at a uniform rate. CBR cells are not discarded, no matter how congested the ATM transmission. Use this service category for voice and edge devices that propagate traffic across an entire network to other edge devices. The CBR functions under AAL1, and is the most rigorous service category.
NRT-VBR	**Non-real-time variable bit rate**. Use for time-sensitive multimedia applications, such as multimedia email. The NRT-VBR functions under AAL3/4.

Table 10-6 Service Categories (Continued)

Parameter	*Definition*
RT-VBR	**Real-time variable bit rate**. Use for time-sensitive multimedia applications, such as voice over IP, which is extra sensitive to time delays. The RT-VBR functions under AAL3/4.
UBR	**Unspecified bit rate**. Use for broad networking applications, such as TCP/IP. The UBR functions under AAL5 and is the most commonly used setting for most applications. This is the default setting.

When you set your service category, you also have to set certain cell rates, some of which are not available until you set the service category:

- **Initial cell rate** (“icr” on page 10-36) is used with ABR
- **Maximum burst size** (“mbs” on page 10-38) is used with RT-VBR and NRT-VBR
- **Minimum cell rate** (“mcr” on page 10-40) is used with ABR
- **Peak cell rate** (“pcr” on page 10-41) is used with all service categories
- **Sustainable cell rate** (“scr” on page 10-55) is used with RT-VBR and NRT-VBR

Table 10-7 shows the relationship between cell rate settings and service categories.

Table 10-7 Cell Rates Needed for Service Categories

Cell Rate	*ABR*	*CBR*	*NRT-VBR*	*RT-VBR*	*UBR*
ICR	Yes	No	No	No	No
MBS	No	No	Yes	Yes	No
MCR	Yes	No	No	No	No
PCR	Yes	Yes	Yes	Yes	Yes
SCR	No	No	Yes	Yes	No

Syntax	*Example*
SERVICE-CATEGORY {ABR | CBR | NRT-VBR | RT-VBR | UBR}	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **service-category abr** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display** SLOT : 1 PORT : 1 CHANNEL : 0 SUBIF : 40 VPI : 0 VCI : 0 SERVICE-CATEGORY: ABR CONNECTION-KIND : PVC PVC-TYPE : PVCC PROTOCOL : NONE PCR : 175 ICR : 175 MCR : 175 TRANSMIT SDU SIZE : 1792 RECEIVE SDU SIZE : 1792 OAM-LOOPBACK : DISABLE ENCAPSULATION : LCC-ROUTED-ENCAPSULATION ADMIN-STATUS : DISABLE OPER-STATUS : DOWN ARP General Information: MAC-ADDR : 02:02:00:00:00:64 ARP-AGE : 15 ARP-MAX-ENTRIES : 32 IP Address(es): IPv4 Addresses | Type 0.0.0.0 PROXY 10.10.10.10 PROXY STATIC ARP(s): List is empty DYNAMIC ARP(s): List is empty

Syntax	Example
SERVICE-CATEGORY {ABR | CBR | NRT-VBR | RT-VBR | UBR}	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **service-category cbr** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display** SLOT : 1 PORT : 1 CHANNEL : 0 SUBIF : 40 VPI : 0 VCI : 0 SERVICE-CATEGORY: CBR CONNECTION-KIND : PVC PVC-TYPE : PVCC PROTOCOL : NONE PCR : 175 TRANSMIT SDU SIZE : 1792 RECEIVE SDU SIZE : 1792 OAM-LOOPBACK : DISABLE ENCAPSULATION : LCC-ROUTED-ENCAPSULATION ADMIN-STATUS : DISABLE OPER-STATUS : DOWN ARP General Information: MAC-ADDR : 02:02:00:00:00:64 ARP-AGE : 15 ARP-MAX-ENTRIES : 32 IP Address(es): IPv4 Addresses | Type 0.0.0.0 PROXY 10.10.10.10 PROXY STATIC ARP(s): List is empty DYNAMIC ARP(s): List is empty

Syntax	*Example*
SERVICE-CATEGORY {ABR | CBR | NRT-VBR | RT-VBR | UBR}	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **service-category nrt-vbr** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display** SLOT : 1 PORT : 1 CHANNEL : 0 SUBIF : 40 VPI : 0 VCI : 0 SERVICE-CATEGORY: NRT-VBR CONNECTION-KIND : PVC PVC-TYPE : PVCC PROTOCOL : NONE PCR : 175 SCR : 175 MBS : 175 TRANSMIT SDU SIZE : 1792 RECEIVE SDU SIZE : 1792 OAM-LOOPBACK : DISABLE ENCAPSULATION : LCC-ROUTED-ENCAPSULATION ADMIN-STATUS : DISABLE OPER-STATUS : DOWN ARP General Information: MAC-ADDR : 02:02:00:00:00:64 ARP-AGE : 15 ARP-MAX-ENTRIES : 32 IP Address(es): IPv4 Addresses | Type 0.0.0.0 PROXY 10.10.10.10 PROXY STATIC ARP(s): List is empty DYNAMIC ARP(s): List is empty

Syntax	*Example*
SERVICE-CATEGORY {ABR | CBR | NRT-VBR | RT-VBR | UBR}	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **service-category rt-vbr** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display** SLOT : 1 PORT : 1 CHANNEL : 0 SUBIF : 40 VPI : 0 VCI : 0 SERVICE-CATEGORY: RT-VBR CONNECTION-KIND : PVC PVC-TYPE : PVCC PROTOCOL : NONE PCR : 175 SCR : 175 MBS : 175 TRANSMIT SDU SIZE : 1792 RECEIVE SDU SIZE : 1792 OAM-LOOPBACK : DISABLE ENCAPSULATION : LCC-ROUTED-ENCAPSULATION ADMIN-STATUS : DISABLE OPER-STATUS : DOWN ARP General Information: MAC-ADDR : 02:02:00:00:00:64 ARP-AGE : 15 ARP-MAX-ENTRIES : 32 IP Address(es): IPv4 Addresses | Type 0.0.0.0 PROXY 10.10.10.10 PROXY STATIC ARP(s): List is empty DYNAMIC ARP(s): List is empty

Syntax	*Example*
SERVICE-CATEGORY {ABR | CBR | NRT-VBR | RT-VBR | UBR}	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **service-category ubr** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display** SLOT : 1 PORT : 1 CHANNEL : 0 SUBIF : 40 VPI : 0 VCI : 0 SERVICE-CATEGORY: UBR CONNECTION-KIND : PVC PVC-TYPE : PVCC PROTOCOL : NONE PCR : 175 TRANSMIT SDU SIZE : 1792 RECEIVE SDU SIZE : 1792 OAM-LOOPBACK : DISABLE ENCAPSULATION : LCC-ROUTED-ENCAPSULATION ADMIN-STATUS : DISABLE OPER-STATUS : DOWN ARP General Information: MAC-ADDR : 02:02:00:00:00:64 ARP-AGE : 15 ARP-MAX-ENTRIES : 32 IP Address(es): IPv4 Addresses | Type 0.0.0.0 PROXY 10.10.10.10 PROXY STATIC ARP(s): List is empty DYNAMIC ARP(s): List is empty

Syntax	Example
NO SERVICE-CATEGORY	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **no service-category** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display** SLOT : 1 PORT : 1 CHANNEL : 0 SUBIF : 40 VPI : 0 VCI : 0 SERVICE-CATEGORY: UBR CONNECTION-KIND : PVC PVC-TYPE : PVCC PROTOCOL : NONE PCR : 175 TRANSMIT SDU SIZE : 1792 RECEIVE SDU SIZE : 1792 OAM-LOOPBACK : DISABLE ENCAPSULATION : LCC-ROUTED-ENCAPSULATION ADMIN-STATUS : DISABLE OPER-STATUS : DOWN ARP General Information: MAC-ADDR : 02:02:00:00:00:64 ARP-AGE : 15 ARP-MAX-ENTRIES : 32 IP Address(es): IPv4 Addresses | Type 0.0.0.0 PROXY 10.10.10.10 PROXY STATIC ARP(s): List is empty DYNAMIC ARP(s): List is empty

Default Value

UBR

Command Level

atm subinterface

Related Commands

icr
mbs
mcr
pcr
scr

transmit-size

Set the transmission size of the service data units (SDUs). The range is from 0 to 9188 bytes, but currently up to 1792 bytes are supported.

Syntax	*Example*
TRANSMIT-SIZE <number>	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **transmit-size 512** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display transmit** TRANSMIT SDU SIZE : 512
NO TRANSMIT-SIZE	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **no transmit-size** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display transmit** TRANSMIT SDU SIZE : 1792

Default Value

1792 bytes

Command Level

atm subinterface

Related Commands

receive-size

vci

Set a virtual channel identifier (VCI). A VCI is a unique numerical tag defined by a 16-bit field in the ATM cell header that identifies a virtual channel. ATM cells travel over channels. Values 0–31 are reserved for OAM and control functions. The VCI value can be from 32 up to the maximum range allowed in the virtual path. Refer to the command "vpi" on page 10-69. Table 10-8 describes possible VCI values.

Table 10-8 VCI Values

If Max-VC-Number is...	*Use VCI values between...*
16384	32 and 16383
8192	32 and 8191
4096	32 and 4095
64	32 and 63

Syntax	*Example*
VCI <number>	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **vci 4096** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display** SLOT : 1 PORT : 1 CHANNEL : 0 SUBIF : 40 VPI : 0 VCI : 4096 SERVICE-CATEGORY: UBR CONNECTION-KIND : PVC PVC-TYPE : PVCC PROTOCOL : NONE PCR : 175 TRANSMIT SDU SIZE : 1792 RECEIVE SDU SIZE : 1792 OAM-LOOPBACK : DISABLE ENCAPSULATION : LCC-ROUTED-ENCAPSULATION ADMIN-STATUS : DISABLE OPER-STATUS : DOWN ARP General Information: MAC-ADDR : 02:02:00:00:00:64 ARP-AGE : 15 ARP-MAX-ENTRIES : 32 IP Address(es): IPv4 Addresses | Type 0.0.0.0 PROXY 10.10.10.10 PROXY STATIC ARP(s): List is empty DYNAMIC ARP(s): List is empty

Default Value

0

Command Level

atm subinterface

Related Commands

vpi

vpi

Set a virtual path identifier (VPI). A VPI is a unique numerical tag defined by an 8-bit field in the ATM cell header that identifies a virtual path (VP). A VP consists of multiple VCs that are bundled together. The VPI value can be from 0 up to the maximum range allowed. Refer to the command "vci" on page 10-66. Table 10-9 describes possible VPI values.

Table 10-9 VPI Values

If Max-VP-Number is ...	*Use VPI values between...*
1	0
2	0 and 1
4	0 and 3
256	0 and 255

Syntax	*Example*
VPI <number>	IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **vpi 1** IPNOS : CONFIG INTERFACE ATM(1/1:0:40)# **display** SLOT : 1 PORT : 1 CHANNEL : 0 SUBIF : 40 VPI : 1 VCI : 4096 SERVICE-CATEGORY: UBR CONNECTION-KIND : PVC PVC-TYPE : PVCC PROTOCOL : NONE PCR : 175 TRANSMIT SDU SIZE : 1792 RECEIVE SDU SIZE : 1792 OAM-LOOPBACK : DISABLE ENCAPSULATION : LCC-ROUTED-ENCAPSULATION ADMIN-STATUS : DISABLE OPER-STATUS : DOWN ARP General Information: MAC-ADDR : 02:02:00:00:00:64 ARP-AGE : 15 ARP-MAX-ENTRIES : 32 IP Address(es): IPv4 Addresses | Type 0.0.0.0 PROXY 10.10.10.10 PROXY STATIC ARP(s): List is empty DYNAMIC ARP(s): List is empty

Default Value

0

Command Level

atm subinterface

Related Commands

vci

CHAPTER

11 POS Commands

Use these commands to configure Packet over SONET (POS). This chapter describes the following commands:

Command	*Page*
pos	11-2
display	11-2
encapsulation	11-7

pos

Go to the POS interface level. The following commands are available.

- "display" on page 11-2
- "encapsulation" on page 11-7
- "oc3" on page 9-2 (OC-3c blade physical interface)

Syntax	*Example*
Configure the POS interface: `INTERFACE POS <slot>/<port>`	`IPNOS : CONFIG#` **`interface pos 4/1`** `IPNOS : CONFIG INTERFACE POS(4/1)#`

Default Value

None

Command Level

interface command

Related Commands

oc3 interface

display

View the current POS interface and OC-3c interface configuration. See "oc3" on page 9-2 for OC-3 **display** information. Table 11-1 lists POS monitoring statistics that come up using the **display stats** command.

Table 11-1 POS Monitoring Statistics

Statistic	*Description*
rx_bytes	Number of bytes received in the POS connection.
tx_bytes	Number of bytes transmitted in the POS connection.
rx_frames	Number of packets in the received transmission.
tx_frames	Number of packets in the sent transmission.
rx_fcs_er	Number of frame check sequence errors received in the POS connection.

Table 11-1 POS Monitoring Statistics (Continued)

Statistic	*Description*
rx_abort_er	Number of abort error messages transmitted through the SONET path.
rx_min_er	Below minimum packet size.
rx_max_er	Above maximum packet size.
tx_user_abt	User aborts transmission.
tx_urun_abt	FIFO buffer underrun event.

Syntax	*Example*

POS interface:
`DISPLAY`
	`IPNOS : CONFIG INTERFACE POS(4/1)# display`
	`SLOT : 4`
	`PORT : 1`
	`ENCAPSULATION : NONE`
	
	`OC-3 PARAMETERS`
	`FRAMING : SONET`
	`CLOCK : INTERNAL`
	`J0 : 0xCC`
	`SCRAMBLE : ENABLE`
	`ALARM : ENABLE`
	`PATH-TRACE :`
	`REMOTE-PATH-TRACE :`
	`LOOP : NONE`
	`ADMIN STATUS : ENABLE`
	`OPER STATUS : UP -`
	
	`VALID INTERVALS : 12`
	`INVALID INTERVALS : 0`
	`ELAPSED TIME(sec) : 615`

Syntax	*Example*

Display current OC-3 interval and POS stats on OC-3c POS blade:
`DISPLAY STATS [0 | 1..32 | SUM]`
	`IPNOS : CONFIG INTERFACE POS(4/1)#` **`display stats 0`**
	
	`OC-3 STATISTICS`
	`Current Interval - elapsed time(min:sec) 0:11`
	
	`Param. Near End`
	`Section:`
	`CV_s 0`
	`ES_s 0`
	`SES_s 0`
	`SEFS_s 0`
	
	`Line:`
	`CV_l 0`
	`ES_l 0`
	`SES_l 0`
	`UAS_l 0`
	`FC_l 0`
	
	`Path:`
	`CV_p 0`
	`ES_p 0`
	`SES_p 0`
	`UAS_p 0`
	`FC_p 0`
	
	`POS STATISTICS`
	`Counters:`
	`Rx_bytes 0`
	`Rx_frames 0`
	`Rx_abort_er 0`
	`Rx_fcs_er 0`
	`Rx_min_er 0`
	`Rx_max_er 0`
	`Tx_bytes 0`
	`Tx_frames 0`
	`Tx_user_abt 0`
	`Tx_urun_abt 0`

Syntax	*Example*

Display previous OC-3 interval and POS stats on OC-3c POS blade:
`DISPLAY STATS [0 | 1..32 | SUM]`
	`IPNOS : CONFIG INTERFACE POS(4/1)# display stats 12`
	
	`OC-3 STATISTICS`
	`Current Interval - elapsed time(min:sec) 5:26`
	
	`Param. Near End`
	`Section:`
	`CV_s 0`
	`ES_s 0`
	`SES_s 0`
	`SEFS_s 0`
	
	`Line:`
	`CV_l 0`
	`ES_l 0`
	`SES_l 0`
	`UAS_l 0`
	`FC_l 0`
	
	`Path:`
	`CV_p 0`
	`ES_p 0`
	`SES_p 0`
	`UAS_p 0`
	`FC_p 0`
	
	`POS STATISTICS`
	`Counters:`
	`Rx_bytes 0`
	`Rx_frames 0`
	`Rx_abort_er 0`
	`Rx_fcs_er 0`
	`Rx_min_er 0`
	`Rx_max_er 0`
	`Tx_bytes 0`
	`Tx_frames 0`
	`Tx_user_abt 0`
	`Tx_urun_abt 0`

Syntax	*Example*

Display OC-3 interval and POS stats summary on OC-3c POS blade:
`DISPLAY STATS [0 | 1..32 | SUM]`
	`IPNOS : CONFIG INTERFACE POS(4/1)#` **`display stats sum`**
	
	`OC-3 STATISTICS`
	
	`Param. Cur. Intv Prev. Intv Cur. Day Prev. Day`
	
	`Section:`
	`CV_s 0 0 0 0`
	`ES_s 0 0 0 0`
	`SES_s 0 0 0 0`
	`SEFS_s 0 0 0 0`
	
	`Line:`
	`CV_l 0 0 0 0`
	`ES_l 0 0 0 0`
	`SES_l 0 0 0 0`
	`UAS_l 0 0 0 0`
	`FC_l 0 0 0 0`
	
	`Path:`
	`CV_p 0 0 0 0`
	`ES_p 0 0 0 0`
	`SES_p 0 0 0 0`
	`UAS_p 0 0 0 0`
	`FC_p 0 0 0 0`
	
	`POS STATISTICS`
	`Counters:`
	`Rx_bytes 0`
	`Rx_frames 0`
	`Rx_abort_er 0`
	`Rx_fcs_er 0`
	`Rx_min_er 0`
	`Rx_max_er 0`
	`Tx_bytes 0`
	`Tx_frames 0`
	`Tx_user_abt 0`
	`Tx_urun_abt 0`

Default Value

None

Command Level

pos interface

Related Commands

None

encapsulation

Encapsulate the POS interface and go to the PPP, Frame Relay, or HDLC command level. See the following chapters for commands.

- Chapter 12, “Frame Relay Commands”
- Chapter 13, “HDLC Commands”
- Chapter 14, “PPP Commands”

Syntax	*Example*
ENCAPSULATION {FRAME-RELAY | HDLC | PPP}	IPNOS : CONFIG INTERFACE POS(4/1)# **encap frame-relay** INTERFACE POS(4/1): Object successfully set IPNOS : CONFIG INTERFACE POS(4/1) FRAME-RELAY#
ENCAPSULATION {FRAME-RELAY | HDLC | PPP}	IPNOS : CONFIG INTERFACE POS(4/1)# **encapsulation HDLC** INTERFACE POS(4/1): Object successfully set IPNOS : CONFIG INTERFACE POS(4/1) HDLC#
ENCAPSULATION {FRAME-RELAY | HDLC | PPP}	IPNOS : CONFIG INTERFACE POS(4/1)# **encapsulation PPP** INTERFACE POS(4/1): Object successfully set IPNOS : CONFIG INTERFACE POS(4/1) PPP#
[NO] ENCAPSULATION	IPNOS : CONFIG INTERFACE POS(4/1)# **no encapsulation** INTERFACE POS(4/1): Object successfully set IPNOS : CONFIG INTERFACE POS(4/1)#

Default Value

None

Command Level

pos interface

Related Commands

interface ethernet encapsulation (control blade)
interface serial encapsulation (channelized E3 blade)
interface serial encapsulation (channelized DS3 blade)

interface serial encapsulation (unchannelized DS3 blade)
interface serial encapsulation (unchannelized E3 blade)

CHAPTER

12 Frame Relay Commands

Use these commands to configure Frame Relay interface and subinterface. You must encapsulate the blade or channel to Frame Relay before you use these commands. This chapter describes the following commands:

Command	*Page*
display	12-2
frame-relay	12-3
pos	12-7
serial	12-8
display	12-9
frame-relay	12-10

display

View the current configuration of the Frame Relay interface.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG INTERFACE SERIAL(4/1:20) FRAME-RELAY# **display** SLOT : 4 PORT : 1 CHANNEL : 20 LMI-TYPE : ANSI LMI-T391DTE : 10 LMI-N391DTE : 5 LMI-N392DTE : 3 LMI-N393DTE : 4 INTF-TYPE : DTE LMI-N392DCE : 3 LMI-N393DCE : 4 LMI-T392DCE : 15 ADDRESS : Q922 LENGTH : TWO-OCTETS MAX-SUPPORTED-VC: 317 MULTICAST : NON-BROADCAST STATUS : RUNNING

Default Value

None

Command Level

frame-relay

Related Commands

display (HDLC encapsulation)
display (PPP encapsulation)
show

frame-relay

Configure the Frame Relay local management interface (LMI).

NOTE: Both ends of the Frame Relay permanent virtual circuit (PVC) must have the same LMI settings or the connection will fail.

Table 12-1 describes the parameters available.

Table 12-1 LMI Parameters

Parameter	*Description*
INTF-TYPE	Interface type. Choose the following equipment: • DCE—data communications equipment • DTE—data terminal equipment
LMI-T391DTE	DTE polling interval time. The default value is 10 seconds. Choose from 5–30.
LMI-N391DTE	DTE number of polling intervals before issuing a status inquiry response. Choose from 1–255.
LMI-N392DTE	DTE error threshold for the DTE. The default value is 3 seconds. Choose from 1–10.
LMI-N393DTE	DTE monitored events count. The default value is 4 seconds. Choose from 1–10.
LMI-T392DCE	DCE polling verification time. The default value is 15 seconds. Choose from 5–30.
LMI-N392DCE	DCE error threshold for the DCE. The default value is 3 seconds. Choose from 1–10.
LMI-N393DCE	DCE monitored events count. The default value is 4 seconds. Choose from 1–10.
LMI-TYPE	Set the LMI type. The following types are available: • ANSI—ANSI annex D • CISCO—Cisco LMI • NONE—No LMI configured • Q933A—CCITT Q993 annex A • REV1—LMI revision 1
No	Reset the parameter to its default value.

Syntax	*Example*
FRAME-RELAY INTF-TYPE {DCE | DTE}	IPNOS : CONFIG INTERFACE SERIAL(4/1:1) FRAME-RELAY# **frame-relay intf-type dce** IPNOS : CONFIG INTERFACE SERIAL(4/1:1) FRAME-RELAY# **dis** SLOT : 4 PORT : 1 CHANNEL : 1 LMI-TYPE : ANSI LMI-T391DTE : 10 LMI-N391DTE : 5 LMI-N392DTE : 3 LMI-N393DTE : 4 INTF-TYPE : DCE LMI-N392DCE : 3 LMI-N393DCE : 4 LMI-T392DCE : 15 ADDRESS : Q922 LENGTH : TWO-OCTETS MAX-SUPPORTED-VC: 317 MULTICAST : NON-BROADCAST STATUS : RUNNING
FRAME-RELAY INTF-TYPE {DCE | DTE}	IPNOS : CONFIG INTERFACE SERIAL(4/1:1) FRAME-RELAY# **frame-relay intf-type dte** IPNOS : CONFIG INTERFACE SERIAL(4/1:1) FRAME-RELAY# **dis** SLOT : 4 PORT : 1 CHANNEL : 1 LMI-TYPE : ANSI LMI-T391DTE : 10 LMI-N391DTE : 5 LMI-N392DTE : 3 LMI-N393DTE : 4 INTF-TYPE : DTE LMI-N392DCE : 3 LMI-N393DCE : 4 LMI-T392DCE : 15 ADDRESS : Q922 LENGTH : TWO-OCTETS MAX-SUPPORTED-VC: 317 MULTICAST : NON-BROADCAST STATUS : RUNNING

Syntax	*Example*
FRAME-RELAY {LMI-N391DTE | LMI-N392DTE | LMI-393DTE | LMI-T391DTE | LMI-N392DCE | LMI-N393DCE | LMI-T392DCE} <value>	IPNOS : CONFIG INTERFACE SERIAL(4/1) FRAME-RELAY# **frame-relay lmi-n391dte 3**
FRAME-RELAY {LMI-N391DTE | LMI-N392DTE | LMI-393DTE | LMI-T391DTE | LMI-N392DCE | LMI-N393DCE | LMI-T392DCE} <value>	IPNOS : CONFIG INTERFACE SERIAL(4/1) FRAME-RELAY# **frame-relay lmi-n392dte 6**
FRAME-RELAY {LMI-N391DTE | LMI-N392DTE | LMI-393DTE | LMI-T391DTE | LMI-N392DCE | LMI-N393DCE | LMI-T392DCE} <value>	IPNOS : CONFIG INTERFACE SERIAL(4/1) FRAME-RELAY# **frame-relay lmi-n393dte 8**
FRAME-RELAY {LMI-N391DTE | LMI-N392DTE | LMI-393DTE | LMI-T391DTE | LMI-N392DCE | LMI-N393DCE | LMI-T392DCE} <value>	IPNOS : CONFIG INTERFACE SERIAL(4/1) FRAME-RELAY# **frame-relay lmi-t391dte 10**
FRAME-RELAY {LMI-N391DTE | LMI-N392DTE | LMI-393DTE | LMI-T391DTE | LMI-N392DCE | LMI-N393DCE | LMI-T392DCE} <value>	IPNOS : CONFIG INTERFACE SERIAL(4/1) FRAME-RELAY# **frame-relay lmi-n392dce 3**
FRAME-RELAY {LMI-N391DTE | LMI-N392DTE | LMI-393DTE | LMI-T391DTE | LMI-N392DCE | LMI-N393DCE | LMI-T392DCE} <value>	IPNOS : CONFIG INTERFACE SERIAL(4/1) FRAME-RELAY# **frame-relay lmi-n393dce 4**

Syntax	*Example*
FRAME-RELAY {LMI-N391DTE | LMI-N392DTE | LMI-393DTE | LMI-T391DTE | LMI-N392DCE | LMI-N393DCE | LMI-T392DCE} <value>	IPNOS : CONFIG INTERFACE SERIAL(4/1) FRAME-RELAY# **frame-relay lmi-t392dce 15**
[NO] FRAME-RELAY LMI-TYPE {ANSI | CISCO | Q933A | NONE | REV1}	IPNOS : CONFIG INTERFACE SERIAL(4/1) FRAME-RELAY# **frame-relay lmi-type ansi**

Default Value

LMI type is ANSI.

Command Level

frame-relay

Related Commands

None

pos

Go to the Frame Relay permanent virtual circuit (PVC) subinterface command level on an OC-3c Packet Over SONET (POS) blade. See "encapsulation" on page 11-7 to encapsulate the POS interface for Frame Relay. The following commands are available.

- "display" on page 12-9
- "frame-relay" on page 12-10

Syntax	*Example*
`INTERFACE POS <slot>/<port>:<channel>:<sub interface> FRAME-RELAY INTERFACE-DLCI <dlci>`	`IPNOS : CONFIG#` **`interface pos 4/1:0:40 frame-relay interface-dlci 16`** `IPNOS : CONFIG INTERFACE POS(4/1:0:40)#`
Create or configure a POS Frame Relay subinterface: `INTERFACE POS <slot>/<port>:<channel>:<sub interface>`	`IPNOS : CONFIG#` **`interface pos 4/1:0:40`** `IPNOS : CONFIG INTERFACE POS(4/1:0:40)#`

Delete a POS Frame Relay subinterface:
`[NO] INTERFACE POS <slot>/<port>:<channel>:<sub interface>`
	`IPNOS : CONFIG#` **`no interface pos 4/1:0:40`**
	`INTERFACE POS(2/1:0:40): Object successfully deleted`
	
	`IPNOS : CONFIG#`

Default Value

None

Command Level

config

Related Commands

interface serial (subinterface)

serial

Go to the Frame Relay permanent virtual circuit (PVC) subinterface command level on a channelized or unchannelized DS3 or E3 blade. See the "encapsulation" command for each blade to encapsulate the interface for Frame Relay.

The following commands are available.

- "display" on page 12-9
- "frame-relay" on page 12-10

Syntax	*Example*
`INTERFACE SERIAL <slot>/<port>:l<channel>:<subinterface>`	`IPNOS : CONFIG#` **`interface serial 2/1:0:40`** `IPNOS : CONFIG INTERFACE SERIAL(2/1:0:40)#`

Default Value

None

Command Level

config

Related Commands

interface pos (subinterface)

display

View the current status of the Frame Relay PVC.

Syntax	*Example*
DISPLAY	IPNOS: CONFIG INTERFACE SERIAL(6/1:0:1)# dis SLOT : 6 PORT : 1 CHANNEL : 0 SUBIF : 1 INTERFACE-DLCI : 0 ROUTE : No Route BC : 0 BE : 1024 CIR : 0 RATE-POLICE : DISABLED OPER-STATUS : UNKNOWN PROTOCOL : NONE RECEIVED-FECN : 0 RECEIVED-BECN : 0 SENT-FRAMES : 0 SENT-OCTETS : 0 RECEIVED-FRAMES : 0 RECEIVED-OCTETS : 0 DISCARDS : 0 RECEIVED-DE : 0 SENT-DE : 0
DISPLAY <argument>	IPNOS: CONFIG INTERFACE SERIAL(6/1:0:1)# display be BE : 1024

Default Value

None

Command Level

serial

Related Commands

None

frame-relay

Configure a Frame Relay PVC. Table 12-2 describes the parameters available.

Table 12-2 Frame Relay PVC Subinterface Parameters

Parameter	*Description*
Bc	Committed burst size (BC) for a measurement interval.The interval is determined by dividing BC by CIR. Typically, BC is set to the same value as the CIR, in which case the measurement interval is one second. If CIR is not zero, BC must not be zero. Choose from 0–2,147,483,647.
Be	Excess burst size (BE). The maximum amount of data beyond the committed burst size that may be burst-transmitted by the provider over the measurement interval. If CIR is zero, BE must not be zero. Choose from 0–2,147,483,647.
Cir	Committed information rate (CIR). The minimum assigned bandwidth for the subinterface. If CIR is set to zero, BC is ignored and the measurement interval is determined by dividing BE by the bandwidth of the host channel. Choose from 0–2,147,483,647.
Protocol	Set IP or PPP over Frame Relay. You need to disable the subinterface before changing the protocol. If the current protocol is IP, then you must also disable the VI connection at the subinterface before changing the protocol. Changing the protocol will automatically disconnect any PPP connections to the subinterface. Note: The protocol is not automatically changed to IP when you run **no frame-relay protocol**. You also need to attach a VI to the subinterface to change the protocol to IP.
Interface-dlci	Data link connection identifier (DLCI). Set and commit this parameter before configuring the rest of the PVC. Only one DLCI is allowed per subinterface. Place on an endpoint of the connection. Choose from 16 to 991. *Note:* You cannot remove a DLCI without removing the subinterface.
Rate-police	Enable ingress traffic rate-policing. Ingress rate policing is controlled by the values configured for BC, BE, and CIR. By default, this command is disabled. When enabled, a frame is processed in one of three ways: • If the number of bits received in the measurement interval is less than or equal to the committed burst size, the frame is forwarded for further processing. • If the number of bits received in the measurement interval is greater than the committed burst size but less than the sum of committed burst size and excess burst size, the discard enable (DE) bit is set in the frame and then forwarded for further processing. • If the number of bits received in the measurement interval is greater than the sum of committed burst size and excess burst size, the frame is dropped. *Note:* Set BC, BE, and CIR appropriately. Do not use default values.
Route	Switch frame forwarding between another DLCI on the same blade. Set the port, slot, channel (default is 0), and another DLCI number.

Syntax	*Example*
[NO] FRAME-RELAY BC <value>	IPNOS : CONFIG INTERFACE SERIAL(2/1:0:40)# **frame-relay bc 57600** IPNOS : CONFIG INTERFACE SERIAL(2/1:0:40)# **display bc** BC : 57600
[NO] FRAME-RELAY BE <value>	IPNOS : CONFIG INTERFACE SERIAL(2/1:0:40)# **frame-relay be 131072** IPNOS : CONFIG INTERFACE SERIAL(2/1:0:40)# **display be** BE : 131072
[NO] FRAME-RELAY CIR <value>	IPNOS : CONFIG INTERFACE SERIAL(2/1:0:40)# **frame-relay cir 57600** IPNOS : CONFIG INTERFACE SERIAL(2/1:0:40)# **display cir** CIR : 57600
[NO] FRAME-RELAY PROTOCOL PPP	IPNOS : CONFIG INTERFACE SERIAL(2/1:0:40)# **frame-relay protocol ppp** IPNOS : CONFIG INTERFACE SERIAL(2/1:0:40)# **display** PROTOCOL : PPP
[NO] FRAME-RELAY PROTOCOL PPP	IPNOS : CONFIG INTERFACE SERIAL(2/1:0:40)# **no frame-relay protocol** IPNOS : CONFIG INTERFACE SERIAL(2/1:0:40)# **display** PROTOCOL : NONE
FRAME-RELAY INTERFACE-DLCI <value>	IPNOS : CONFIG INTERFACE SERIAL(2/1:0:40)# **frame-relay interface-dlci 3** IPNOS : CONFIG INTERFACE SERIAL(2/1:0:40)# **display interface** INTERFACE-DLCI : 3
[NO] FRAME-RELAY RATE-POLICE	IPNOS : CONFIG INTERFACE SERIAL(2/1:0:40)# **frame-relay rate-police** IPNOS : CONFIG INTERFACE SERIAL(2/1:0:40)# **display rate-police** RATE-POLICE : ENABLED

Syntax	*Example*
[NO] FRAME-RELAY RATE-POLICE	IPNOS : CONFIG INTERFACE SERIAL(2/1:0:40)# no **frame-relay rate-police** IPNOS : CONFIG INTERFACE SERIAL(2/1:0:40)# **display rate-police** RATE-POLICE : DISABLED
[NO] FRAME-RELAY ROUTE <slot>/<port>:<channel> DLCI <dlci>	IPNOS : CONFIG INTERFACE SERIAL(2/1:0:40)# **frame-relay route 2/1:0 dlci 43** IPNOS : CONFIG INTERFACE SERIAL(2/1:0:40)# **display route** ROUTE : 2/1:0 DLCI 43

Default Value

See Table 12-2.

Command Level

serial

Related Commands

None

CHAPTER

13 HDLC Commands

Use this command to configure the high-level data link control (HDLC) encapsulation. You must encapsulate the blade or channel to HDLC before getting to this level.

display

View the current HDLC configuration.

Syntax	*Example*
Unchannelized DS3: DISPLAY	IPNOS: CONFIG INTERFACE SERIAL(4/1) HDLC# **display** SLOT : 4 PORT : 1 CHANNEL : 0 KEEPALIVE : 10 ADMIN-STATUS : ENABLE OPER-STATUS : DOWN
Channelized DS3: DISPLAY	IPNOS: CONFIG INTERFACE SERIAL(3/1:1) HDLC# **display** SLOT : 3 PORT : 1 CHANNEL : 1 KEEPALIVE : 10 ADMIN-STATUS : ENABLE OPER-STATUS : DOWN
DISPLAY <argument>	IPNOS: CONFIG INTERFACE SERIAL(3/1:1) HDLC# **display channel** CHANNEL : 1

Default Value

None

Command Level

HDLC

Related Commands

display (Frame Relay encapsulation)
display (PPP encapsulation)
show

CHAPTER

14 PPP Commands

Use these commands to configure the Point-to-Point Protocol (PPP). To aggregate packet transport over several PPP links, use the Multilink PPP (MLPPP) feature. You must encapsulate the blade or channel to PPP before you usethese commands. This chapter describes the following commands:

Command	*Page*
display	14-2
event-log	14-3
ppp	14-7
bundle	14-9
admin-status	14-10
display	14-11
link	14-12
max-links	14-13
queue-size	14-13
reorder-timeout	14-14
site-name	14-14

display

View the current PPP configuration.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG INTERFACE SERIAL(4/1:1) PPP# **display** SLOT : 4 PORT : 1 CHANNEL : 1 AUTHENTICATION : NONE CONFIG-REQ-RETRANSMIT: 10 TERM-REQ-RETRANSMIT: 2 TIMEOUT : 5 REAUTH : 0 ADMIN-STATUS : DISABLE OPER-STATUS : DOWN
DISPLAY <argument>	IPNOS : CONFIG INTERFACE SERIAL(4/1:1) PPP# **display admin-status** ADMIN-STATUS : DISABLE

Default Value

None

Command Level

ppp

Related Commands

display (Frame Relay encapsulation)
display (HDLC encapsulation)
show

event-log

Enable the display of event messages to the screen. There are two parameters.

The **area** parameter filters event messages by type. Use the **no event-log area <areatype>** to not show messages of a specific area. Use the no **event-log area** to reset the parameter to its default value. Use the **no event-log area all** to disable all areas. Table 14-1 describes the area types.

Table 14-1 PPP Event Message Area Types

Area	*Description*
All	All possible log messages.
Apps	Application messages, such as firewall, RADIUS, or PPP.
Auth	Authentication-related messages.
Configuration	Configuration messages, such as creating a VR.
Emergency	Emergency messages, such as the loss of a control blade.
IPSec	IPSec and IKE specific messages.
Link	Link layer messages, such as PPP, Frame Relay, or ATM.
Protocol	Protocol-related messages, except for IPSec, IKE, or link layers.
System	General system messages.
Trap	SNMP trap messages.

The **level** parameter filters event messages by severity. Choose a lower level to show more messages. Use the **no event-log level** to reset the parameter to its default value. Use the **no event-log** to hide all messages. Table 14-2 describes the levels.

Table 14-2 PPP Event Message Severity Levels

Level	*Description*
Critical	Critical system problems that affect normal IPSX operations, such as overheating, or PPP fails to negotiate any connection. This is the highest level. Only critical level messages will be displayed.
Major	Major problems, such as tunnels down. This is the default level.
Minor	Minor errors, such as transform set not found.

Table 14-2 PPP Event Message Severity Levels

Level	*Description*
Inform	General information, such as PPP connection is up or down.
Debug	Debug codes, such as programming messages. This is the lowest level. All levels of messages will be displayed. *Note:* IPNOS performance will drop dramatically because of sheer number of messages it has to display at this level.

Syntax	*Example*
[NO] EVENT-LOG {AREA ALL | APPS | AUTH | CONFIGURATION | EMERGENCY | IPSEC | LINK | PROTOCOL | SYSTEM | TRAP}	IPNOS: CONFIG INTERFACE SERIAL(2/1:100) PPP# **event-log area all** IPNOS: CONFIG INTERFACE SERIAL(2/1:100) PPP# **display** SLOT : 2 PORT : 1 CHANNEL : 100 SUBIF : 0 AUTHENTICATION : NONE CONFIG-REQ-RETRANSMIT: 10 TERM-REQ-RETRANSMIT: 2 TIMEOUT : 3 REAUTH : 0 ADMIN-STATUS : DISABLE OPER-STATUS : DOWN EVENT-LOG AREA : APPS AUTH CONFIGURATION EMERGENCY IPSEC LINK PROTOCOL SYSTEM TRAP EVENT-LOG LEVEL : MAJOR
[NO] EVENT-LOG {AREA ALL | APPS | AUTH | CONFIGURATION | EMERGENCY | IPSEC | LINK | PROTOCOL | SYSTEM | TRAP}	IPNOS: CONFIG INTERFACE SERIAL(2/1:100) PPP# **no event-log area** IPNOS: CONFIG INTERFACE SERIAL(2/1:100) PPP# **display** SLOT : 2 PORT : 1 CHANNEL : 100 SUBIF : 0 AUTHENTICATION : NONE CONFIG-REQ-RETRANSMIT: 10 TERM-REQ-RETRANSMIT: 2 TIMEOUT : 3 REAUTH : 0 ADMIN-STATUS : DISABLE OPER-STATUS : DOWN EVENT-LOG AREA : LINK EVENT-LOG LEVEL : MAJOR

Syntax	*Example*
[NO] EVENT-LOG {AREA ALL | APPS | AUTH | CONFIGURATION | EMERGENCY | IPSEC | LINK | PROTOCOL | SYSTEM | TRAP}	IPNOS: CONFIG INTERFACE SERIAL(2/1:100) PPP# **event-log area apps** IPNOS: CONFIG INTERFACE SERIAL(2/1:100) PPP# **display** SLOT : 2 PORT : 1 CHANNEL : 100 SUBIF : 0 AUTHENTICATION : NONE CONFIG-REQ-RETRANSMIT: 10 TERM-REQ-RETRANSMIT: 2 TIMEOUT : 3 REAUTH : 0 ADMIN-STATUS : DISABLE OPER-STATUS : DOWN EVENT-LOG AREA : APPS LINK EVENT-LOG LEVEL : MAJOR
EVENT-LOG LEVEL { CRITICAL | MAJOR | MINOR | INFORM | DEBUG }	IPNOS: CONFIG INTERFACE SERIAL(2/1:100) PPP# **event-log level critical** IPNOS: CONFIG INTERFACE SERIAL(2/1:100) PPP# **display** SLOT : 2 PORT : 1 CHANNEL : 100 SUBIF : 0 AUTHENTICATION : NONE CONFIG-REQ-RETRANSMIT: 10 TERM-REQ-RETRANSMIT: 2 TIMEOUT : 3 REAUTH : 0 ADMIN-STATUS : DISABLE OPER-STATUS : DOWN EVENT-LOG AREA : APPS LINK EVENT-LOG LEVEL : CRITICAL

Syntax	*Example*
NO EVENT-LOG LEVEL	IPNOS: CONFIG INTERFACE SERIAL(2/1:100) PPP# **no event-log level** IPNOS: CONFIG INTERFACE SERIAL(2/1:100) PPP# **display** SLOT : 2 PORT : 1 CHANNEL : 100 SUBIF : 0 AUTHENTICATION : NONE CONFIG-REQ-RETRANSMIT: 10 TERM-REQ-RETRANSMIT: 2 TIMEOUT : 3 REAUTH : 0 ADMIN-STATUS : DISABLE OPER-STATUS : DOWN EVENT-LOG AREA : APPS LINK EVENT-LOG LEVEL : MAJOR

Default Value

Link area, Major level

Command Level

PPP

Related Command

None

ppp

Configure PPP. Table 14-3 describes the parameters available.

Table 14-3 PPP Parameters

Parameter	*Description*
Authentication	Authentication algorithm. The default value is none. Choose from three types: • Password Authentication Protocol (PAP) • Challenge Handshake Authentication Protocol, Message Digest 5 (CHAPMD5) • Microsoft Challenge Handshake Authentication Protocol, Version 1 (MSCHAPV1)
Config-req-retransmit	Maximum retransmissions of configuration requests. The default is 10 retransmissions.
Enable	Enable PPP interface. The default value is disabled.
Keepalive	Timeout period for inactive PPP connections. Choose from 0 to 65,535 seconds. The default value is 10 seconds.
Reauth	Time before reauthentication takes place. Select CHAPMD5 or MSCHAPv1 authentication before setting this parameter. The default value is 0.
Term-req-retransmit	Maximum retransmissions of termination requests. The default is 2 retransmissions.
Timeout	Time elapsed before an unanswered request for authentication is disconnected. The default value is 3 seconds.

Syntax	*Example*
`PPP AUTHENTICATION {PAP | CHAPMD5 | MSCHAPV1}`	`IPNOS : CONFIG INTERFACE SERIAL(4/1) PPP#` **`ppp authentication pap`**
`[NO] PPP AUTHENTICATION`	`IPNOS : CONFIG INTERFACE SERIAL(4/1) PPP#` **`no ppp authentication`**
`PPP CONFIG-REQ-RETRANSMIT <value>`	`IPNOS : CONFIG INTERFACE SERIAL(4/1) PPP#` **`ppp config-req-retransmit 10`**
`[NO] PPP CONFIG-REQ-RETRANSMIT`	`IPNOS : CONFIG INTERFACE SERIAL(4/1) PPP#` **`no ppp config-req-retransmit`**
`PPP ENABLE`	`IPNOS : CONFIG INTERFACE SERIAL(4/1) PPP#` **`ppp enable`**
`[NO] PPP ENABLE`	`IPNOS : CONFIG INTERFACE SERIAL(4/1) PPP#` **`no ppp enable`**
`PPP KEEPALIVE <interval-in-secs>`	`IPNOS : CONFIG INTERFACE SERIAL(4/1) PPP#` **`ppp keepalive 150`**

Syntax	*Example*
NO PPP KEEPALIVE	IPNOS : CONFIG INTERFACE SERIAL(4/1) PPP# **no ppp keepalive**
PPP REAUTH<seconds>	IPNOS : CONFIG INTERFACE SERIAL(4/1) PPP# **ppp reauth 0**
[NO] PPP REAUTH	IPNOS : CONFIG INTERFACE SERIAL(4/1) PPP# **no ppp reauth**
PPP TERM-REQ-RETRANSMIT <value>	IPNOS : CONFIG INTERFACE SERIAL(4/1) PPP# **ppp term-req-retransmit 10**
[NO] PPP TERM-REQ-RETRANSMIT	IPNOS : CONFIG INTERFACE SERIAL(4/1) PPP# **no ppp term-req-retransmit**
PPP TIMEOUT <timeout-in-secs>	IPNOS : CONFIG INTERFACE SERIAL(4/1) PPP# **ppp timeout 5**
[NO] PPP TIMEOUT	IPNOS : CONFIG INTERFACE SERIAL(4/1) PPP# **no ppp timeout**

Default Value

None

Command Level

ppp

Related Commands

l2tp (l2tp commands)
pptp (pptp commands)

bundle

Create a Multilink PPP (MLPPP) bundle and go to the MLPPP command level. The MLPPP protocol (based on RFC 1990) allows packets to be fragmented and sent at the same time over multiple point-to-point links to the same remote address. Packets are encapsulated and then bundled before being sent to a VR. The following commands are available at this level.

- "admin-status" on page 14-10
- "display" on page 14-11
- "link" on page 14-12
- "max-links" on page 14-13
- "queue-size" on page 14-13
- "reorder-timeout" on page 14-14
- "site-name" on page 14-14

Syntax	*Example*
`[NO] INTERFACE BUNDLE <slotNum/bundleNum>`	`IPNOS: CONFIG#` **`interface bundle 3/1`** `IPNOS: CONFIG INTERFACE BUNDLE(3/1)#`

`[NO] INTERFACE BUNDLE <slotNum/bundleNum>`	`IPNOS: CONFIG#` **`no interface bundle 3/1`**
	`INTERFACE BUNDLE Object successfully deleted`
	
	`IPNOS: CONFIG#`

Default Value

None

Command Mode

config and show

Related Commands

None

admin-status

Enable administrative status of the MLPPP bundle. Disable the bundle before configuring it.

Syntax	*Example*
ADMIN-STATUS { ENABLE | DISABLE }	IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **admin-status enable** IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **display admin** ADMIN-STATUS : ENABLE
ADMIN-STATUS { ENABLE | DISABLE }	IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **admin-status disable** IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **display admin** ADMIN-STATUS : DISABLE

Default Value

Disabled

Command Level

bundle

Related Commands

None

display

View current configuration of the MLPPP bundle.

Syntax	*Example*
DISPLAY	IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **display** SLOT : 3 BUNDLE : 1 SITE-NAME : MAX-LINKS : 8 QUEUE-SIZE : 16 FRAGMENTS PER PACKET : 1 REORDER-TIMEOUT : 3000 TRANSMIT-MRRU : 1500 RECEIVE-MRRU : 1500 ADMIN-STATUS : DISABLE OPER-STATUS : DOWN ACTIVE-LINKS : 0 PACKETS-SENT : 0 PACKETS-RECEIVED : 0 BYTES-SENT : 0 BYTES-RECEIVED : 0 LINK(s) : List is empty IPNOS: CONFIG INTERFACE BUNDLE(3/1)#

Default Value

None

Command Level

bundle

Related Commands

None

link

Link PPP session to the MLPPP bundle. You need to disable the PPP interface before linking it to the bundle.

Syntax	*Example*
[NO] LINK <slot>/<port>[:<channel>[:<subif>]]	IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **link 3/1:3** BUNDLE: Object successfully set LINK 3/1:3:0: Object successfully set IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **display link** LINK(s) : 3/1:1 3/1:2 3/1:3
[NO] LINK <slot>/<port>[:<channel>[:<subif>]]	IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **no link 3/1:3** LINK 3/1:3:0: Object successfully deleted IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **display link** LINK(s) : 3/1:1 3/1:2

Default Value

None

Command Level

bundle

Related Commands

max-links

max-links

Set maximum number of PPP links in the bundle. Choose from 1 to 256 links.

Syntax	*Example*
MAX-LINKS <numLinks>	IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **max-links 20** IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **display max-links** MAX-LINKS : 20
NO MAX-LINKS	IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **no max-links** IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **display max-links** MAX-LINKS : 8

Default Value

8 links

Command Level

bundle

Related Commands

link

queue-size

Set resequencing queue size. Choose from 0 to 256.

Syntax	*Example*
QUEUE-SIZE <queueSize>	IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **queue-size 35** IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **display queue** QUEUE-SIZE : 35
NO QUEUE-SIZE	IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **no queue-size** IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **display queue** QUEUE-SIZE : 16

Default Value

16

Command Level

bundle

Related Commands

None

reorder-timeout

Set time limit for reordering fragmented packets. Choose from 10 to 10,000 milliseconds.

Syntax	*Example*
REORDER-TIMEOUT <reorderTimeout>	IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **reorder-timeout 5000** IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **display reorder-timeout** REORDER-TIMEOUT : 5000
NO REORDER-TIMEOUT	IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **no reorder-timeout** IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **display reorder-timeout** REORDER-TIMEOUT : 3000

Default Value

3000 milliseconds

Command Level

bundle

Related Commands

None

site-name

Assign a name to the MLPPP bundle.

Syntax	*Example*
SITE-NAME <name>	IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **site-name RedwoodCity** IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **display site** SITE-NAME : RedwoodCity
NO SITE-NAME	IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **no site-name** IPNOS: CONFIG INTERFACE BUNDLE(3/1)# **display site** SITE-NAME :

Default Value

None

Command Level

bundle

Related Commands

None

CHAPTER

15 VPN Commands

Use these commands to configure a virtual private network (VPN). This chapter describes the following commands:

Command	*Page*
create	15-2
delete	15-5
display	15-6
name	15-7
show	15-8
vr	15-11

create

Create a virtual router (VR). VR names are in standard IP address format, x.x.x.x, where x is a number from 0 to 255. By default, a new VR is assigned to the control blade.

Spread the workload of a VR across multiple blades by assigning their slot numbers and processor engine numbers. Assign a single trunk, access, or control blade to a VR, and then combine one or more processor blades.

Refer to the *CoSine IPSX 9000 Hardware Installation and Reference* for details about blades and processor engine cards. In Table 15-1 are guidelines for assigning blades to a new VR.

Table 15-1 Blade Combinations Allowed When Creating VRs

	Processor	***Control***	***Access***	***Trunk***
Processor	Yes, with a control, access or trunk blade	Yes	Yes	Yes
Control	Yes	No	No	No
Access	Yes	No	No	No
Trunk	Yes	No	No	No

Syntax	***Example***

Create a VR on the control blade: `[CREATE | DELETE] VR vrId [slot[/pe],[slot[/pe]],...]`	`IPNOS : CONFIG VPN(50)# create vr 5.5.5.5`
	`VPN(50): Object successfully set`
	
	`IPNOS : CONFIG VPN(50) VR(5.5.5.5)# commit`
	`Commit configuration`
	`SSH IPSEC Express IKE version 1.1.1. using library version 1.1.1`
	`Copyright 1997,1998 SSH Communications Security Ltd.`
	`VR(5.5.5.5): Object successfully set`
	
	`IPNOS : CONFIG VPN(50)# vr 5.5.5.5`
	`VPN(50): Object successfully set`
	
	`IPNOS : CONFIG VPN(50) VR(5.5.5.5)# display slot`
	`SLOT  :  (config)`
	`SLOT  : 2 (actual)`

Syntax	Example
Create a VR across multiple blades: [CREATE | DELETE] VR vrId [slot[/pe],[slot[/pe]],...]	IPNOS : CONFIG VPN(50)# **create vr 20.20.20.20 1,2,3,4,5** VPN(50): Object successfully set IPNOS : CONFIG VPN(50) VR(20.20.20.20)# **commit** Commit configuration SSH IPSEC Express IKE version 1.1.1. using library version 1.1.1 Copyright 1997,1998 SSH Communications Security Ltd. VR(20.20.20.20): Object successfully set IPNOS : CONFIG VPN(50)# **vr 20.20.20.20** VPN(50): Object successfully set IPNOS : CONFIG VPN(50) VR(20.20.20.20)# **display slot** SLOT : 1,2,3,4,5 (config) SLOT : 1 (actual)
Create a VR on a specific processor engine: [CREATE | DELETE] VR vrId [slot[/pe],[slot[/pe]],...]	IPNOS: CONFIG VPN(50)# **create vr 2.2.2.2 2/1** VPN(50): Object successfully set IPNOS: CONFIG VPN(50) VR(2.2.2.2)# **dis slot** SLOT : 2/1 (config) SLOT : (actual)
Incorrect VR name: CREATE VR [CREATE | DELETE] VR vrId [slot[/pe],[slot[/pe]],...]	IPNOS: CONFIG VPN(50)# **create vr fred** fred: Not a valid IP address IPNOS: CONFIG VPN(50)#
Failed attempt to assign two access blades to a VR: [CREATE | DELETE] VR vrId [slot[/pe],[slot[/pe]],...]	IPNOS: CONFIG VPN(50)# **create vr 20.20.20.20 2,4** VPN(50): Object successfully set IPNOS: CONFIG VPN(50) VR(20.20.20.20)# **commit** Commit configuration VR(20.20.20.20): Can NOT set object VIRTUAL ROUTER: Access or Control Blade can be specified only once IPNOS: CONFIG VPN(50)#

Default Value

None

Command Level

vpn

Related Commands

create vpn
create vi
delete vr
vr

delete

Remove aVR from a VPN. You cannot delete a VR while in the VR level.

Syntax	*Example*

`DELETE VR <vrId>`	`IPNOS : CONFIG VPN(2)# delete vr 3.3.3.3`
	`VPN(2): Object successfully set`
	`VR 3.3.3.3: Object successfully deleted`
	
	`IPNOS : CONFIG VPN(2)#`

Default Value

None

Command Level

vpn

Related Commands

create vr
delete vpn
delete vi
vr

display

View the current configuration of the VPN. You can also limit the display to only the items you want to see.

Syntax	*Example*
DISPLAY	IPNOS: CONFIG VPN(50)# **display** VPN : 50 NAME : VPN 50 STATUS : ENABLE VR(s): VR ID | VR NAME | CONFIG SLOT | OPER SLOT | STATE 1.1.1.1 | |2 | 2 | ENABLE IPNOS: CONFIG VPN(50)#
DISPLAY <argument>	IPNOS : CONFIG VPN(50)# **display status** STATUS : ENABLE

Default Value

None

Command Level

vpn

Related Commands

None

name

Provide a name for the current VPN. The name string you use must contain no spaces, but can include alphanumeric or special characters.

Syntax	*Example*
NAME <string>	IPNOS : CONFIG VPN(2)# **name CoSine2** IPNOS : CONFIG VPN(2)# **display** VPN : 2 NAME : CoSine2 STATUS : ENABLE VR(s): 2.2.2.2

Default Value

None

Command Level

vpn

Related Commands

sysname
name

show

Display the current VPN configurations, including VR and VI functions. The configurations are the ones already set by the **commit** command. Functions not committed are not displayed.

Syntax	*Example*
New VPN: SHOW CONFIGURATION	IPNOS: CONFIG# **vpn 50** IPNOS: CONFIG VPN(50)# **show** SHOW CONFIGURATION IPNOS: CONFIG VPN(50)# **show config** NAME VPN 50
Multiple new VRs: SHOW CONFIGURATION	IPNOS: CONFIG VPN(50)# **create vr 1.1.1.1** VPN(50): Object successfully set IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **com** Commit configuration SSH IPSEC Express IKE version 1.1.1. using library version 1.1.1 Copyright 1997,1998 SSH Communications Security Ltd. VR(1.1.1.1): Object successfully set IPNOS: CONFIG VPN(50)# **create vr 2.2.2.2 2** VPN(50): Object successfully set IPNOS: CONFIG VPN(50) VR(2.2.2.2)# **comm** Commit configuration VR(2.2.2.2): Object successfully set IPNOS: CONFIG VPN(50)# **create vr 3.3.3.3 4** VPN(50): Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3)# **comm** Commit configuration VR(3.3.3.3): Object successfully set IPNOS: CONFIG VPN(50)# **show config** NAME VPN 50 CREATE VPN 50 CREATE VR 1.1.1.1 COMMIT VR 1.1.1.1 VI 1

Syntax	Example
SHOW CONFIGURATION (continued)	IP ADDRESS 1.1.1.1 255.255.255.255 IP BROADCAST 0 COMMIT # VI 1 TELNET COMMIT QOS COMMIT COMMIT # VR 1.1.1.1 CONFIG # COMMIT VPN 50 CREATE VPN 50 CREATE VR 2.2.2.2 2 RADIUS-SERVER RETRANSMIT 5 COMMIT VR 2.2.2.2 VI 1 IP ADDRESS 2.2.2.2 255.255.255.255 IP BROADCAST 0 COMMIT # VI 1 TELNET COMMIT QOS COMMIT L2TP COMMIT # L2TP L2TP COMMIT # LNS L2TP COMMIT # LAC L2TP COMMIT # TUNNEL-SWITCH COMMIT # VR 2.2.2.2 CONFIG # COMMIT VPN 50 CREATE VPN 50 CREATE VR 3.3.3.3 4 RADIUS-SERVER RETRANSMIT 5 COMMIT VR 3.3.3.3 VI 1 IP ADDRESS 3.3.3.3 255.255.255.255 IP BROADCAST 0 COMMIT # VI 1 TELNET COMMIT QOS COMMIT L2TP COMMIT # L2TP

Syntax	*Example*
SHOW CONFIGURATION (continued)	L2TP COMMIT # LNS L2TP COMMIT # LAC L2TP COMMIT # TUNNEL-SWITCH COMMIT # VR 3.3.3.3 CONFIG # COMMIT VPN 50 IPNOS: CONFIG VPN(50)#

Default Value

None

Command Level

vpn

Related Commands

None

vr

Go to the virtual router (VR) command level. See Chapter 16, “VR Commands” for command information.

Syntax	*Example*

Go to the VR level:
`VR <vrId>`
	`IPNOS : config vpn(50)# vr 1.1.1.1`
	
	`IPNOS : config vpn(50) vr(1.1.1.1)#`

Default Value

None

Command Level

vpn

Related Commands

create vr
delete vr
vpn
vi

CHAPTER

16 VR Commands

Use these commands to configure a virtual router (VR). This chapter describes the following commands:

Command	*Page*
access-list	16-5
acl	16-7
display	16-8
dstaddr	16-9
dstport	16-11
protocol	16-11
srcaddr	16-12
srcport	16-14
tos	16-14
arp	16-16
certificate	16-17
certificate	16-18
display	16-19
flag	16-20
private-key	16-20
clear	16-22
create	16-24
crypto	16-25
decr-ttl	16-27
delete	16-28

Command	*Page*
dhcp-relay	16-29
display	16-37
dns	16-43
ds-byte	16-44
export-policy	16-46
acl	16-46
action	16-47
bgp-as	16-48
bgp-community	16-49
display	16-51
metric	16-52
peer-as	16-53
protocol	16-54
rip-ip	16-56
froip	16-57
fw	16-58
hostname	16-60
import-policy	16-61
acl	16-62
action	16-62
bgp-community	16-63
display	16-64
home-as	16-65
origin	16-65
peer-as	16-66
preference	16-67
ip	16-68
l2tp	16-78
display	16-80
l2tp	16-82
lns	16-91
ppp	16-93
tunnel	16-96

Command	Page
display	16-97
session	16-98
name	16-106
nbns	16-107
packet-filter	16-108
ping	16-112
pptp	16-113
display	16-114
pptp	16-115
qos	16-118
display	16-118
ds-byte-filter	16-119
radius-server	16-122
reassembly	16-128
router	16-129
service-selection	16-130
connection-profile	16-131
display	16-132
ip-local-pool-name	16-132
lns-address	16-133
vpn	16-134
vr	16-134
display	16-135
interface	16-136
service-selection-template	16-136
show	16-139
snmp-server	16-144
syslogd	16-151
telnet	16-157
display	16-157
enable	16-158
kill	16-159
show	16-160

Command	*Page*
timeout	16-160
traceroute	16-162
ttl	16-163
type	16-164
username	16-165
vi	16-167

access-list

Create an IP address list for dynamic network address translation (NAT). See "ip" on page 16-68 for NAT commands.

NOTE: The **access-list** command has no relation to the **acl** command. Refer to "acl" on page 16-7 for more information.

Table 16-1 describes the parameters available.

Table 16-1 NAT Access List Command Parameters

Parameter	*Description*
listId	A list identifier
PERMIT	Permit network access for the VI
start-address	Starting IP address range
end-address	Ending IP address range
viId	Name of the VI

Syntax	*Example*
ACCESS-LIST <listId> PERMIT <start-address> <end-address> <viId>	IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **access-list 1 permit 192.168.121.1 192.168.121.64 32** VR(1.1.1.1): Object successfully set ACCESS-LIST 1: Object successfully set IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **display access-list** ACCESS-LIST(s): ACCESS-LIST | VI | START-ADDRESS | END-ADDRESS | POOL 1 | 32 | 192.168.121.1 | 192.168.121.64 | 0

Syntax	*Example*
NO ACCESS-LIST <listId> <viId>	IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **display access-list** ACCESS-LIST(s): ACCESS-LIST | VI | START-ADDRESS | END-ADDRESS | POOL 1 | 32 | 192.168.121.1 | 192.168.121.64 | 0 2 | 40 | 202.168.48.1 | 202.168.48.32 | 0 IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **no access-list 2 40** VR(1.1.1.1): Object successfully set ACCESS-LIST = 2, VI = 40: Object successfully deleted IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **display access-list** ACCESS-LIST(s): ACCESS-LIST | VI | START-ADDRESS | END-ADDRESS | POOL 1 | 32 | 192.168.121.1 | 192.168.121.64 | 0 IPNOS: CONFIG VPN(50) VR(1.1.1.1)#

Default Value

None

Command Level

VR

Related Command

vr ip

acl

Create or modify a general access control list (ACL). An ACL classifies a set of packets for a feature such as firewall or IPSec or routing. This command also puts you in the ACL level. The following commands are available.

- "display" on page 16-8
- "dstaddr" on page 16-9
- "dstport" on page 16-11
- "protocol" on page 16-11
- "srcaddr" on page 16-12
- "srcport" on page 16-14
- "tos" on page 16-14

NOTE: The **acl** command has no relationship to the **access-list** command. Refer to "access-list" on page 16-5.

Syntax	*Example*
ACL <aclID>	IPNOS : CONFIG VPN(50) VR(1.1.1.1)# **acl 1** 1.1.1.10: Object successfully set IPNOS : CONFIG VPN(50) VR(1.1.1.1) ACL(1)#
NO ACL <aclID>	IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **no acl 1** VR(1.1.1.1): Object successfully set ACL 1: Object successfully deleted IPNOS: CONFIG VPN(50) VR(1.1.1.1)#

Default Value

None

Command Level

VR

Related Commands

access-list
crypto
export-policy
fw

import-policy
packet-filter
router

display

View the current ACL configuration.

Syntax	*Example*
DISPLAY	IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **display** VPN : 50 VR : 1.1.1.1 ACL : 1 SRCADDR : ALL DSTADDR : ALL PROTOCOL : * TOS : * SRCPORT : * DSTPORT : *
DISPLAY <argument>	IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **display acl** ACL : 1

Default Value

None

Command Level

ACL

Related Commands

None

dstaddr

Set the destination IP address and subnet mask. Use this command for setting routing policy destinations. Table 16-2 describes the special function of the **dstaddr** command when used for a specific feature.

Table 16-2 Special Functions of the dstaddr Command

Feature	*IP Address*	*SubNet Mask*	*Description*
IPSec	any	255.255.255.255	Use a 32-bit subnet mask (255.255.255.255) if you want the ACL to reference one specific host, not a subnet of hosts. This is used in ipsec-isakmp and ipsec-dialup crypto maps to indicate the ISAKMP Phase II ID is one host.
BGP	0.0.0.0	0.0.0.0	Default route for Import and Export policies.
BGP	0.0.0.0	255.255.255.255	Any route for Import and Export policies.
All routing policies	any	any	Routing summary for all routing policies.

Syntax	*Example*

`DSTADDR {ip-address mask | ALL | DEFAULT}`	`IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# dstaddr 192.128.201.23 255.255.255.0`
	
	`IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# display`
	`VPN : 50`
	`VR : 1.1.1.1`
	`ACL : 1`
	`SRCADDR : ALL`
	`DSTADDR : 192.128.201.23|255.255.255.0`
	`PROTOCOL : *`
	`TOS : *`
	`SRCPORT : *`
	`DSTPORT : *`

Syntax	*Example*
DSTADDR {ip-address mask | ALL | DEFAULT}	IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **dstaddr all** IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **display** VPN : 50 VR : 1.1.1.1 ACL : 1 SRCADDR : ALL DSTADDR : ALL PROTOCOL : * TOS : * SRCPORT : * DSTPORT : * IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **commit** Commit configuration ACL(1): Object successfully set IPNOS: CONFIG VPN(50) VR(1.1.1.1)#
DSTADDR {ip-address mask | ALL | DEFAULT}	IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **dstaddr default** IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **display** VPN : 50 VR : 1.1.1.1 ACL : 1 SRCADDR : ALL DSTADDR : DEFAULT PROTOCOL : * TOS : * SRCPORT : * DSTPORT : * IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **commit** Commit configuration ACL(1): Object successfully set IPNOS: CONFIG VPN(50) VR(1.1.1.1)#

Default Value

All addresses

Command Level

ACL

Related Commands

dstport

dstport

Define the range of destination ports to be used.

Syntax	*Example*
DSTPORT <start-port> <end-port>	IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **dstport 200 300** IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **display dstport** DSTPORT : 200 - 300
DSTPORT <start-port> <end-port>	IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **dstport *** IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **display dstport** DSTPORT : *

Default Value

Wildcard (denoted as *)

Command Level

ACL

Related Commands

dstaddr

protocol

Identify the next level protocol. Refer to the Internet Assigned Numbers Authority (IANA) website http://www.iana.org/ for a list of protocol numbers. Type an asterisk (*) to denote all protocol values.

Syntax	*Example*
Set L2TP Protocol number: PROTOCOL <protocol-Id>	IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **protocol 115** IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **display protocol** PROTOCOL : 115 IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **commit** Commit configuration ACL(1): Object successfully set IPNOS: CONFIG VPN(50) VR(1.1.1.1)#

Syntax	*Example*

Set for any protocol: `PROTOCOL *`	`IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# protocol *`
	
	`IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# display protocol`
	`PROTOCOL  : *`
	
	`IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# commit`
	`Commit configuration`
	`ACL(1): Object successfully set`
	
	`IPNOS: CONFIG VPN(50) VR(1.1.1.1)#`

Default Value

Wildcard (denoted as *)

Command Level

ACL

Related Commands

None

srcaddr

Set the source IP address and subnet mask. To set a wildcard subnet mask, type an asterisk (*). Use ALL when setting up the ACL for routing. To set ISAKMP Phase II to reference one specific host, use a 32-bit subnet mask (255.255.255.255). See Chapter 17, “Crypto Commands” for details.

Syntax	Example
SRCADDR {ip-address mask| ALL | DEFAULT}	IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **srcaddr 192.128.201.15 255.255.255.0** IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **display** VPN : 50 VR : 1.1.1.1 ACL : 1 SRCADDR : 192.128.201.15|255.255.255.0 DSTADDR : DEFAULT PROTOCOL : * TOS : * SRCPORT : * DSTPORT : 200 - 300
SRCADDR {ip-address mask| ALL | DEFAULT}	IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **srcaddr all** IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **display** VPN : 50 VR : 1.1.1.1 ACL : 1 SRCADDR : ALL DSTADDR : DEFAULT PROTOCOL : * TOS : * SRCPORT : * DSTPORT : 200 - 300
SRCADDR {ip-address mask| ALL | DEFAULT}	IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **srcaddr default** IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **display** VPN : 50 VR : 1.1.1.1 ACL : 1 SRCADDR : DEFAULT DSTADDR : DEFAULT PROTOCOL : * TOS : * SRCPORT : * DSTPORT : 200 - 300

Default Value

All addresses

Command Level

ACL

Related Commands

srcport

srcport

Define the range of source ports.

Syntax	*Example*
SRCPORT <start-port> <end-port>	IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **srcport 100 200** IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **display srcport** SRCPORT : 100 - 200

Default Value

Wildcard (denoted as *)

Command Level

ACL

Related Commands

srcaddr

tos

Set the type of service (TOS) value. Refer to RFC 1349. This command affects only the incoming packets from IP addresses specified in the ACL. In contrast, the **ds-byte** command affects all outgoing packets from the VR. See "ds-byte" on page 16-44. The following is a list of possible values.

- 0 (normal service)
- 1 (minimize monetary cost)
- 2 (maximize reliability)
- 3 (maximize throughput)
- 4 (minimize delay)

Syntax	*Example*
TOS <tos-Id>	IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **tos 2** IPNOS: CONFIG VPN(50) VR(1.1.1.1) ACL(1)# **display** VPN : 50 VR : 1.1.1.1 ACL : 1 SRCADDR : DEFAULT DSTADDR : DEFAULT PROTOCOL : * TOS : 2 SRCPORT : 100 - 200 DSTPORT : 200 - 300

Default Value

Wildcard (denoted as *)

Command Level

ACL

Related Commands

ds-byte (VR command)

arp

Add or remove a static entry to the Address Resolution Protocol (ARP) cache. This cache provides media access control (MAC) addresses for all devices known to the router. All device addresses obtained from each router are added to the cache. Table 16-3 describes the **arp** command parameters. Use the **no** option to remove an entry.

Table 16-3 ARP Command Parameters

Parameter	*Description*
No	Remove entry from ARP cache
ip-address	IP address in 1.1.1.1 format
hw-address	Device address in 1a:2b:3c:4d:5e:6f format
type	Type of entry. The default value is the ARPA framing format
viId	VI name

Syntax	*Example*
`[NO] ARP <ip-address> <hw-address> <type> <viId>`	`IPNOS : CONFIG VPN(50) vr(1.1.1.10)# arp 10.2.1.2 1a:2b:3c:4d:5e:6f arpa 1`

Default Value

None

Command Level

VR

Related Commands

clear

certificate

Create a digital certificate. Certificates are used in ISAKMP Phase I to set up a secure network connection between two gateways. Private keys are generated on both hosts and exchanged via certificates. Refer to "Crypto Commands" in Chapter 17 and especially "authentication" on page 17-21 for more information.

NOTE: Certificates require digital signatures. Refer to "authentication" on page 17-21 for information on setting digital signatures in the ISAKMP policy.

This command puts you in the certificate level. The following commands are available in the certificate level:

- "certificate" on page 16-18
- "display" on page 16-19
- "flag" on page 16-20
- "private-key" on page 16-20

Syntax	*Example*
CERTIFICATE <cert-name>	IPNOS: CONFIG VPN(50) VR(2.2.2.2)# **certificate cert1** VR(2.2.2.2): Object successfully set IPNOS: CONFIG VPN(50) VR(2.2.2.2) CERTIFICATE(cert1)#
NO CERTIFICATE <cert-name>	IPNOS: CONFIG VPN(50) VR(2.2.2.2)# **no certificate cert1** VR(2.2.2.2): Object successfully set CERTIFICATE cert1 default: Object successfully deleted

Default Value

None

Command Level

VR

Related Commands

None

certificate

Set the public internet key exchange (IKE).

NOTE: Because of the length of the certificate, it is impractical to type it in manually. You should use InVision™ SMS or a telnet script to automatically enter the certificate.

Syntax	*Example*
CERTIFICATE <PEM-format-key>	IPNOS : CONFIG VPN(50) VR(10.1.1.1) CERTIFICATE(cert1 joemap1)# **cert 1A2B3C4D5E6F71829304A5B6C7D8E9F1A2B3C4D5E6F71829304A5B6C7D 8E9F1A2B3C4D5E6F71829304A5B6C7D8E9F1A2B3C4D5E6F71829304A5B 6C7D8E9F1A2B3C4D5E6F71829304A5B6C7D8E9F** IPNOS : CONFIG VPN(50) VR(10.1.1.1) CERTIFICATE(cert1 joemap1)# **display** VPN : 1 VR : 10.1.1.1 NAME : cert1 MAP : default FLAG : NO-CRL CERTIFICATE : 1A2B3C4D5E6F71829304A5B6C7D8E9F1A2B3C4D5E6F71829304A5B6C7D 8E9F1A2B3C4D5E6F71829304A5B6C7D8E9F1A2B3C4D5E6F71829304A5B 6C7D8E9F1A2B3C4D5E6F71829304A5B6C7D8E9F PRIVATE-KEY :

Default Value

None

Command Level

certificate

Related Commands

None

display

View the current certificate configuration.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG VPN(50) VR(10.1.1.1) CERTIFICATE(cert1 joemap1)# **flag no-crl** IPNOS : CONFIG VPN(50) VR(10.1.1.1) CERTIFICATE(cert1 joemap1)# **display** VPN : 1 VR : 10.1.1.1 NAME : cert1 MAP : default FLAG : NO-CRL CERTIFICATE : 1A2B3C4D5E6F71829304A5B6C7D8E9F1A2B3C4D5E6F71829304A5B6C7D8E9F1A2B3C4D5E6F71829304A5B6C7D8E9F1A2B3C4D5E6F71829304A5B6C7D8E9F PRIVATE-KEY :

Default Value

None

Command Level

certificate

Related Commands

show

flag

Set a security flag to ignore carriage returns and line feeds in the certificate.

Syntax	*Example*
FLAG {NO_CRL}	IPNOS : CONFIG VPN(50) VR(10.1.1.1) CERTIFICATE(cert1 joemap1)# **flag no-crl** IPNOS : CONFIG VPN(50) VR(10.1.1.1) CERTIFICATE(cert1 joemap1)# **display** VPN : 1 VR : 10.1.1.1 NAME : cert1 MAP : default FLAG : NO-CRL CERTIFICATE : 1A2B3C4D5E6F71829304A5B6C7D8E9F1A2B3C4D5E6F71829304A5B6C7D8E9F1A2B3C4D5E6F71829304A5B6C7D8E9F1A2B3C4D5E6F71829304A5B6C7D8E9F PRIVATE-KEY :

Default Value

None

Command Level

certificate

Related Commands

None

private-key

Set a private key.

NOTE: Because of the length of the private key, it is impractical to type it in manually. You should either use InVision SMS or a telnet script to automatically enter the private key.

Syntax	*Example*
PRIVATE-KEY <PEM-format-key>	IPNOS : CONFIG VPN(50) VR(10.1.1.1) CERTIFICATE(cert1 joemap1)# **private-key 1F2E3DC4B5A6079889706A5B4C3D2E1F1F2E3DC4B5A6079889706A5B4C3D2E1F1F2E3DC4B5A6079889706A5B4C3D2E1F1F2E3DC4B5A6079889706A5B4C3D2E1F1F2E3DC4B5A6079889706A5B4C3D2E1F** IPNOS : CONFIG VPN(50) VR(10.1.1.1) CERTIFICATE(cert1 joemap1)# **display** VPN : 1 VR : 10.1.1.1 NAME : cert1 MAP : default FLAG : NO-CRL CERTIFICATE : 1A2B3C4D5E6F71829304A5B6C7D8E9F1A2B3C4D5E6F71829304A5B6C7D8E9F1A2B3C4D5E6F71829304A5B6C7D8E9F1A2B3C4D5E6F71829304A5B6C7D8E9F1A2B3C4D5E6F71829304A5B6C7D8E9F PRIVATE-KEY : 1F2E3DC4B5A6079889706A5B4C3D2E1F1F2E3DC4B5A6079889706A5B4C3D2E1F1F2E3DC4B5A6079889706A5B4C3D2E1F1F2E3DC4B5A6079889706A5B4C3D2E1F1F2E3DC4B5A6079889706A5B4C3D2E1F

Default Value

None

Command Level

certificate

Related Commands

None

clear

Reset the following items:

- ARP cache
- Border Gateway Protocol (BGP) route connections
- DHCP relay statistics
- System-to-Intermediate System (IS-IS) information
- VR and VI statistics counters

Syntax	*Example*
CLEAR ARP-CACHE	IPNOS: CONFIG VPN(50) VR(10.10.10.1)# **clear arp** VR(3.3.3.3): Object successfully set ARP-CACHE DYNAMIC ENTRY: Object successfully deleted
CLEAR IP BGP {address|group -name}	IPNOS: CONFIG VPN(50) VR(10.10.10.1)# **clear ip bgp 172.16.24.2** VR(3.3.3.3): Object successfully set
CLEAR IP BGP {address|group -name}	IPNOS: CONFIG VPN(50) VR(10.10.10.1)# **clear ip bgp cosine** VR(3.3.3.3): Object successfully set
CLEAR COUNTER	IPNOS: CONFIG VPN(50) VR(10.10.10.1)# **clear counter** COUNTER: Vr Counter successfully cleared COUNTER: Vi 1 - Counter successfully cleared COUNTER: Vi 2 - Counter successfully cleared
CLEAR ISIS ALL	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **clear isis all** VR(5.5.5.5): Object successfully set Everything cleared!!
CLEAR ISIS ADJACENCY	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **clear isis adjacency** VR(5.5.5.5): Object successfully set Adjacency cleared!!
CLEAR ISIS DATABASE	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **clear isis database** VR(5.5.5.5): Object successfully set Isis Database cleared!!
CLEAR ISIS STATISTICS	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **clear isis statistics** VR(5.5.5.5): Object successfully set Statistics cleared!!

Default Value

None

Command Level

VR

Related Commands

arp

create

Create a VI for this VR. This command puts you in the VI level. See Chapter 21, “VI Commands” for a list of commands.

Syntax	*Example*

`CREATE VI <viId>`	`IPNOS : CONFIG VPN(50) VR(10.10.10.1)# create vi 5`
	`VR(10.10.10.1): Object successfully set`
	
	`IPNOS : CONFIG VPN(50) VR(10.10.10.1) VI(5)#`

Default Value

None

Command Level

VR

Related Commands

create vpn
create vr

crypto

Go to the Internet Protocol Security (IPSec) environment. This command will put you into one of several command levels:

- **IPSec Transform Set** configures encryption and authentication protocols and transportation modes. Refer to "ipsec" on page 17-3.
- **ISAKMP Parameter** configures the Internet Security Association Key Management Protocol (ISAKMP) server settings. Refer to the commands "isakmp" on page 17-8 and "parameter" on page 17-8.
- **ISAKMP Policy** configures the key management guidelines. Refer to "isakmp" on page 17-8 and "policy" on page 17-20.
- **IPSec Dialup Crypto Map** combines IPSec transform sets and ISAKMP policies with access control lists (ACLs) to define a secure network connection for dialup customers. Refer to "ipsec-dialup" on page 17-30.
- **IPSec ISAKMP Crypto Map** combines IPSec transform sets and ISAKMP policies with access control lists (ACLs) to define a secure network connection. Refer to "ipsec-isakmp" on page 17-39.
- **IPSec Manual Crypto Map** manually configures a secure network connection for testing purposes. Refer to "ipsec-manual" on page 17-46.

See "Crypto Commands" in Chapter 17 for a list of commands. Also, refer to the *CoSine IPNOS CLI Configuration Guide* for detailed information about setting up a secure VPN. Use the **no** option to delete a transform-set, ISAKMP policy or crypto map.

NOTE: You must complete the ACL configuration before setting up the cryptography elements.

Syntax	*Example*
[NO] CRYPTO IPSEC TRANSFORM-SET <set-name> {<transform1>} [<transform2>] [<transform3>]	IPNOS : CONFIG VPN(50) VR(10.10.10.1)# **crypto ipsec transform-set test espdes** VR(10.10.10.1): Object successfully set IPNOS : CONFIG VPN(50) VR(10.10.10.1) TRANSFORM-SET(test)#

Syntax	*Example*
CRYPTO ISAKMP PARAMETER	IPNOS : CONFIG VPN(50) VR(10.10.10.1)# **crypto isakmp parameter** VR(10.10.10.1): Object successfully set IPNOS : CONFIG VPN(50) VR(10.10.10.1) ISAKMP PARAMETER#
NO CRYPTO ISAKMP POLICY <policy-name> <priority>	IPNOS : CONFIG VPN(50) VR(10.10.10.1)# **crypto isakmp policy joenet 1** VR(10.1.1.1): Object successfully set IPNOS : CONFIG VPN(50) VR(10.10.10.1) POLICY(joenet 1)#
[NO] CRYPTO MAP <map-name> <priority> IPSEC-MANUAL	IPNOS : CONFIG VPN(50) VR(10.10.10.1)# **crypto map map1 1 ipsec-manual** VR(10.10.10.1): Object successfully set IPNOS : CONFIG VPN(50) VR(10.10.10.1) MAP IPSEC-MANUAL(map1 1)#
[NO] CRYPTO MAP <map-name> <priority> IPSEC-ISAKMP	IPNOS : CONFIG VPN(50) VR(172.16.16.1)# **crypto map map1 1 ipsec-isakmp** VR(172.16.16.1): Object successfully set IPNOS : CONFIG VPN(50) VR(172.16.16.1) MAP IPSEC-ISAKMP(map1 1)#

Default Value

None

Command Level

VR

Related Commands

None

decr-ttl

Enable or disable the decrement time to live (TTL) counter.

Syntax	*Example*
DECR-TTL {ENABLE | DISABLE}	IPNOS : CONFIG VPN(50) VR(2.2.2.2)# **decr-ttl enable** IPNOS : CONFIG VPN(50) VR(2.2.2.2)# **display decr-ttl** DECR-TTL : ENABLE
DECR-TTL {ENABLE | DISABLE}	IPNOS : CONFIG VPN(50) VR(2.2.2.2)# **decr-ttl disable** IPNOS : CONFIG VPN(50) VR(2.2.2.2)# **display decr-ttl** DECR-TTL : DISABLE

Default Value

Enable

Command Level

VR

Related Commands

ttl

delete

Remove a VI from the VR.

NOTE: You cannot delete a VI while you are in the VI level.

Syntax	*Example*
DELETE VI <viId>	IPNOS : CONFIG VPN(50) VR(10.10.10.1)# **del vi 5** VR(10.10.10.1): Object successfully set VI 5: Object successfully deleted IPNOS : CONFIG VPN(50) VR(10.10.10.1)#

Default Value

None

Command Level

VR

Related Commands

delete vpn
delete vr

dhcp-relay

Configure the dynamic host configuration protocol (DHCP) relay agent. It is used to allow PCs and other devices to acquire DHCP parameters from a remote server. You can assign multiple DHCP servers. Table 16-4 describes the parameters available.

NOTE: Use **clear counters** command to clear the DHCP relay statistics. Refer to "clear" on page 16-22 for more information.

Table 16-4 DHCP Relay Parameters

Setting	*Definition*
Server	The IP address of any available DHCP servers. More than one DHCP server may be added to the list. Use the **no** option to delete a server from the list.
Maxhops	The maximum number of hops that the relay agent will allow for operation with a DHCP server. When the hop count of the DHCP/bootp packets exceeds this number, the relay agent will not forward them to servers configured. Choose from 1 to 16 hops. The default value is 4 hops. Use the **no** option to reset the default value.
Threshold	If multiple DHCP servers are available, the DHCP threshold feature allows clients to attempt to acquire IP addresses and other information from the closest server that has IP address leases available. See RFC 1533 for details. Choose from 0 to 300 seconds. The default value is 4 seconds. Use the **no** option to reset the default value.
Status	Use the keyword **enable** to activate the DHCP relay agent, or **disable** to stop its operation.

Syntax	*Example*
[NO] DHCP-RELAY SERVER <ip-address> [<name>]	IPNOS : CONFIG VPN(50) VR(2.2.2.1)# **dhcp server 192.128.201.4** VR(2.2.2.1): Object successfully set DHCP-RELAY SERVER 192.128.201.4: Object successfully set IPNOS : CONFIG VPN(50) VR(2.2.2.1)# **display dhcp** DHCP-RELAY(s): MAXHOPS : 4 THRESHOLD : 4 ADMIN-STATUS : DISABLE OPER-STATUS : DOWN DHCP-RELAY SERVER(s): ADDRESS | NAME 192.128.201.4 DHCP Relay Statistics: Recv'd Discover Packets 0 Recv'd Offer Packets 0 Recv'd Ack Packets 0 Recv'd Nak Packets 0 Recv'd Request Packets 0 Recv'd Release Packets 0 Recv'd Decline Packets 0 Out Discover Packets 0 Out Offer Packets 0 Out Ack Packets 0 Out Nak Packets 0 Out Request Packets 0 Out Release Packets 0 Out Decline Packets 0

Syntax	Example
[NO] DHCP-RELAY SERVER <ip-address> [<name>]	IPNOS : CONFIG VPN(50) VR(2.2.2.1)# **dhcp server 201.120.105.12 dhcpserv1** VR(2.2.2.1): Object successfully set DHCP-RELAY SERVER 201.120.105.12: Object successfully set IPNOS : CONFIG VPN(50) VR(2.2.2.1)# **display dhcp** DHCP-RELAY(s): MAXHOPS : 4 THRESHOLD : 4 ADMIN-STATUS : DISABLE OPER-STATUS : DOWN DHCP-RELAY SERVER(s): ADDRESS | NAME 192.128.201.4 201.120.105.12 | dhcpserv1 DHCP Relay Statistics: Recv'd Discover Packets 0 Recv'd Offer Packets 0 Recv'd Ack Packets 0 Recv'd Nak Packets 0 Recv'd Request Packets 0 Recv'd Release Packets 0 Recv'd Decline Packets 0 Out Discover Packets 0 Out Offer Packets 0 Out Ack Packets 0 Out Nak Packets 0 Out Request Packets 0 Out Release Packets 0 Out Decline Packets 0

Syntax	*Example*
[NO] DHCP-RELAY SERVER <ip-address> [<name>]	IPNOS : CONFIG VPN(50) VR(2.2.2.1)# **no dhcp server 192.128.201.4** VR(2.2.2.1): Object successfully set DHCP-RELAY SERVER 192.128.201.4: Object successfully deleted IPNOS : CONFIG VPN(50) VR(2.2.2.1)# **display dhcp** DHCP-RELAY(s): MAXHOPS : 4 THRESHOLD : 4 ADMIN-STATUS : DISABLE OPER-STATUS : DOWN DHCP-RELAY SERVER(s): ADDRESS | NAME 201.120.105.12 | dhcpserv1 DHCP Relay Statistics: Recv'd Discover Packets 0 Recv'd Offer Packets 0 Recv'd Ack Packets 0 Recv'd Nak Packets 0 Recv'd Request Packets 0 Recv'd Release Packets 0 Recv'd Decline Packets 0 Out Discover Packets 0 Out Offer Packets 0 Out Ack Packets 0 Out Nak Packets 0 Out Request Packets 0 Out Release Packets 0 Out Decline Packets 0

Syntax	*Example*
[NO] DHCP-RELAY MAXHOPS {1..16}	IPNOS : CONFIG VPN(50) VR(2.2.2.1)# **dhcp maxhops 10** VR(2.2.2.1): Object successfully set DHCP-RELAY MAXHOPS 10: Object successfully set IPNOS : CONFIG VPN(50) VR(2.2.2.1)# **display dhcp** DHCP-RELAY(s): MAXHOPS : 10 THRESHOLD : 4 ADMIN-STATUS : DISABLE OPER-STATUS : DOWN DHCP-RELAY SERVER(s): ADDRESS | NAME 201.120.105.12 | dhcpserv1 DHCP Relay Statistics: Recv'd Discover Packets 0 Recv'd Offer Packets 0 Recv'd Ack Packets 0 Recv'd Nak Packets 0 Recv'd Request Packets 0 Recv'd Release Packets 0 Recv'd Decline Packets 0 Out Discover Packets 0 Out Offer Packets 0 Out Ack Packets 0 Out Nak Packets 0 Out Request Packets 0 Out Release Packets 0 Out Decline Packets 0

Syntax	*Example*
[NO] DHCP-RELAY THRESHOLD {0..300}	IPNOS : CONFIG VPN(50) VR(2.2.2.1)# **dhcp threshold 50** VR(2.2.2.1): Object successfully set DHCP-RELAY THRESHOLD 50: Object successfully set IPNOS : CONFIG VPN(50) VR(2.2.2.1)# **display dhcp** DHCP-RELAY(s): MAXHOPS : 10 THRESHOLD : 50 ADMIN-STATUS : DISABLE OPER-STATUS : DOWN DHCP-RELAY SERVER(s): ADDRESS | NAME 201.120.105.12 | dhcpserv1 DHCP Relay Statistics: Recv'd Discover Packets 0 Recv'd Offer Packets 0 Recv'd Ack Packets 0 Recv'd Nak Packets 0 Recv'd Request Packets 0 Recv'd Release Packets 0 Recv'd Decline Packets 0 Out Discover Packets 0 Out Offer Packets 0 Out Ack Packets 0 Out Nak Packets 0 Out Request Packets 0 Out Release Packets 0 Out Decline Packets 0

Syntax	*Example*
DHCP-RELAY STATUS {ENABLE | DISABLE}	IPNOS : CONFIG VPN(50) VR(2.2.2.1)# **dhcp status enable** VR(2.2.2.1): Object successfully set DHCP-RELAY STATUS enable: Object successfully set IPNOS : CONFIG VPN(50) VR(2.2.2.1)# **display dhcp** DHCP-RELAY(s): MAXHOPS : 10 THRESHOLD : 50 ADMIN-STATUS : ENABLE OPER-STATUS : UP DHCP-RELAY SERVER(s): ADDRESS | NAME 201.120.105.12 | dhcpserv1 DHCP Relay Statistics: Recv'd Discover Packets 0 Recv'd Offer Packets 0 Recv'd Ack Packets 0 Recv'd Nak Packets 0 Recv'd Request Packets 0 Recv'd Release Packets 0 Recv'd Decline Packets 0 Out Discover Packets 0 Out Offer Packets 0 Out Ack Packets 0 Out Nak Packets 0 Out Request Packets 0 Out Release Packets 0 Out Decline Packets 0

Syntax	*Example*
DHCP-RELAY STATUS {ENABLE | DISABLE}	IPNOS : CONFIG VPN(50) VR(2.2.2.1)# **dhcp status disable** VR(2.2.2.1): Object successfully set DHCP-RELAY STATUS disable: Object successfully set IPNOS : CONFIG VPN(50) VR(2.2.2.1)# **display dhcp** DHCP-RELAY(s): MAXHOPS : 10 THRESHOLD : 50 ADMIN-STATUS : DISABLE OPER-STATUS : DOWN DHCP-RELAY SERVER(s): ADDRESS | NAME 201.120.105.12 | dhcpserv1 DHCP Relay Statistics: Recv'd Discover Packets 0 Recv'd Offer Packets 0 Recv'd Ack Packets 0 Recv'd Nak Packets 0 Recv'd Request Packets 0 Recv'd Release Packets 0 Recv'd Decline Packets 0 Out Discover Packets 0 Out Offer Packets 0 Out Ack Packets 0 Out Nak Packets 0 Out Request Packets 0 Out Release Packets 0 Out Decline Packets 0

Default Value

See Table 16-4

Command Level

VR

Related Commands

None

display

View a VR's current configuration. Used without arguments, it displays all of the router's parameters. Used with a specific parameter, it displays the status of that parameter.

Syntax	*Example*

DISPLAY	`IPNOS: CONFIG VPN(50) VR(7.7.7.7)# display`
	`VPN : 50`
	`VR : 7.7.7.7`
	`SLOT : 2 (config)`
	`SLOT : 2/0,2/2 (actual)`
	`TYPE : CUSTOMER`
	`NAME :`
	`TTL : 255`
	`DECR-TTL : ENABLE`
	`DS-BYTE : DISABLE`
	`PACKET-FILTER DENY EVENT-LOG: DISABLE`
	`ROUTING : IP ROUTING`
	`REASSEMBLY : 5`
	`STATUS : ENABLE`
	`PPTP : DISABLE`
	`SNMP-SERVER : DISABLE`
	`TRAPS : DISABLE`
	`MPLS : ENABLE`
	`SYSLOGD : DISABLE`
	`SYSLOGD HOST : 0.0.0.0`
	`FACILITY 1 : 0`
	`FACILITY 2 : 0`
	`FACILITY 3 : 0`
	`FACILITY 4 : 0`
	`FACILITY 5 : 0`
	`FACILITY 6 : 0`
	`FACILITY 7 : 0`
	`FACILITY 8 : 0`
	`FACILITY 9 : 0`
	`FACILITY 10 : 0`
	`FACILITY 11 : 0`
	`LEVEL 1 : 0`
	`LEVEL 2 : 0`
	`LEVEL 3 : 0`
	`LEVEL 4 : 0`
	`LEVEL 5 : 0`
	`HOSTNAME :`
	`DNS PRIMARY : 0.0.0.0`
	`DNS SECONDARY : 0.0.0.0`

Syntax	Example
DISPLAY (cont.)	

```
WINS PRIMARY     : 0.0.0.0
WINS SECONDARY   : 0.0.0.0
AOL-PROXY        : DISABLE    (Active Session(s): 0)
CS-PROXY         : DISABLE    (Active Session(s): 0)
FTP-PROXY        : DISABLE    (Active Session(s): 0)
HTTP-PROXY       : DISABLE    (Active Session(s): 0)
LDAP-PROXY       : DISABLE    (Active Session(s): 0)
LNOTES-PROXY     : DISABLE    (Active Session(s): 0)
NETM-PROXY       : DISABLE    (Active Session(s): 0)
NNTP-PROXY       : DISABLE    (Active Session(s): 0)
PLUG-PROXY       : DISABLE    (Active Session(s): 0)
POP3-PROXY       : DISABLE    (Active Session(s): 0)
RSH-PROXY        : DISABLE    (Active Session(s): 0)
SMTP-PROXY       : DISABLE    (Active Session(s): 0)
SQL-PROXY        : DISABLE    (Active Session(s): 0)
TELNET-PROXY     : DISABLE    (Active Session(s): 0)

FW EVENT-LOG:
LEVEL            : MAJOR

VI(s):

VI ID| IP ADDRESS/MASK              | CONNECTED-TO
| STATE
1    | 7.7.7.7/255.255.255.255       | LOOPBACK
| UP/ENABLE
2    | 172.24.32.2/255.255.255.0     | V 50/6.6.6.6/2
| UP/ENABLE

FRoIP(s):       Disabled

ROUTE(s):
PREFIX   | NETMASK   | HOP   | VI   | TOS   | METRIC | AGE   | OWNER
7.7.7.7 | 255.255.255.255 | 0.0.0.0 | 1 | 0 | 1 | 697   | LOCAL
172.24.32.0 | 255.255.255.0 | 0.0.0.0 | 2 | 0 | 1 | 647 | LOCAL
```

Syntax	Example
DISPLAY (cont.)	

```
IP Statistics:
Recv'd Packets                    0
Recv'd Hdr Errors                 0
Recv'd Addr Errors                0
Forwarded Packets                 0
Recv'd Unknown Protocol           0
Discarded Packets                 0
Delivered Packets                 0
Out Requests                      0
Out Discards                      0
Out No Routes                     0
Require Reassembly                0
Successful Reassembly             0
Failed Reassembly                 0
Successful Fragmentation          0
Failed Fragmentation              0
Created Fragments                 0

ARP(s):
List is empty

ACCESS-LIST(s):
List is empty

POOL(s):
List is empty

IP STATIC NAT(s):
List is empty

IP NAT TRANSLATION(s):
List is empty

ACL(s):
List is empty
```

Syntax	Example

DISPLAY (cont.)	`CRYPTO TRANSFORM-SET(s):`
	`List is empty`
	
	`ISAKMP POLICY(s):`
	`List is empty`
	
	`CRYPTO MAP IPSEC-MANUAL(s):`
	`List is empty`
	
	`CRYPTO MAP IPSEC-ISAKMP(s):`
	`List is empty`
	
	`ISAKMP PARAMETER(s):`
	`List is empty`
	
	`CERTIFICATE(s):`
	`List is empty`
	
	`COMMUNITY(s):`
	`List is empty`
	
	`TRAPS HOST(s):`
	`List is empty`
	
	`IMPORT-POLICY(s):`
	`List is empty`
	
	`EXPORT-POLICY(s):`
	`List is empty`
	
	`PACKET-FILTER(s):`
	`List is empty`
	
	`PPP USERNAME(s):`
	`List is empty`
	
	`LOCAL POOL(s):`
	`List is empty`

Syntax	Example
DISPLAY (cont.)	DS-BYTE-FILTER(s): List is empty RADIUS-SERVER CONFIG: RETRANSMIT : 3 TIMEOUT : 5 RADIUS-AUTHENTICATION-SERVER(s): List is empty RADIUS-ACCOUNTING-SERVER(s): List is empty DHCP-RELAY(s): MAXHOPS : 4 THRESHOLD : 4 ADMIN-STATUS : DISABLE OPER-STATUS : DOWN DHCP-RELAY SERVER(s): List is empty DHCP Relay Statistics: Recv'd Discover Packets 0 Recv'd Offer Packets 0 Recv'd Ack Packets 0 Recv'd Nak Packets 0 Recv'd Request Packets 0 Recv'd Release Packets 0 Recv'd Decline Packets 0 Out Discover Packets 0 Out Offer Packets 0 Out Ack Packets 0 Out Nak Packets 0 Out Request Packets 0 Out Release Packets 0 Out Decline Packets 0

Syntax	Example
DISPLAY <argument>	IPNOS : CONFIG VPN(50) VR(172.16.16.1)# **display vi** VI(s): 1 2 3 4 5

Default Value

None

Command Level

VR

Related Commands

show

dns

Set the primary and secondary domain name servers (DNSs). These servers have an addressing system that associates a domain name with an IP address.

Syntax	*Example*
`DNS {PRIMARY | SECONDARY}`	`IPNOS : CONFIG VPN(50) VR(2.2.2.2)#` **`dns primary 3.3.3.3`** `VR(2.2.2.2): Object successfully set`

Default Value

None

Command Level

VR

Related Commands

None

ds-byte

Set the differentiated services (DS) field in an IP header for a packet leaving the VR. The **ds-byte** sets the precedence for forwarding packets by the nodes inside the network. Choose from 0 to 255. Use RFC 1349 and 2474 to calculate the appropriate byte value. Use the **no** option to disable the command. Table 16-5 describes the pre-set values for compatibility with non-CoSine routers.

Table 16-5 Possible Values for IP Precedence Compatibility

Value	*Description*
0	Routine
32	Priority
64	Immediate
96	Flash
128	Flash Override
160	Critical/ECP
192	Internet Control
224	Network Control

NOTE: The **ds-byte** command affects all outgoing packets from the VR. The ACL **tos** command, in contrast, affects only the incoming packets from IP addresses specified in the ACL.

Syntax	*Example*
DS-BYTE <value>	IPNOS : CONFIG VPN(50) VR(1.1.1.1)# **ds-byte 32** IPNOS : CONFIG VPN(50) VR(1.1.1.1)# **display ds-byte** DS-BYTE : 32
[NO] DS-BYTE	IPNOS : CONFIG VPN(50) VR(1.1.1.1)# **no ds-byte** IPNOS : CONFIG VPN(50) VR(1.1.1.1)# **display ds-byte** DS-BYTE : DISABLE

Default Value

Disabled

Command Level

VR

Related Commands

None

export-policy

Go to to the export-policy command level. Use for setting rules for outgoing packets. Use the **no** option to delete the export policy. The following local commands are available in the export-policy level.

- "acl" on page 16-46
- "action" on page 16-47
- "bgp-as" on page 16-48
- "bgp-community" on page 16-49
- "display" on page 16-51
- "metric" on page 16-52
- "metric" on page 16-52
- "peer-as" on page 16-53
- "protocol" on page 16-54
- "rip-ip" on page 16-56

Syntax	*Example*

`[NO] EXPORT-POLICY <policyID>`	`IPNOS : CONFIG VPN(50) VR(2.2.2.2)# export-policy 4`
	`VR(2.2.2.2): Object successfully set`
	
	`IPNOS : CONFIG VPN(50) VR(2.2.2.2) EXPORT-POLICY(4)#`

Default Value

None

Command Level

VR

Related Commands

import-policy

acl

Assign an access control list (ACL) to the specified export policy. See "acl" on page 16-7 for more information.

Syntax	Example
ACL <aclId>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) EXPORT-POLICY(4)# **acl 202** IPNOS : CONFIG VPN(50) VR(2.2.2.2) EXPORT-POLICY(4)# **commit** Commit configuration EXPORT-POLICY(4): Object successfully set

Default Value

None

Command Level

export-policy

Related Commands

acl
import-policy acl

action

Use the **action** command to set the export policy to permit export to or block export from the virtual router (VR). If used for BGP, set Action to export.

Syntax	Example
ACTION {EXPORT | BLOCK}	IPNOS :CONFIG VPN(50) VR(2.2.2.2) EXPORT-POLICY(2)# **action block** IPNOS : CONFIG VPN(50) VR(1.1.1.1) EXPORT-POLICY(1)# **action export**

Default Value

block

Command Level

export-policy

Related Commands

import-policy action

bgp-as

Set the Border Gateway Protocol (BGP) autonomous system (AS) number.

Syntax	*Example*
BGP-AS <as-number>	IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)# **bgp-as 333**

Default Value

Wildcard (*)

Command Level

export-policy

Related Commands

bgp (routing protocol)
bgp-community (export policy)
bgp-community (import policy)

bgp-community

Set up BGP route filtering or grouping between subdivisions of large autonomous systems called confederations. A community is a group of destinations which share some common property. Up to two BGP community filters can be set up. Table 16-6 describes the available parameters. Refer to RFCs 1965, 1997, and 1998 for details. Set the BGP-community string to 0:0 to remove it.

Table 16-6 BGP-Community Parameters

Parameter	*Description*
a:b	The local BGP AS number and the community number. The values are displayed in decimal form, representing the two 16-bit halves of a 32-bit string. The full 32-bit string is then shown in parentheses: 16bit:16bit (32bit)
No-export	Routes go no further than the receiving AS. All routes received carrying a communities attribute containing this value MUST NOT be advertised outside a BGP confederation boundary (a stand-alone autonomous system that is not part of a confederation should be considered a confederation itself). The hex value 0xFFFFFF01 is represented by the following 16bit:16bit (32bit) decimal combination: `65535:65281(-255)`
No-advertise	Routes go no further than the receiving router. All routes received carrying a communities attribute containing this value MUST NOT be advertised to other BGP peers. The hex value 0xFFFFFF02 is represented by the following 16bit:16bit (32bit) decimal combination: `65535:65282(-254)`
No-export-sub	Routes go no further than the sub-AS of a confederation. All routes received carrying a communities attribute containing this value MUST NOT be advertised to external BGP peers (this includes peers in other members autonomous systems inside a BGP confederation). The hex value 0xFFFFFF03 is represented by the following 16bit:16bit (32bit) decimal combination: `65535:65283(-253)`

Syntax	*Example*

BGP-COMMUNITY {<as_number>:<community_number>}	`IPNOS : CONFIG VPN(50) VR(1.1.1.1) EXPORT-POLICY(1)#` **`bgp-community 30:80`**
	
	`IPNOS : CONFIG VPN(50) VR(1.1.1.1) EXPORT-POLICY(1)#` **`display bgp`**
	`BGP-AS : *`
	`BGP-COMMUNITY : 30:80(1966160) 0:0(0)`

Syntax	*Example*
BGP COMMUNITY NO-EXPORT	IPNOS : CONFIG VPN(50) VR(1.1.1.1) EXPORT-POLICY(1)# **bgp-community no-export** IPNOS : CONFIG VPN(50) VR(1.1.1.1) EXPORT-POLICY(1)# **display bgp** BGP-AS : * BGP-COMMUNITY : 65535:65281(-255) 0:0(0)
BGP-COMMUNITY NO-ADVERTISE	IPNOS : CONFIG VPN(50) VR(1.1.1.1) EXPORT-POLICY(1)# **bgp-community no-advertise** IPNOS : CONFIG VPN(50) VR(1.1.1.1) EXPORT-POLICY(1)# **display bgp** BGP-AS : * BGP-COMMUNITY : 65535:65282(-254) 0:0(0)
BGP-COMMUNITY NO-EXPORT-SUB	IPNOS : CONFIG VPN(50) VR(1.1.1.1) EXPORT-POLICY(1)# **bgp-community no-export-sub** IPNOS : CONFIG VPN(50) VR(1.1.1.1) EXPORT-POLICY(1)# **display bgp** BGP-AS : * BGP-COMMUNITY : 65535:65283(-253) 0:0(0)
BGP-COMMUNITY <parameter1> <parameter2>	IPNOS : CONFIG VPN(50) VR(1.1.1.1) EXPORT-POLICY(1)# **bgp-community no-export no-advertise** IPNOS : CONFIG VPN(50) VR(1.1.1.1) EXPORT-POLICY(1)# **display bgp** BGP-AS : * BGP-COMMUNITY : 65535:65281(-255) 65535:65282(-254)
BGP-COMMUNITY <parameter1> <parameter2>	IPNOS : CONFIG VPN(50) VR(1.1.1.1) EXPORT-POLICY(1)# **bgp-community 30:80 40:100** IPNOS : CONFIG VPN(50) VR(1.1.1.1) EXPORT-POLICY(1)# **display bgp** BGP-AS : * BGP-COMMUNITY : 30:80(1966160) 40:100(2621540)

Default Value

None

Command Level

export-policy

Related Commands

bgp (routing protocol)
bgp-as (export policy)
bgp-community (import policy)

display

Use the **display** command to show the current configuration of the export policy on the VR. You can also limit the display to only the items you want to see.

Syntax	*Example*
DISPLAY	IPNOS: CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(100)# **display** VPN : 50 VR : 10.10.10.1 EXPORT-POLICY : 100 ACTION : BLOCK ACL : 0 PEER-AS : * PROTOCOL : ALL BGP-AS : * EXT-RTE-TAG : * RIP-IP : 0.0.0.0 NEIGHBOR : 0.0.0.0 METRIC : -1 BGP-COMMUNITY : 0:0(0) 0:0(0) ROUTE-TYPE : ANY MET-TYPE : ANY LEVEL : ANY
DISPLAY <argument>	IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)# **display action** ACTION : BLOCK

Default Value

None

Command Level

export-policy

Related Commands

import-policy display
show

metric

Assign a metric value to all packets subject to the export policy. The metric value determines which route or link is the best path to a destination. It is sometimes known as a routing cost. The fewer hops a packet has to take to get to its destination, the lower the metric. Possible values are from -1 (wildcard) and 2 to 2,147,483,648.

Syntax	*Example*

Set metric to 5 hops:
`METRIC <value>`
	`IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)#`
	`metric 5`
	
	`IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)#`
	`display metric`
	`METRIC : 5`

Set metric to the current metric:
`METRIC <value>`
	`IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)#`
	`metric -1`
	
	`IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)#`
	`display metric`
	`METRIC : -1`

Default Value

-1

Command Level

export-policy

Related Commands

None

peer-as

Set the autonomous system (AS) number of the remote peer.

Syntax	*Example*
`PEER-AS <as-number>`	`IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)#` **`peer-as 202`**

Default Value

Wildcard (*)

Command Level

export-policy

Related Commands

import-policy peer-as

protocol

Set the routes to be exported by protocol type. Table 16-7 lists available protocols:

Table 16-7 Export Policy Protocol Parameters

Protocol	*Description*
LOCAL	Export local routes to other routers.
BGP	Export external BGP routes to other routers.
RIP	Export RIP routes to other routers.
OSPF	Export OSPF routes to other routers.
STATIC	Export static routes to other routers.
All	Export all routes to other routers.
ISIS	Export IS-IS routes to other routers
Aggregate	Export an aggregate of available routes to other routers.

Syntax	*Example*
PROTOCOL {LOCAL | BGP | RIP | OSPF | STATIC | ALL | ISIS | AGGREGATE}	IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)# **protocol local** IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)# **display protocol** PROTOCOL : LOCAL
PROTOCOL {LOCAL | BGP | RIP | OSPF | STATIC | ALL | ISIS | AGGREGATE}	IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)# **protocol bgp** IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)# **display protocol** PROTOCOL : BGP
PROTOCOL {LOCAL | BGP | RIP | OSPF | STATIC | ALL | ISIS | AGGREGATE}	IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)# **protocol i-bgp** IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)# **display protocol** PROTOCOL : I-BGP

Syntax	*Example*
PROTOCOL {LOCAL | BGP | RIP | OSPF | STATIC | ALL | ISIS | AGGREGATE}	IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)# **protocol rip** IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)# **display protocol** PROTOCOL : RIP
PROTOCOL {LOCAL | BGP | RIP | OSPF | STATIC | ALL | ISIS | AGGREGATE}	IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)# **protocol OSPF** IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)# **display protocol** PROTOCOL : OSPF
PROTOCOL {LOCAL | BGP | RIP | OSPF | STATIC | ALL | ISIS | AGGREGATE}	IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)# **protocol OSPF-EXT** IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)# **display protocol** PROTOCOL : OSPF-EXT
PROTOCOL {LOCAL | BGP | RIP | OSPF | STATIC | ALL | ISIS | AGGREGATE}	IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)# **protocol static** IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)# **display protocol** PROTOCOL : STATIC
PROTOCOL {LOCAL | BGP | RIP | OSPF | STATIC | ALL | ISIS | AGGREGATE}	IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)# **protocol all** IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)# **display protocol** PROTOCOL : ALL
PROTOCOL {LOCAL | BGP | RIP | OSPF | STATIC | ALL | ISIS | AGGREGATE}	IPNOS: CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(100)# **protocol isis** IPNOS: CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(100)# **dis proto** PROTOCOL : ISIS
PROTOCOL {LOCAL | BGP | RIP | OSPF | STATIC | ALL | ISIS | AGGREGATE}	IPNOS: CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(100)# **protocol aggregate** IPNOS: CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(100)# **display protocol** PROTOCOL : AGGREGATE

Default Value

All routes are exported.

Command Level

export-policy

Related Commands

None

rip-ip

Assign an IP address for the Router Information Protocol (RIP-IP).

Syntax	*Example*
RIP-IP <address>	IPNOS : CONFIG VPN(50) VR(10.10.10.1) EXPORT-POLICY(1)# **rip-ip 1.1.1.1**

Default Value

0.0.0.0

Command Level

export-policy

Related Commands

None

froip

Go to the frame relay interface over an Internet Protocol (IP) command level. The IPSX switch with Frame Relay over IP can be considered a virtual Frame Relay device from the customer's perspective. See Chapter 19, "Frame Relay over IP Commands" for a list of commands.

NOTE: Frame Relay over IP works only with DS3 and E3 blades. Also, you must assign VI1 to either a public or private IP address before using **froip**.

Syntax	*Example*

`FROIP`	`IPNOS : CONFIG VPN(50) VR(172.16.16.1)#froip`
	`FROIP: Object successfully set`
	
	`IPNOS : CONFIG VPN(50) VR(172.16.16.1) FROIP#`

Default Value

None

Command Level

VR

Related Commands

None

fw

Go to the firewall proxy command level. See Chapter 18, “Firewall Commands” for a list of commands. Each proxy has its own command level.

NOTE: Firewall proxies are processor intensive features. You can spread the processing load across blades by creating a VR that includes at least one processor blade or a trunk or access blade with an extra processor engine card. Refer to the “vr” on page 15-11 for more information. Firewalls cannot be set up on control blades.

Syntax	*Example*
FW ACCESS <priority-number>	IPNOS : CONFIG VPN(50) VR(172.16.16.1)# **fw access 1** VR(172.16.16.1): Object successfully set IPNOS : CONFIG VPN(50) VR(172.16.16.1) ACCESS(1)#
FW AOL-PROXY	IPNOS : CONFIG VPN(50) VR(172.16.16.1)# **fw aol-proxy** VR(172.16.16.1): Object successfully set IPNOS : CONFIG VPN(50) VR(172.16.16.1) AOL-PROXY#
FW CS-PROXY	IPNOS : CONFIG VPN(50) VR(172.16.16.1)# **fw cs-proxy** VR(172.16.16.1): Object successfully set IPNOS : CONFIG VPN(50) VR(172.16.16.1) CS-PROXY#
FW FTP-PROXY	IPNOS : CONFIG VPN(50) VR(172.16.16.1)# **fw ftp-proxy** VR(172.16.16.1): Object successfully set IPNOS : CONFIG VPN(50) VR(172.16.16.1) FTP-PROXY#
FW HTTP-PROXY	PSX : CONFIG VPN(50) VR(172.16.16.1)# **fw http-proxy** VR(172.16.16.1): Object successfully set IPNOS : CONFIG VPN(50) VR(172.16.16.1) HTTP-PROXY#
FW LDAP-PROXY	IPNOS : CONFIG VPN(50) VR(10.1.1.1)# **fw ldap-proxy** VR(10.1.1.1): Object successfully set IPNOS : CONFIG VPN(50) VR(10.1.1.1) LDAP-PROXY#
FW LNOTES-PROXY	IPNOS : CONFIG VPN(50) VR(10.1.1.1)# **fw lnotes-proxy** VR(10.1.1.1): Object successfully set IPNOS : CONFIG VPN(50) VR(10.1.1.1) LNOTES-PROXY#

Syntax	Example
FW NETM-PROXY	IPNOS : CONFIG VPN(50) VR(172.16.16.1)# **fw netm-proxy** VR(172.16.16.1): Object successfully set IPNOS : CONFIG VPN(50) VR(172.16.16.1) NETM-PROXY#
FW NNTP-PROXY	IPNOS : CONFIG VPN(50) VR(10.1.1.1)# **fw nntp-proxy** VR(10.1.1.1): Object successfully set IPNOS : CONFIG VPN(50) VR(10.1.1.1) NNTP-PROXY#
FW PLUG-PROXY	IPNOS : CONFIG VPN(50) VR(10.1.1.1)# **fw plug-proxy** VR(10.1.1.1): Object successfully set IPNOS : CONFIG VPN(50) VR(10.1.1.1) PLUG-PROXY#
FW POP3-PROXY	IPNOS : CONFIG VPN(50) VR(10.1.1.1)# **fw pop3-proxy** VR(10.1.1.1): Object successfully set IPNOS : CONFIG VPN(50) VR(10.1.1.1) pop3-PROXY
FW RSH-PROXY	IPNOS : CONFIG VPN(50) VR(172.16.16.1)# **fw rsh-proxy** VR(172.16.16.1): Object successfully set IPNOS : CONFIG VPN(50) VR(172.16.16.1) RSH-PROXY#
FW SMTP-PROXY	IPNOS : CONFIG VPN(50) VR(10.1.1.1)# **fw smtp-proxy** VR(10.1.1.1): Object successfully set IPNOS : CONFIG VPN(50) VR(10.1.1.1) SMTP-PROXY
FW SQL-PROXY	IPNOS : CONFIG VPN(50) VR(172.16.16.1)# **fw sql-proxy** VR(172.16.16.1): Object successfully set IPNOS : CONFIG VPN(50) VR(172.16.16.1) SQL-PROXY#
FW TELNET-PROXY	IPNOS : CONFIG VPN(50) VR(10.1.1.1)# **fw telnet-proxy** VR(10.1.1.1): Object successfully set IPNOS : CONFIG VPN(50) VR(10.1.1.1) TELNET-PROXY

Default Value

None

Command Level

VR

Related Commands

None

hostname

Specify the VR hostname for setting up PPTP and L2TP. Refer to“l2tp” on page 16-78 and “pptp” on page 16-113 for more information.

NOTE: This command has no relationship to the command “name” on page 16-106.

Syntax	*Example*
HOSTNAME <string>	IPNOS : CONFIG VPN(50) VR(2.2.2.2)# **hostname CoSinecom.com** VR(10.10.10.1): Object successfully set HOSTNAME CoSinecom.com: Object successfully set IPNOS : CONFIG VPN(50) VR(2.2.2.2)#

Default Value

None

Command Level

VR

Related Commands

l2tp
pptp

import-policy

Go to to the import-policy command level. Use for setting rules for incoming packets. Use the **no** option to delete the import policy. The following local commands are available in the import-policy level.

- "acl" on page 16-62
- "action" on page 16-62
- "bgp-community" on page 16-63
- "display" on page 16-64
- "home-as" on page 16-65
- "origin" on page 16-65
- "peer-as" on page 16-66
- "preference" on page 16-67

Syntax	*Example*

`[NO] IMPORT-POLICY <policyId>`	`IPNOS : CONFIG VPN(50) VR(10.10.10.1)# import-policy 200`
	`VR(10.10.10.1): Object successfully set`
	
	`IPNOS : CONFIG VPN(50) VR(10.10.10.1) IMPORT-POLICY(200)#`

Default Value

None

Command Level

VR

Related Commands

export-policy

acl

Assign an access control list (ACL) to the specified import policy. See "acl" on page 16-7 for more information.

Syntax	*Example*
ACL <aclId>	IPNOS: CONFIG VPN(50) VR(10.10.10.1) IMPORT-POLICY(1)# **acl 32** IPNOS: CONFIG VPN(50) VR(10.10.10.1) IMPORT-POLICY(1)# **display acl** ACL : 32

Default Value

None

Command Level

import-policy

Related Commands

acl
export-policy acl

action

Permit or deny packets affected by the import policy.

Syntax	*Example*
ACTION {PERMIT | DENY}	IPNOS: CONFIG VPN(50) VR(10.10.10.1) IMPORT-POLICY(1)# **action permit** IPNOS: CONFIG VPN(50) VR(10.10.10.1) IMPORT-POLICY(1)# **display action** ACTION : PERMIT
ACTION {PERMIT | DENY}	IPNOS: CONFIG VPN(50) VR(10.10.10.1) IMPORT-POLICY(1)# **action deny** IPNOS: CONFIG VPN(50) VR(10.10.10.1) IMPORT-POLICY(1)# **display action** ACTION : DENY

Default Value

deny

Command Level

import-policy

Related Commands

export-policy action

bgp-community

Set up BGP route filtering or grouping between sub-divisions of large autonomous systems called confederations. A community is a group of destinations that share some common property. Table 16-8 describes the available parameters. Refer to RFCs 1965, 1997, and 1998 for details. Set the BGP-community string to 0:0 to remove it.

Table 16-8 BGP-Community Parameters (Import Policy)

Parameter	*Description*
a:b	The local BGP AS number and the community number. The values are displayed in decimal form, representing the two 16-bit halves of a 32-bit string. The full 32-bit string is then shown in parentheses: 16bit:16bit (32bit)

Syntax	*Example*
BGP-COMMUNITY {a:b [c:d]}	IPNOS : CONFIG VPN(50) VR(1.1.1.1) IMPORT-POLICY(1)# **bgp-community 30:80** IPNOS : CONFIG VPN(50) VR(1.1.1.1) IMPORT-POLICY(1)# display bgp BGP-COMMUNITY : 30:80(1966160)

Default Value

None

Command Level

import-policy

Related Commands

bgp (routing protocol)
bgp-as (export policy)
bgp-community (export policy)

display

Use the **display** command to show the current configuration for the import policy on the VR. You can also limit the display to only the items you want to see.

Syntax	*Example*
DISPLAY	IPNOS: CONFIG VPN(50) VR(10.10.10.1) IMPORT-POLICY(200)# display VPN : 50 VR : 10.10.10.1 IMPORT-POLICY : 200 ACTION : DENY ACL : 32 PEER-AS : * HOME-AS : * ORIGIN : ANY PREFERENCE : -1 BGP-COMMUNITY : 30:80(1966160)
DISPLAY <argument>	IPNOS : CONFIG VPN(50) VR(10.10.10.1) IMPORT-POLICY(1)# **display import-policy** IMPORT-POLICY : 200

Default Value

None

Command Level

import-policy

Related Commands

export-policy display
show

home-as

Assign the home autonomous system (AS) address.

Syntax	*Example*
`HOME-AS <as-number>`	`IPNOS : CONFIG VPN(50) VR(2.2.2.2) IMPORT-POLICY(4)# home-as 203`

Default Value

Wildcard (*)

Command Level

import-policy

Related Commands

export-policy bgp-as

origin

Filter a BGP route based on its origin. BGP peers send each other update messages to advertise feasible routes. The origin attribute in these messages define the origin of a route. Table 16-9 describes the attributes available.

Table 16-9 BGP Origin Attributes

Origin Attribute	*Description*
IGP	Origin of the route came from an internal gateway protocol within the originating AS. For example, a route may come from an OSPF area inside the peer AS.
Unknown	Origin of the route cannot be determined. For example, a packet may come via a static route instead of another protocol.
Any	Any origin. Ignore the origin attribute in the BGP update message. This is the default value.

Syntax	Example
ORIGIN {IGP | UNKNOWN | ANY}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) IMPORT-POLICY(4)# **origin igp** IPNOS : CONFIG VPN(50) VR(2.2.2.2) IMPORT-POLICY(4)# **display origin** ORIGIN : IGP

Default Value

Any

Command Level

import-policy

Related Commands

export-policy metric
router bgp

peer-as

Set the autonomous system (AS) number for a remote peer.

Syntax	Example
PEER-AS <as-number>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) IMPORT-POLICY(4)# **peer-as 203**

Default Value

Wildcard (*)

Command Level

import-policy

Related Commands

export-policy peer-as

preference

Set the local preference for an incoming BGP route. Use when two or more BGP routes point to the same destination. The higher value is preferred. This command is similar to a multi-exit discriminator, but in the opposite direction. This command applies only to an import-policy whose action is set to PERMIT. Possible values range from -1 (wildcard) to 2,147,483,647.

Syntax	*Example*
PREFERENCE <preference-number>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) IMPORT-POLICY(4)# **preference 4**

Default Value

-1

Command Level

import-policy

Related Commands

import-policy action

ip

Configure the following features.

- IP Address pools dynamic NAT and PPTP. See “pptp” on page 16-113.
- Static NAT
- Dual static NAT (based on RFC 2663)
- Dynamic NAT
- Static route
- Activate or deactivate IP routing

Use the **no** option to delete a route or an address pool. Figure 16-1 and Figure 16-2 describe networks for basic static NAT, dynamic NAT, and dual static NAT used in the examples.

Figure 16-1 Static and Dynamic NAT

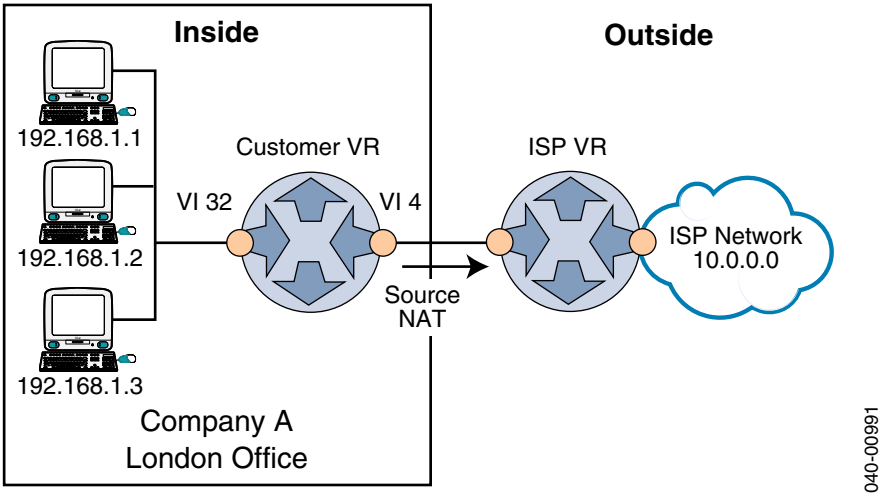


Figure 16-2 Dual Static NAT

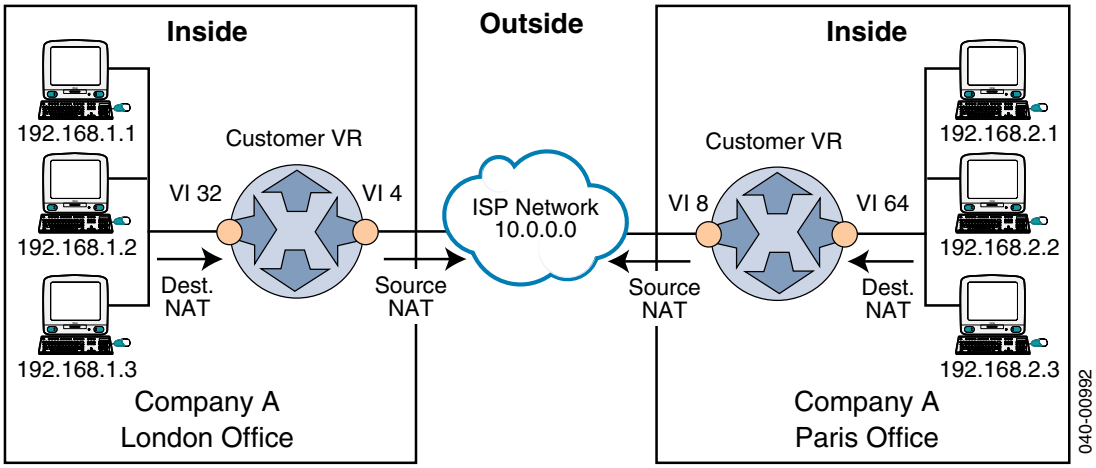


Table 16-10 describes the parameters available.

Table 16-10 IP Address Routing Commands

Routing Command	*Description*
IP local pool	Create a PPTP address pool. The command has the following parameters: • Name—The index number identifying the address pool. • Start-range—The first available IP address in the pool. • End-range—The final address in the pool (up to 256).
IP nat inside source static	Create a source static NAT to the VR or a VI. It translates a source IP address to a publicly known IP address. Add a subnet mask to retain the host part. The command has the following parameters: • Local IP address—The source IP address of the internal node. • Global IP address—The global IP address used for external communication. • Netmask—A network address mask. (Optional) • VI—Virtual Interface (Optional)
IP nat inside destination static	Create a destination static NAT to the VR or VI. . It translates a destination IP address to a publicly known IP address. Used with **ip nat inside source static**, the combination protects both the source and destination by using public IP addresses in their place. The command has the following parameters: • Global IP address—The destination IP address • Local IP address—The new public IP address. • Netmask—A network address mask. (Optional) • VI—Virtual Interface (Optional)

Table 16-10 IP Address Routing Commands (Continued)

Routing Command	*Description*
IP nat list	Attach a dynamic NAT pool to an access-list. See "access-list" on page 16-5. The command has the following parameters: • List ID—Access list • Pool ID—NAT address pool to which the access-list looks for the next available IP address for translation. • VI ID—VI to watch for inbound translatable packets. Must be the same VI in the access-list table.
IP nat pool	Create a dynamic NAT address pool. The IP address range is 0–255. The port ID range is 1–2048. The maximum number of IP addresses and ports for a single NAT pool is 100,000. The command has the following parameters: • Pool ID—index number identifying the address pool. • Start-address—The first available IP address in the pool. • End-address—The final address in the pool (the maximum size of an address pool is 255 IP addresses, or a class-C subnet). • Start IP port—The first available port for the NAT pool. • End IP port—The last available port for the NAT pool.
IP route	Set up a static route between a VI and a specific destination network. The command has the following parameters: • Prefix—The IP address of the destination network. • Mask—The IP subnet mask of the destination network. • Next-hop—The IP address of the next hop toward the destination. • VI <index>—The index number of the VI. • Metric—The metric value for the best path to a destination. Choose from -1 (wildcard) up to 2,147,483,648. • Tag—A marker to distinguish two static routes pointing to the same destination. Possible values are 1 to 256. *Note:* To set up a default route, set the IP address and the subnet mask to all zeroes.
IP routing	The **ip routing** command enables or disables IP routing on the VR.

Syntax	*Example*

Create local address pool for PPTP:
`[NO] IP LOCAL POOL <name> <start-range> <end-range>`
	`IPNOS: CONFIG VPN(50) VR(10.10.10.1)# ip local pool cosine 192.168.24.1 192.168.24.32`
	`VR(10.10.10.1): Object successfully set`
	`POOL cosine: Object successfully set`
	
	`IPNOS: CONFIG VPN(50) VR(10.10.10.1)# dis local`
	
	`LOCAL POOL(s):`
	`NAME | START-RANGE | END-RANGE`
	`cosine | 192.168.24.1 | 192.168.24.32`

Create source static NAT with a subnet to a VR:
`IP NAT INSIDE SOURCE STATIC <local-ip> <global-ip> [NETMASK <mask>] [VI <viid>]`
	`IPNOS: CONFIG VPN(50) VR(10.10.10.1)# ip nat inside source static 192.168.1.1 10.1.1.1`
	`VR(10.10.10.1): Object successfully set`
	`STATIC TRANSLATION (192.168.1.1 10.1.1.1 * *): Object successfully set`
	
	`IPNOS: CONFIG VPN(50) VR(10.10.10.1)# dis nat`
	
	`POOL(s):`
	`List is empty`
	
	`IP STATIC NAT(s):`
	`INSIDE LOCAL IP | INSIDE GLOBAL IP| NETMASK | VI | TYPE`
	`192.168.1.1 | 10.1.1.1 | * | * | SOURCE`
	
	`IP NAT TRANSLATION(s):`
	`List is empty`

Create source static NAT to a VI:
`IP NAT INSIDE SOURCE STATIC <local-ip> <global-ip> [NETMASK <mask>] [VI <viid>]`
	`IPNOS: CONFIG VPN(50) VR(10.10.10.1)# ip nat inside source static 192.168.1.2 10.1.1.2 vi 4`
	`VR(10.10.10.1): Object successfully set`
	`STATIC TRANSLATION (192.168.1.2 10.1.1.2 * 4): Object successfully set`
	
	`IPNOS: CONFIG VPN(50) VR(10.10.10.1)# dis nat`
	
	`POOL(s):`
	`List is empty`
	
	`IP STATIC NAT(s):`
	`INSIDE LOCAL IP | INSIDE GLOBAL IP| NETMASK | VI | TYPE`
	`192.168.1.1 | 10.1.1.1 | * | * | SOURCE`
	`192.168.1.2 | 10.1.1.2 | * | 4 | SOURCE`
	
	`IP NAT TRANSLATION(s):`
	`List is empty`

Syntax	*Example*

Create source static NAT with a subnet to a VI:
`IP NAT INSIDE SOURCE STATIC <local-ip> <global-ip> [NETMASK <mask>] [VI <viid>]`
	`IPNOS: CONFIG VPN(50) VR(10.10.10.1)# ip nat inside source static 192.168.1.0 10.1.1.0 netmask 255.255.255.0 vi 4`
	`VR(10.10.10.1): Object successfully set`
	`STATIC TRANSLATION (192.168.1.0 10.1.1.0 255.255.255.0 4): Object successfully set`
	
	`IPNOS: CONFIG VPN(50) VR(10.10.10.1)# dis nat`
	
	`POOL(s):`
	`List is empty`
	
	`IP STATIC NAT(s):`
	`INSIDE LOCAL IP | INSIDE GLOBAL IP| NETMASK | VI | TYPE`
	`192.168.1.0 | 10.1.1.0 | 255.255.255.0 | 4 | SOURCE`
	
	`IP NAT TRANSLATION(s):`
	`List is empty`

Delete source static NAT:
`NO IP NAT INSIDE SOURCE STATIC <local-ip> <global-ip> [VI <viId>]`
	`IPNOS: CONFIG VPN(50) VR(10.10.10.1)# no ip nat inside source static 192.168.1.1 10.1.1.1`
	`VR(10.10.10.1): Object successfully set`
	`STATIC TRANSLATION (192.168.1.1 10.1.1.1): Object successfully deleted`
	
	`IPNOS: CONFIG VPN(50) VR(10.10.10.1)# dis nat`
	
	`POOL(s):`
	`List is empty`
	
	`IP STATIC NAT(s):`
	`INSIDE LOCAL IP | INSIDE GLOBAL IP| NETMASK | VI | TYPE`
	`192.168.1.2 | 10.1.1.2 | * | 4 | SOURCE`
	
	`IP NAT TRANSLATION(s):`
	`List is empty`

Syntax	Example
Create destination static NAT with a subnet to a VI: IP NAT INSIDE DESTINATION STATIC <global-ip> <local-ip> [NETMASK <mask>] [VI <viid>]	IPNOS: CONFIG VPN(50) VR(10.10.10.1)# **ip nat inside destination static 192.168.2.0 10.1.2.0 netmask 255.255.255.0 vi 32** VR(10.10.10.1): Object successfully set STATIC TRANSLATION (192.168.2.0 10.1.2.0 255.255.255.0 32): Object successfully set IPNOS: CONFIG VPN(50) VR(10.10.10.1)# **dis nat** POOL(s): List is empty IP STATIC NAT(s): INSIDE LOCAL IP | INSIDE GLOBAL IP| NETMASK | VI | TYPE 192.168.1.0 | 10.1.1.0 | 255.255.255.0 | 4 | SOURCE 10.1.2.0 | 192.168.2.0 | 255.255.255.0 | 32 | DESTINATION IP NAT TRANSLATION(s): List is empty
Delete destination static NAT: NO IP NAT INSIDE DESTINATION STATIC <global-ip> <local-ip> [VI <viId>]	IPNOS: CONFIG VPN(50) VR(10.10.10.1)# **no ip nat inside destination static 192.168.2.0 10.1.2.0 vi 32** VR(10.10.10.1): Object successfully set STATIC TRANSLATION (192.168.2.0 10.1.2.0 32): Object successfully deleted IPNOS: CONFIG VPN(50) VR(10.10.10.1)# **dis nat** POOL(s): List is empty IP STATIC NAT(s): INSIDE LOCAL IP | INSIDE GLOBAL IP| NETMASK | VI | TYPE 192.168.1.0 | 10.1.1.0 | 255.255.255.0 | 4 | SOURCE IP NAT TRANSLATION(s): List is empty

Syntax	*Example*

Create destination static NAT to a VR:
`[NO] IP NAT INSIDE DESTINATION STATIC <global-ip> <local-ip> [VI <viid>]`
	`IPNOS: CONFIG VPN(50) VR(10.10.10.1)# ip nat inside destination static 192.168.2.1 10.1.2.1`
	`VR(10.10.10.1): Object successfully set`
	`STATIC TRANSLATION (192.168.2.1 10.1.2.1 * *): Object successfully set`
	
	`IPNOS: CONFIG VPN(50) VR(10.10.10.1)# dis nat`
	
	`POOL(s):`
	`List is empty`
	
	`IP STATIC NAT(s):`
	`INSIDE LOCAL IP | INSIDE GLOBAL IP| NETMASK | VI | TYPE`
	`10.1.2.1 | 192.168.2.1 | * | * | DESTINATION`
	
	`IP NAT TRANSLATION(s):`
	`List is empty`

Create destination static NAT to a VI:
`[NO] IP NAT INSIDE DESTINATION STATIC <global-ip> <local-ip> [VI <viid>]`
	`IPNOS: CONFIG VPN(50) VR(10.10.10.1)# ip nat inside destination static 192.168.2.2 10.1.2.2 vi 32`
	`VR(10.10.10.1): Object successfully set`
	`STATIC TRANSLATION (192.168.2.2 10.1.2.2 * 32): Object successfully set`
	
	`IPNOS: CONFIG VPN(50) VR(10.10.10.1)# dis nat`
	
	`POOL(s):`
	`List is empty`
	
	`IP STATIC NAT(s):`
	`INSIDE LOCAL IP | INSIDE GLOBAL IP| NETMASK | VI | TYPE`
	`10.1.2.1 | 192.168.2.1 | * | * | DESTINATION`
	`10.1.2.2 | 192.168.2.2 | * | 32 | DESTINATION`
	
	`IP NAT TRANSLATION(s):`
	`List is empty`

Syntax	*Example*
For NAT access lists: [NO] IP NAT LIST <listId> POOL <poolId> <viId>	IPNOS : CONFIG VPN(50) VR(10.1.1.1)# **access-list 1 permit 10.1.1.2 10.1.100.100 1** VR(10.1.1.1): Object successfully set ACCESS-LIST 1: Object successfully set IPNOS : CONFIG VPN(50) VR(10.1.1.1)# **ip nat list 1 pool 1 1** VR(10.1.1.1): Object successfully set LIST 1: Object successfully set IPNOS : CONFIG VPN(50) VR(10.1.1.1)# **Note:** You need to create a NAT access list using the **access-list** command before creating a NAT list.
Create dynamic NAT pool: [NO] IP NAT POOL <poolId> <start-address> <end-address> <start-port> <end-port>	IPNOS: CONFIG VPN(50) VR(10.10.10.1)# **access-list 1 permit 192.168.1.1 192.168.1.3 32** VR(10.10.10.1): Object successfully set ACCESS-LIST 1: Object successfully set IPNOS: CONFIG VPN(50) VR(10.10.10.1)# **dis access** ACCESS-LIST(s): ACCESS-LIST | VI | START-ADDRESS | END-ADDRESS | POOL 1 | 32 | 192.168.1.1 | 192.168.1.3 | 0 IPNOS: CONFIG VPN(50) VR(10.10.10.1)# **ip nat pool 1 10.1.1.1 10.1.1.10 1000 1050** VR(10.10.10.1): Object successfully set POOL 1: Object successfully set IPNOS: CONFIG VPN(50) VR(10.10.10.1)# **dis nat** POOL(s): POOL | START-ADDRESS | END-ADDRESS | START-PORT | END-PORT 1 | 10.1.1.1 | 10.1.1.10 | 1000 | 1050 IP STATIC NAT(s): List is empty IP NAT TRANSLATION(s): List is empty

Syntax	*Example*
Delete dynamic NAT pool: [NO] IP NAT POOL <poolId> <start-address> <end-address> <start-port> <end-port>	IPNOS: CONFIG VPN(50) VR(10.10.10.1)# **no ip nat pool 1 10.1.1.1 10.1.1.10 1000 1050** VR(10.10.10.1): Object successfully set POOL 1: Object successfully deleted IPNOS: CONFIG VPN(50) VR(10.10.10.1)# **dis nat** POOL(s): List is empty IP STATIC NAT(s): List is empty IP NAT TRANSLATION(s): List is empty
Attach NAT pool to access list: [NO] IP NAT LIST <listId> POOL <poolId> <viId>	IPNOS: CONFIG VPN(50) VR(10.10.10.1)# **dis access** ACCESS-LIST(s): ACCESS-LIST | VI | START-ADDRESS | END-ADDRESS | POOL 1 | 32 | 192.168.1.1 | 192.168.1.3 | 0 IPNOS: CONFIG VPN(50) VR(10.10.10.1)# **dis nat** POOL(s): POOL | START-ADDRESS | END-ADDRESS | START-PORT | END-PORT 1 | 10.1.1.1 | 10.1.1.10 | 1000 | 1050 IP STATIC NAT(s): List is empty IP NAT TRANSLATION(s): List is empty IPNOS: CONFIG VPN(50) VR(10.10.10.1)# i**p nat list 1 pool 1 32** VR(10.10.10.1): Object successfully set LIST 1: Object successfully set IPNOS: CONFIG VPN(50) VR(10.10.10.1)# **dis access** ACCESS-LIST(s): ACCESS-LIST | VI | START-ADDRESS | END-ADDRESS | POOL 1 | 32 | 192.168.1.1 | 192.168.1.3 | 1

Syntax	*Example*
For static routes: [NO] IP ROUTE <prefix> <mask> <next-hop> [VI <interface>] [<metric> [<tag>]]	IPNOS : CONFIG VPN(50) VR(10.1.1.1)# **ip route 1.1.1.1 255.255.255.0 1 5** VR(10.1.1.1): Object successfully set 1.1.1.1|255.255.255.0|0.0.0.0: Object successfully set IPNOS : CONFIG VPN(50) VR(10.1.1.1)# IPNOS : CONFIG VPN(50) VR(10.1.1.1)# **ip route 1.1.1.0 255.255.255.0 2.2.2.2** VR(10.1.1.1): Object successfully set 1.1.1.0|255.255.255.0|2.2.2.2: Object successfully set IPNOS : CONFIG VPN(50) VR(10.1.1.1)#
For two static routes pointing to the same destination: [NO] IP ROUTE <prefix> <mask> <next-hop> [VI <interface>] [<metric> [<tag>]]	IPNOS : CONFIG VPN(50) VR(10.1.1.1)# **ip route 1.1.1.1 255.255.255.0 1 5 10** VR(10.1.1.1): Object successfully set 1.1.1.1|255.255.255.0|0.0.0.0: Object successfully set IPNOS : CONFIG VPN(50) VR(10.1.1.1)# **ip route 1.1.1.1 255.255.255.0 2 20 12** VR(10.1.1.1): Object successfully set 1.1.1.1|255.255.255.0|0.0.0.0: Object successfully set
For enabling/disabling IP routing on the VR: [NO] IP ROUTING	IPNOS : CONFIG VPN(50) VR(10.1.1.1)# **ip routing** IPNOS : CONFIG VPN(50) VR(10.1.1.1)# **display routing** ROUTING : IP ROUTING

Default Value

IP routing is enabled.

Command Level

VR

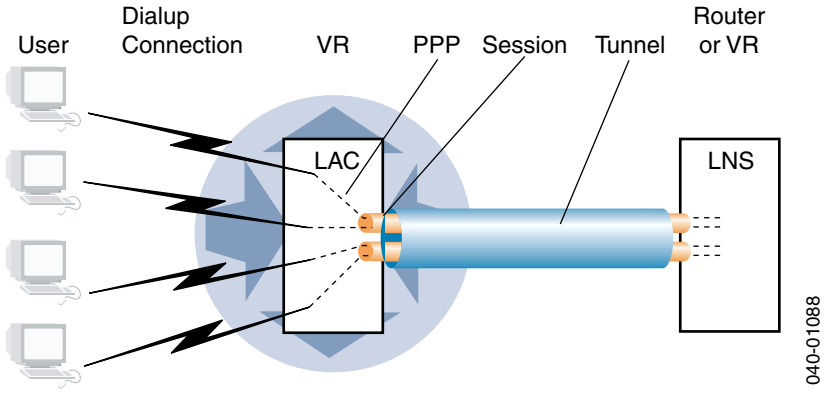
Related Commands

access-list
pptp

l2tp

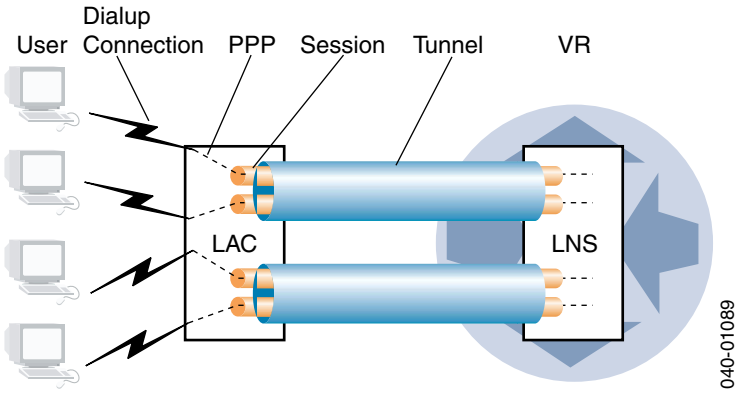
Configure the Level 2 Tunneling Protocol (L2TP). Similar to **pptp**, this feature creates a tunnel for a dialup customer. There are two components, the L2TP access concentrator (LAC), and the L2TP network server (LNS). The LAC receives incoming PPP packets from a user. The LAC serves as a passive relay to the LNS. It waits for incoming calls before contacting the LNS. Figure 16-3 describes L2TP LAC.

Figure 16-3 L2TP LAC



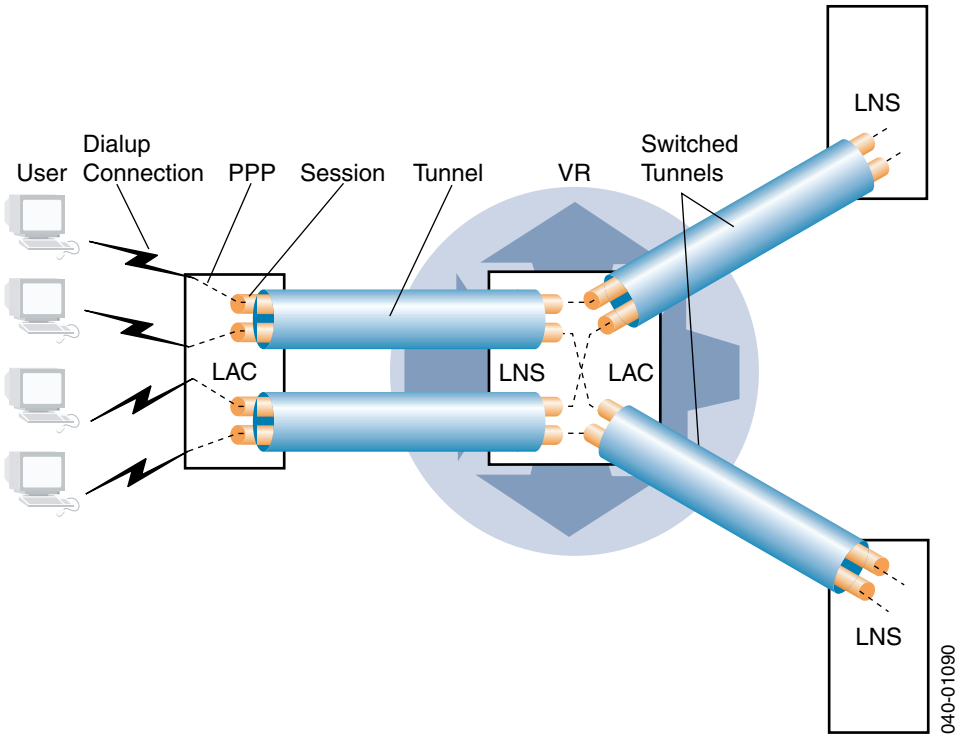
The LNS waits for an incoming LAC tunnel request. A tunnel is created between the LAC and LNS to transport multiple sessions. You can have multiple tunnels between the same LAC and LNS. Each tunnel has a peer IP address. The tunnel address is selected from an address pool. Create an IP address pool using the **ip local pool** command. Refer to "ip" on page 16-68 for details. A session is then created inside the tunnel for the PPP packets. Figure 16-4 describes L2TP LNS.

Figure 16-4 L2TP LNS



A VR can act as both as LAC and LNS. As an LNS, it receives sessions from other LACs. Then, as a LAC it forwards sessions through tunnels to other LNSs in the network. Figure 16-5 describes tunnel switching.

Figure 16-5 L2TP Tunnel Switching



Use an optional RADIUS servers on the LAC and the LNS for extra authentication support.

Using the **l2tp** command also puts you into the L2TP command level. The following commands are available at this level:

- "display" on page 16-80
- "l2tp" on page 16-82
- "lac" on page 16-90
- "lns" on page 16-91
- "ppp" on page 16-93
- "tunnel" on page 16-96
- "tunnel-switch" on page 16-99

NOTE: The L2TP feature works only on VRs that include a processor blade. See "create" on page 15-2 for information on creating VRs across multiple blades.

Syntax	*Example*

`L2TP`	`IPNOS : CONFIG VPN(50) VR(2.2.2.2)# l2tp`
	`VR(2.2.2.2): Object successfully set`
	
	`IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP#`

Default Value

None

Command Level

VR

Related Commands

pptp

display

View the current L2TP configuration.

Syntax	*Example*
DISPLAY	

```
IPNOS: CONFIG VPN(100) VR(2.2.2.2) L2TP# display
VPN                           : 100
VR                            : 1.1.1.1

SERVICE-VI                    : 1
MAX-TUNNELS                   : 256
MAX-SESSIONS-PER-TUNNEL       : 256

TUNNEL AUTHENTICATION         : ENABLE
TUNNEL HELLO                  : 60
TUNNEL MAX-RETRANSMISSION     : 5
TUNNEL RETRANSMISSION-TIMEOUT : 16
TUNNEL RECEIVE-WINDOW-SIZE    : 10
TUNNEL EMPTY-TUNNEL-TIMEOUT   : 120

SESSION FLOW-CONTROL          : ENABLE
SESSION RECEIVE-WINDOW-SIZE   : 10
SESSION TRANSMIT-QUEUE-SIZE   : 5
SESSION PACKET-SEQUENCING     : DISABLE

PPP AUTHENTICATION            : PAP
PPP CONFIG-REQ-RETRANSMIT     : 10
PPP TERM-REQ-RETRANSMIT       : 2
PPP TIMEOUT                   : 3
PPP REAUTH                    : 0
PPP KEEPALIVE                 : 10
PPP IP-LOCAL-POOL-NAME        : IP address pool group 3

STATISTICS:
1 Total tunnels: 1 active, 0 failed, 0 authentication failed
2 Total sessions: 2 active, 1 failed
Control: 0 packets sent, 0 packets received
Payload: 0 packets sent, 0 packets received, 0 packets discarded

LAC ADMIN-STATUS              : DISABLE
LNS ADMIN-STATUS              : DISABLE

TUNNEL(s):
LOCAL-TUNNEL-ID  | PEER-IP-ADDRESS | SESSION-COUNT  | UP-
TIME (sec)  |
1                | 192.168.150.80  | 2              | 4202
        |

TUNNEL-SWITCH TABLE:
CLIENT-NAME             | MODE            | SWITCH-TO
LAC-1                   | PASS-THROUGH    | 13.13.11.90
```

Default Value

None

Command Level

l2tp

Related Commands

pptp display

l2tp

Configure L2TP tunnels and sessions. Changes that you make affect future tunnels and sessions and will not change currently running ones. Use the **no** option to reset a command to its default value. Table 16-11 describes the parameters available.

Table 16-11 L2TP Parameters

Parameter	*Description*
Max-sessions-per-tunnel	Maximum sessions allowed per tunnel. The range is from 1 to 65535 sessions. The default value is 256 sessions.
Max-tunnels	Maximum tunnels allowed. The range is from 1 to 65535 tunnels. The default value is 256 tunnels.
Service-VI	Designated VI for LAC and LNS transmissions. Choose from 1 to 64,000. The default value is VI 1 (The loopback VI). This parameter is required. *Note:* The VI must have an IP address assigned to it, and the LAC or LNS must be disabled before you run this command.

Table 16-11 L2TP Parameters (Continued)

Parameter	*Description*
Session	Session configuration. The default values should work for all new LACs and LNSs. *Note:* New changes to these parameters will not affect pre-existing sessions. You can configure the following parameters: • Flow-control—Enables or disables payload flow control to prevent overloading the LNS with packets. This feature was in the draft specification but was not part of the final RFC 2661. The final RFC does not have flow control, but still uses packet-sequencing. The default value is disabled. *Note:* LACs that follow the final RFC may reject sessions with flow-control enabled. • Packet-sequencing—Enables or disables sending payloads with sequence numbers. The default value is disabled. *Note:* Windows 2000 supports only packet sequencing based on the older RFC 2661 draft. To make Windows 2000 send sequence numbers on its packets, enable packet-sequencing, enable flow-control, and set receive-window-size to 0. • Receive-window-size—Session receive window size. Works with session flow control. Choose from 0 to 16 packets. The default value is 10 packets. When set to 0, the LNS requests the LACs to send packets with sequence numbers. • Transmit-queue-size—Sets the packet queue size. If flow control is enabled, and the receive window size is full, packets sent to the peer will be held in a queue until the receive window is open again. Choose from 1 to 16 packets. The default value is 5 packets.

Table 16-11 L2TP Parameters (Continued)

Parameter	*Description*
Session	Session configuration. The default values should work for all new LACs and LNSs. *Note:* New changes to these parameters will not affect pre-existing sessions. You can configure the following parameters: • Flow-control—Enables or disables payload flow control to prevent overloading the LNS with packets. This feature was in the draft specification but was not part of the final RFC 2661. The final RFC does not have flow control, but still uses packet-sequencing. The default value is disabled. *Note:* LACs that follow the final RFC may reject sessions with flow-control enabled. • Packet-sequencing—Enables or disables sending payloads with sequence numbers. The default value is disabled. *Note:* Windows 2000 supports only packet sequencing based on the older RFC 2661 draft. To make Windows 2000 send sequence numbers on its packets, enable packet-sequencing, enable flow-control, and set receive-window-size to 0. • Receive-window-size—Session receive window size. Works with session flow control. Choose from 0 to 16 packets. The default value is 10 packets. When set to 0, the LNS requests the LACs to send packets with sequence numbers. • Transmit-queue-size—Sets the packet queue size. If flow control is enabled, and the receive window size is full, packets sent to the peer will be held in a queue until the receive window is open again. Choose from 1 to 16 packets. The default value is 5 packets.

Table 16-11 L2TP Parameters (Continued)

Parameter	*Description*
Tunnel	Tunnel configuration. *Note:* New changes to these parameters will not affect pre-existing tunnels. You can configure the following parameters: • Authentication—Enables or disables tunnel authentication. If enabled, a challenge is sent to the peer during tunnel creation. The tunnel is torn down if the peer does not send a valid response. The default value is disabled. • Empty-tunnel-timeout—Timeouts in seconds before an empty tunnel is deleted. Choose from 120 to 3600 seconds. The default value is 120 seconds (2 minutes). To keep the empty tunnel running indefinitely, set the value to 0 seconds. • Hello—Interval in seconds for transmitting HELLO (keepalive) packets. These packets are used to keep the tunnel alive by testing the LAC-LNS connection. Choose from 30 to 3600 seconds. The default value is 60 seconds. To stop sending HELLO packets, set the value to 0. • Max-retransmission—Maximum number of retransmissions the tunnel will attempt before assuming the peer is no longer responding. Choose from 5 to 10. The default value is 5 retransmissions. • Receive-window-size—Specifies the maximum number of packets the tunnel peer can send without waiting for an acknowledgement from the peer. Choose from 1 to 16. The default value is 10 packets. • Retransmission-timeout—Timeout interval in seconds the tunnel will wait before retransmitting a control packet that has not been acknowledged. Choose from 10 to 30 seconds. The default value is 16 seconds.

Syntax	*Example*
`[NO] L2TP MAX-SESSIONS-PER-TUNNEL <value>`	`IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP#` **`l2tp max-sessions-per-tunnel 128`** `IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP#` **`display max-sessions`** `MAX-SESSIONS-PER-TUNNEL : 128`
`[NO] L2TP MAX-SESSIONS-PER-TUNNEL <value>`	`IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP#` **`no l2tp max-sessions-per-tunnel`** `IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP#` **`display max-sessions`** `MAX-SESSIONS-PER-TUNNEL : 256`

Syntax	*Example*
[NO] L2TP MAX-TUNNELS <value>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **l2tp max-tunnels 128** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display max-tunnel** MAX-TUNNELS : 128
[NO] L2TP MAX-TUNNELS <value>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **no l2tp max-tunnel** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display max-tunnel** MAX-TUNNELS : 256
[NO] L2TP SERVICE-VI <viId>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **l2tp service-vi 2** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display service** SERVICE-VI : 2
[NO] L2TP SERVICE-VI <viId>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **no l2tp service-vi** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display service-vi** SERVICE-VI : 1
L2TP SESSION FLOW-CONTROL {ENABLE | DISABLE}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **l2tp session flow-control enable** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display session flow-control** SESSION FLOW-CONTROL : ENABLE
L2TP SESSION FLOW-CONTROL {ENABLE | DISABLE}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **l2tp session flow-control disable** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display session flow-control** SESSION FLOW-CONTROL : DISABLE
L2TP SESSION PACKET-SEQUENCING {ENABLE | DISABLE}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **l2tp session packet-sequencing enable** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display session packet-sequencing** SESSION PACKET-SEQUENCING : ENABLE

Syntax	Example
L2TP SESSION PACKET-SEQUENCING {ENABLE | DISABLE}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **l2tp session packet-sequencing disable** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display session packet-sequencing** SESSION PACKET-SEQUENCING : DISABLE
[NO] L2TP SESSION RECEIVE-WINDOW-SIZE <value>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **l2tp session receive-window-size 50** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display session receive-window-size** TUNNEL RECEIVE-WINDOW-SIZE : 10 SESSION RECEIVE-WINDOW-SIZE : 50
[NO] L2TP SESSION RECEIVE-WINDOW-SIZE <value>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **no l2tp session receive-window-size** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display session receive-window-size** TUNNEL RECEIVE-WINDOW-SIZE : 10 SESSION RECEIVE-WINDOW-SIZE : 10
[NO] L2TP SESSION TRANSMIT-QUEUE-SIZE <value>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **l2tp session transmit-queue-size 55** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display session transmit-queue** SESSION TRANSMIT-QUEUE-SIZE : 55
[NO] L2TP SESSION TRANSMIT-QUEUE-SIZE <value>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **no l2tp session transmit-queue-size** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display session transmit-queue** SESSION TRANSMIT-QUEUE-SIZE : 5
L2TP TUNNEL AUTHENTICATION {ENABLE | DISABLE}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **l2tp tunnel authentication enable** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display tunnel authentication** TUNNEL(s): List is empty TUNNEL AUTHENTICATION : ENABLE PPP AUTHENTICATION : NONE

Syntax	*Example*
L2TP TUNNEL AUTHENTICATION {ENABLE | DISABLE}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **l2tp tunnel authentication disable** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display tunnel authentication** TUNNEL(s): List is empty TUNNEL AUTHENTICATION : DISABLE PPP AUTHENTICATION : NONE
[NO] L2TP TUNNEL EMPTY-TUNNEL-TIMEOUT <seconds>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **l2tp tunnel EMPTY-TUNNEL-TIMEOUT 225** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display tunnel EMPTY-TUNNEL-TIMEOUT** TUNNEL(s): List is empty TUNNEL EMPTY-TUNNEL-TIMEOUT : 225
[NO] L2TP TUNNEL EMPTY-TUNNEL-TIMEOUT <seconds>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **no l2tp tunnel EMPTY-TUNNEL-TIMEOUT** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display tunnel EMPTY-TUNNEL-TIMEOUT** TUNNEL(s): List is empty TUNNEL EMPTY-TUNNEL-TIMEOUT : 120
[NO] L2TP TUNNEL HELLO <seconds>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **l2tp tunnel hello 120** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display tunnel hello** TUNNEL(s): List is empty TUNNEL HELLO : 120
[NO] L2TP TUNNEL HELLO <seconds>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **no l2tp tunnel hello** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display tunnel hello** TUNNEL(s): List is empty TUNNEL HELLO : 60

Syntax	*Example*
[NO] L2TP TUNNEL MAX-RETRANSMISSION <value>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **l2tp tunnel max-retransmission 25** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display tunnel max-retransmission** TUNNEL(s): List is empty TUNNEL MAX-RETRANSMISSION : 25
[NO] L2TP TUNNEL MAX-RETRANSMISSION <value>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **no l2tp tunnel max-retransmission** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display tunnel max-retransmission** TUNNEL(s): List is empty TUNNEL MAX-RETRANSMISSION : 5
[NO] L2TP TUNNEL RECEIVE-WINDOW-SIZE <value>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **l2tp tunnel receive-window-size 50** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display tunnel receive-window-size** TUNNEL(s): List is empty TUNNEL RECEIVE-WINDOW-SIZE : 50 SESSION RECEIVE-WINDOW-SIZE : 10
[NO] L2TP TUNNEL RECEIVE-WINDOW-SIZE <value>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **no l2tp tunnel receive-window-size** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display tunnel receive-window-size** TUNNEL(s): List is empty TUNNEL RECEIVE-WINDOW-SIZE : 10 SESSION RECEIVE-WINDOW-SIZE : 10
[NO] L2TP TUNNEL RETRANSMISSION-TIMEOUT <value>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **l2tp tunnel retransmission-timeout 25** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display tunnel retransmission-timeout** TUNNEL(s): List is empty TUNNEL RETRANSMISSION-TIMEOUT : 25

Syntax	*Example*
[NO] L2TP TUNNEL RETRANSMISSION-TIMEOUT <value>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **no l2tp tunnel retransmission-timeout** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display tunnel retransmission-timeout** TUNNEL(s): List is empty TUNNEL RETRANSMISSION-TIMEOUT : 16

Default Value

See Table 16-11

Command Level

l2tp

Related Commands

None

lac

Enable or disable the L2TP access concentrator (LAC). Table 16-13 describes the LAC administration status parameters. You must set the VR **hostname** and the **L2TP service-vi** before you enable the LAC. Refer to "hostname" on page 16-60 and "l2tp" on page 16-82 for more details. Also, this command will automatically commit L2TP and LAC status change.

Table 16-12 LAC Admin Status Parameters

Parameter	*Description*
Enable	Allows creation of tunnels and sessions.
Disable	Deletes all tunnels and sessions. No new tunnels or sessions can be created.
Drain	No new tunnels or sessions can be created. Current tunnels and sessions are unaffected.

Syntax	*Example*
LAC ADMIN-STATUS {ENABLE | DRAIN | DISABLE}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **lac admin-status enable** L2TP: Object successfully set LAC: Object successfully set IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display lac** LAC ADMIN-STATUS : ENABLE
LAC ADMIN-STATUS {ENABLE | DRAIN | DISABLE}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **lac admin-status drain** L2TP: Object successfully set LAC: Object successfully set IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display lac** LAC ADMIN-STATUS : DRAIN
LAC ADMIN-STATUS {ENABLE | DRAIN | DISABLE}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **lac admin-status disable** L2TP: Object successfully set LAC: Object successfully set IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display lac** LAC ADMIN-STATUS : DISABLE

Default Value

See Table 16-13

Command Level

l2tp

Related Commands

None

Ins

Enable or disable the L2TP network server (LNS). Table 16-13 describes the LNS administration status parameters. You must set the VR **hostname** and the **L2TP service-vi** before you enable the LNS. Refer to “hostname” on page 16-60 and “l2tp” on page 16-82 for more details. Also, this command will automatically commit L2TP and LNS status change.

Table 16-13 LNS Admin Status Parameters

Parameter	*Description*
Enable	Allow creation of tunnels and sessions.
Disable	Delete all tunnels and sessions. No new tunnels or sessions can be created.
Drain	No new tunnels or sessions can be created. Current tunnels and sessions are unaffected.

Syntax	*Example*
LNS ADMIN-STATUS {ENABLE | DRAIN | DISABLE}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **lns admin-status enable** L2TP: Object successfully set LNS: Object successfully set IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display lns** LNS ADMIN-STATUS : ENABLE
LNS ADMIN-STATUS {ENABLE | DRAIN | DISABLE}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **lns admin-status drain** L2TP: Object successfully set LNS: Object successfully set IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display lns** LNS ADMIN-STATUS : DRAIN
LNS ADMIN-STATUS {ENABLE | DRAIN | DISABLE}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **lns admin-status disable** L2TP: Object successfully set LNS: Object successfully set IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display lns** LNS ADMIN-STATUS : DISABLE

Default Value

See Table 16-13

Command Level

l2tp

Related Commands

None

ppp

Configure Point-to-Point Protocol (PPP) to run under L2TP. Use the **no** option to disable a command or to reset it to its default value. Table 16-14 describes the parameters available. These commands function exactly like the PPP encapsulation commands. Refer to "ppp" on page 14-7 for more details.

Table 16-14 PPP Parameters for L2TP

Parameter	*Description*
Authentication	Sets the authentication type. Choose from None, PAP, CHAPMD5, or MSCHAPV1. If you choose CHAPMD5 or MSCHAPV1, you need to use a RADIUS server. Refer to "pptp" on page 16-115 and "radius-server" on page 16-122 for more information. The default value is none.
Config-req-retransmit	Sets the number of configuration request retransmissions. Choose from 0 to 64. The default value is 10 seconds.
IP-local-pool-name	Gives the name of the IP address pool. Refer to"ip" on page 16-68 for setting up an IP address pool. This parameter is required for LNS.
Keepalive	Sets the interval for sending keepalive packets. The default value is 10 seconds.
Reauth	Sets the reauthentication request interval. Choose from 0 to 65535 seconds. The default value is 5 seconds.
Term-req-retransmit	Sets the number of termination request retransmissions. Choose from 0 to 64. The default value is 2 retransmissions.
Timeout	Sets a limit to retransmitting a packet. Choose from 1 to 64. The default value is 3 attempts.

Syntax	*Example*
PPP AUTHENTICATION {PAP | CHAPMD5 | MSCHAPV1}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **ppp authentication pap** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display authentication** TUNNEL AUTHENTICATION : DISABLE PPP AUTHENTICATION : PAP

Syntax	*Example*
PPP AUTHENTICATION {PAP | CHAPMD5 | MSCHAPV1}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **ppp authentication chapmd5** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display authentication** TUNNEL AUTHENTICATION : DISABLE PPP AUTHENTICATION : CHAPMD5
PPP AUTHENTICATION {PAP | CHAPMD5 | MSCHAPV1}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **ppp authentication mschapv1** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display authentication** TUNNEL AUTHENTICATION : DISABLE PPP AUTHENTICATION : MSCHAPV1
NO PPP AUTHENTICATION	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **no ppp authentication** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display authentication** TUNNEL AUTHENTICATION : DISABLE PPP AUTHENTICATION : NONE
PPP CONFIG-REQ-RETRANSMIT <value>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **ppp config-req-retransmit 200** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display config** PPP CONFIG-REQ-RETRANSMIT : 200
NO PPP CONFIG-REQ-RETRANSMIT	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **no ppp config-req-retransmit** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display config** PPP CONFIG-REQ-RETRANSMIT : 10
PPP IP-LOCAL-POOL-NAME <value>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **ppp ip-local-pool-name pool1** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display ip-local-pool-name** PPP IP-LOCAL-POOL-NAME : pool1
NO PPP IP-LOCAL-POOL-NAME	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **no ppp ip-local-pool-name** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display ip-local-pool-name** PPP IP-LOCAL-POOL-NAME :

Syntax	Example
PPP KEEPALIVE <interval-in-secs>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **ppp keepalive 15000** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display keepalive** PPP KEEPALIVE : 15000
NO PPP KEEPALIVE	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **no ppp keepalive** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display keepalive** PPP KEEPALIVE : 10
PPP REAUTH <interval-in-secs>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **ppp reauth 100** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display reauth** PPP REAUTH : 100
NO PPP REAUTH	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **no ppp reauth** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display reauth** PPP REAUTH : 5
PPP TERM-REQ-RETRANSMIT <value>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **ppp term-req-retransmit 50** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display term** PPP TERM-REQ-RETRANSMIT : 50
NO PPP TERM-REQ-RETRANSMIT	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **no ppp term-req-retransmit** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display term** PPP TERM-REQ-RETRANSMIT : 2
PPP TIMEOUT <timeout-in-secs>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **ppp timeout 500** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display timeout** PPP TIMEOUT : 500
NO PPP TIMEOUT	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **no ppp timeout** IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display timeout** PPP TIMEOUT : 5

Default Value

See Table 16-14

Command Level

L2TP

Related Commands

ppp (ppp commands)
pptp (pptp commands)

tunnel

View or tear down a tunnel. This command works only when someone has dialed in remotely and initiated a session and started a tunnel. Each tunnel is assigned a unique identifier when it is created. Use this ID for accessing the tunnel. This command also puts you into the L2TP tunnel level. Use the **no** option to delete a tunnel. The following commands are available at this level:

- "display" on page 16-97
- "session" on page 16-98

Syntax	*Example*
[NO] TUNNEL <tunnel-id>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display tunnel** TUNNEL(s): LOCAL-ID | PEER-IP-ADDRESS | SESSION-COUNT | UP-TIME (sec) 45268 | 192.168.121.232 | 1 | 179 IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **tunnel 45268** L2TP: Object successfully set IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP TUNNEL(45268)#
[NO] TUNNEL <tunnel-id>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **no tunnel 45268** 0000b45a: L2TP[1/2.2.2.2]: Tunnel 45268 torn down IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP# **display tunnel** TUNNEL(s): List is empty

Default Value

None

Command Level

L2TP

Related Commands

None

display

View the current status of a specific L2TP tunnel.

Syntax	*Example*

DISPLAY	`IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP TUNNEL(45268)# display`
	`VPN : 100`
	`VR : 1.1.1.1`
	`LOCAL-TUNNEL-ID : 45268`
	
	`AUTHENTICATION : ENABLE`
	`HELLO : 60`
	`MAX-RETRANSMISSION : 5`
	`RETRANSMISSION-TIMEOUT : 16`
	`RECEIVE-WINDOW-SIZE : 10`
	`EMPTY-TUNNEL-TIMEOUT : 120`
	`STATE : ESTABLISHED`
	`UP-TIME : 203`
	
	`PEER-TUNNEL-ID : 2`
	`PEER-IP-ADDRESS : 192.168.121.232`
	`PEER-UDP-PORT : 1701`
	`PEER-RECEIVE-WINDOW-SIZE : 4`
	`PEER-HOSTNAME : iVasionVPN`
	
	`SESSION(s):`
	`LOCAL-ID | PEER-ID | USER-NAME | CALL-TYPE | UP-TIME (sec)`
	`1 | 501 | joe | LAC-INCOMING | 2477`
	`2 | 502 | sam | LNS-INCOMING | 1906`

Default Value

None

Command Level

L2TP tunnel

Related Commands

l2tp display

session

View or tear down a session. This command works only when someone has dialed in remotely and initiated a session and started a tunnel. Each session is assigned a unique identifier when it is created. Use this ID for accessing the session. Use the **no** option to delete a session. Currently, only LAC-incoming and LNS-incoming session call types are supported. An external device must initate the call to the IPSX switch.

Syntax	*Example*
[NO] SESSION <session-id>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP TUNNEL(45268)# **session 1** IP-ADDRESS : 2.2.2.2 USERNAME : sam FLOW-CONTROL : ENABLE RECEIVE-WINDOW-SIZE : 10 TRANSMIT-QUEUE-SIZE : 5 PACKET-SEQUENCING : DISABLE CALL-TYPE : LAC-INCOMING PEER-SESSION_ID : 8 PEER-IP-ADDRESS : 10.10.10.2 PEER-RECEIVE-WINDOW-SIZE : 10 STATE : ESTABLISHED UP-TIME : 2485
[NO] SESSION <session-id>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP TUNNEL(45268)# **no session 1** SESSION 1: Object successfully deleted IPNOS : CONFIG VPN(50) VR(2.2.2.2) L2TP TUNNEL(45268)# **display session** SESSION(s): LOCAL-ID | PEER-ID | USER-NAME | CALL-TYPE | UP-TIME (sec) 2 | 502 | sam | LNS-INCOMING | 1906

Default Value

None

Command Level

L2TP tunnel

Related Commands

None

tunnel-switch

Go to tunnel switch command level. A tunnel switch is a prioritized list of tunnels to other LNSs. Sessions are forwarded to the highest priority tunnel until the session limit is reached for that tunnel. Then sessions are forwarded to the next highest priority tunnel. Refer to Figure 16-5 for details. Use the **no** option to delete a tunnel. The following commands are available at this level.

- "display" on page 16-100
- "mode" on page 16-100
- "switch-to" on page 16-103

Syntax	*Example*
`TUNNEL-SWITCH <tunnel-client-name>`	`IPNOS: CONFIG VPN(50) VR(2.2.2.2) L2TP#` **`tunnel-switch LAC-1`** `IPNOS: CONFIG VPN(50) VR(2.2.2.2) L2TP TUNNEL-SWITCH(LAC-1)#`

Default Value

None

Command Level

L2TP

Related Commands

None

display

View the current configuration of the tunnel switch.

Syntax	*Example*

DISPLAY	`IPNOS: CONFIG VPN(50) VR(2.2.2.2) L2TP TUNNEL-SWITCH(LAC-1)#` **`display`**
	`VPN : 50`
	`VR : 2.2.2.2`
	`CLIENT-NAME : LAC-1`
	`MODE : PASS-THROUGH`
	
	`SWITCH-TO:`
	`ADDRESS | PRIORITY | LIMIT | SESSION-COUNT`
	`13.13.11.90 | 1 | 256 | 100`
	`13.13.13.56 | 2 | 30 | 12`
	`13.13.13.100 | 3 | 100 | 0`
	`13.13.140.50 | 65535 | 256 | 0`

Default Value

None

Command Level

tunnel-switch

Related Commands

None

mode

Set the tunnel switch mode. Table 16-15 describes the mode parameters.

Table 16-15 L2TP LAC Mode Parameters

Parameter	*Description*
Pass-through	Simply forward sessions to the LNS, without user authentication.
Authenticate	Authenticate the user before forwarding the session to the LNS.

Syntax	*Example*
MODE {PASS-THROUGH | AUTHENTICATE}	IPNOS: CONFIG VPN(50) VR(2.2.2.2) L2TP TUNNEL-SWITCH(LAC-1)# **mode pass-through** IPNOS: CONFIG VPN(50) VR(2.2.2.2) L2TP TUNNEL-SWITCH(LAC-1)# **display** VPN : 50 VR : 2.2.2.2 CLIENT-NAME : LAC-1 MODE : PASS-THROUGH SWITCH-TO: ADDRESS | PRIORITY | LIMIT | SESSION-COUNT 13.13.11.90 | 1 | 256 | 100 13.13.13.56 | 2 | 30 | 12 13.13.13.100 | 3 | 100 | 0 13.13.140.50 | 65535 | 256 | 0 IPNOS: CONFIG VPN(50) VR(2.2.2.2) L2TP TUNNEL-SWITCH(LAC-1)# **commit** Commit configuration TUNNEL-SWITCH(LAC-1): Object successfully set IPNOS: CONFIG VPN(50) VR(2.2.2.2) L2TP#

Syntax	*Example*
MODE {PASS-THROUGH | AUTHENTICATE}	IPNOS: CONFIG VPN(50) VR(2.2.2.2) L2TP TUNNEL-SWITCH(LAC-1)# **mode authenticate** IPNOS: CONFIG VPN(50) VR(2.2.2.2) L2TP TUNNEL-SWITCH(LAC-1)# **display** VPN : 50 VR : 2.2.2.2 CLIENT-NAME : LAC-1 MODE : AUTHENTICATE SWITCH-TO: ADDRESS | PRIORITY | LIMIT | SESSION-COUNT 13.13.11.90 | 1 | 256 | 100 13.13.13.56 | 2 | 30 | 12 13.13.13.100 | 3 | 100 | 0 13.13.140.50 | 65535 | 256 | 0 IPNOS: CONFIG VPN(50) VR(2.2.2.2) L2TP TUNNEL-SWITCH(LAC-1)# **commit** Commit configuration TUNNEL-SWITCH(LAC-1): Object successfully set IPNOS: CONFIG VPN(50) VR(2.2.2.2) L2TP#
NO MODE	IPNOS: CONFIG VPN(50) VR(2.2.2.2) L2TP TUNNEL-SWITCH(LAC-1)# **no mode** IPNOS: CONFIG VPN(50) VR(2.2.2.2) L2TP TUNNEL-SWITCH(LAC-1)# **display** VPN : 50 VR : 2.2.2.2 CLIENT-NAME : LAC-1 MODE : PASS-THROUGH SWITCH-TO: ADDRESS | PRIORITY | LIMIT | SESSION-COUNT 13.13.11.90 | 1 | 256 | 100 13.13.13.56 | 2 | 30 | 12 13.13.13.100 | 3 | 100 | 0 13.13.140.50 | 65535 | 256 | 0 IPNOS: CONFIG VPN(50) VR(2.2.2.2) L2TP TUNNEL-SWITCH(LAC-1)# **commit** Commit configuration TUNNEL-SWITCH(LAC-1): Object successfully set IPNOS: CONFIG VPN(50) VR(2.2.2.2) L2TP#

Default Value

pass-through

Command Level

tunnel-switch

Related Commands

None

switch-to

Configure or remove a target tunnel for switching. Set the priority (1–65,535) of the tunnel and the limit its size (1–65,535 sessions). When the highest priority tunnel hits the session limit, the next priority tunnel starts receiving tunnels. When the last tunnel is full, the tunnel switch will attempt to forward sessions to the first tunnel. If tunnels have the same priority, the one with the lower IP address will get the sessions first.

IPNOS will still try to create sessions with a dead tunnel until the session limit is reached. Then the next priority tunnel will be tried. There is currently no method for detecting dead tunnels.

Table 16-16 describes the **switch-to** parameters.

Table 16-16 L2TP switch-to parameters

Parameter	*Description*
IP Address	IP address of the tunnel.
Limit	Number of L2TP sessions allowed in the tunnel.
Priority	Priority of the tunnel.

Syntax	*Example*
SWITCH-TO <ip-address> [LIMIT <limit-number>] [PRIORITY <priority-number>]	IPNOS: CONFIG VPN(50) VR(2.2.2.2) L2TP TUNNEL-SWITCH(LAC-1)# **switch-to 13.13.140.50 limit 256 priority 65535** TUNNEL-SWITCH(13.13.140.50): Object successfully set IPNOS: CONFIG VPN(50) VR(2.2.2.2) L2TP TUNNEL-SWITCH(LAC-1)# **display** VPN : 50 VR : 2.2.2.2 CLIENT-NAME : lac-1 MODE : PASS-THROUGH SWITCH-TO: ADDRESS | PRIORITY | LIMIT | SESSION-COUNT 13.13.11.90 | 1 | 256 | 100 13.13.13.56 | 2 | 30 | 12 13.13.13.100 | 3 | 100 | 0 13.13.140.50 | 65535 | 256 | 0
NO SWITCH-TO <ip-address>	IPNOS: CONFIG VPN(50) VR(2.2.2.2) L2TP TUNNEL-SWITCH(lac-1)# **no switch-to 13.13.140.50** TUNNEL-SWITCH(13.13.140.50): Object successfully set IPNOS: CONFIG VPN(50) VR(2.2.2.2) L2TP TUNNEL-SWITCH(lac-1)# **display** VPN : 50 VR : 2.2.2.2 CLIENT-NAME : lac-1 MODE : PASS-THROUGH SWITCH-TO: ADDRESS | PRIORITY | LIMIT | SESSION-COUNT 13.13.11.90 | 1 | 256 | 100 13.13.13.56 | 2 | 30 | 12 13.13.13.100 | 3 | 100 | 0

Default Value

None

Command Level

tunnel-switch

Related Commands

None

name

Give the VR a name. Use only alphanumeric characters.

NOTE: This command has no relationship to the commands "hostname" on page 16-60 and "username" on page 16-165.

Syntax	*Example*

`NAME <string>`	`IPNOS : CONFIG VPN(50) VR(10.1.1.1)# name CoSine`
	
	`IPNOS : CONFIG VPN(50) VR(10.1.1.1)# display name`
	`NAME : CoSine`

Default Value

None

Command Level

VR

Related Commands

sysname

nbns

Set the primary and secondary domain name server for Microsoft Windows name service (WINS). Also known as the Net BIOS name service (NBNS), the name server maintains a list of domain names in a network. Refer to "dns" on page 16-43 for the non-Microsoft version of the domain name service feature.

Syntax	*Example*
NBNS {PRIMARY | SECONDARY} address	IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **nbns primary 192.167.24.1** VR(1.1.1.1): Object successfully set NBNS SERVER 192.167.24.1: Object successfully set IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **display wins** WINS PRIMARY : 192.167.24.1 WINS SECONDARY : 0.0.0.0 IPNOS: CONFIG VPN(50) VR(1.1.1.1)#
NBNS {PRIMARY | SECONDARY} address	IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **nbns secondary 192.167.24.2** VR(1.1.1.1): Object successfully set NBNS SERVER 192.167.24.2: Object successfully set IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **display wins** WINS PRIMARY : 192.167.24.1 WINS SECONDARY : 192.167.24.2 IPNOS: CONFIG VPN(50) VR(1.1.1.1)#

Default Value

None

Command Level

VR

Related Commands

dns
wins

packet-filter

Create and manage all packet filters within the virtual router. A packet filter sets the rules for deciding whether to drop or forward a packet. This filter is applied to all virtual interfaces (VIs) on this VR. Use the **no** option to delete a packet filter. You must create a general access control list (ACL) before referring to it in the packet filter. Refer to "acl" on page 16-7 for information on creating ACLs. Table 16-17 describes the packet-filter parameters.

Table 16-17 VR Packet Filter Parameters

Parameter	*Description*
Permit	Permit the packet into the VR.
Deny	Deny the packet access to the VR. Drop the packet.
Firewall	Apply a firewall proxy to the packet. *Note:* You need to configure the appropriate firewall proxy.
Inbound	Affect inbound packets.
Outbound	Affect outbound packets.
GenAclId	Affect packets coming from the selected range of IP addresses stated in the ACL.
Counter	Enable a packet filter counter. Every time a packet hits a particular filter, the filter's counter increments by one. When the counter reaches 2,147,483,647 (2 billion), the counter resets to zero. To see the current count, type **display packet-filter**. If event logs are enabled, the packet filter count is also sent to the syslog daemon (syslogd) server. The default value is disabled. *WARNING:* When enabled, the counting operation slows down IPNOS. Use this parameter for debugging purposes only.
Event-log	Enable packet filter event logs. Every time a packet hits a particular filter, an event log is generated. To reduce chances of a denial of service attack (DoS), a limit of 100 packet filter event logs are generated per filter in a ten second interval. These event logs are then sent to the syslog server. Refer to "syslogd" on page 16-151 for information on enabling the syslog daemon (syslogd) server. Also, the counter is enabled (or reset to zero if it is already enabled) when packet filter event logs are enabled. The default value is disabled. *Note:* Use **no packet-filter deny event-log** command to enable packet filter event logs for all VR and VI packet filters configured to deny packets. *WARNING:* When enabled, the event log generation slows down IPNOS dramatically. Use this parameter for debugging purposes only.

Syntax	*Example*
[NO] PACKET-FILTER <filterId> {PERMIT | DENY | FIREWALL} {INBOUND | OUTBOUND} <genAclId> [COUNTER | EVENT-LOG]	IPNOS : CONFIG VPN(50) VR(10.10.10.1)# **packet-filter 1 permit inbound 1** VR(3.3.3.3): Object successfully set PACKET-FILTER 1: Object successfully set IPNOS: CONFIG VPN(50) VR(10.10.10.1)# **display packet** PACKET-FILTER DENY EVENT-LOG: DISABLE PACKET-FILTER(s): PACKET-FILTER | ACTION | DIRECTION | ACL | EVENT-LOG | COUNTER 1 | PERMIT | INBOUND | 1 | DISABLE | 0
[NO] PACKET-FILTER <filterId> {PERMIT | DENY | FIREWALL} {INBOUND | OUTBOUND} <genAclId> [COUNTER | EVENT-LOG]	IPNOS : CONFIG VPN(50) VR(10.10.10.1)# **packet-filter 2 deny inbound 2** VR(3.3.3.3): Object successfully set PACKET-FILTER 2: Object successfully set IPNOS: CONFIG VPN(50) VR(10.10.10.1)# **display packet** PACKET-FILTER DENY EVENT-LOG: DISABLE PACKET-FILTER(s): PACKET-FILTER | ACTION | DIRECTION | ACL | EVENT-LOG | COUNTER 1 | PERMIT | INBOUND | 1 | DISABLE | 0 2 | DENY | INBOUND | 2 | DISABLE | 0
[NO] PACKET-FILTER <filterId> {PERMIT | DENY | FIREWALL} {INBOUND | OUTBOUND} <genAclId> [COUNTER | EVENT-LOG]	IPNOS : CONFIG VPN(50) VR(10.10.10.1)# **packet-filter 3 firewall inbound 3 counter** VR(3.3.3.3): Object successfully set PACKET-FILTER 3: Object successfully set IPNOS: CONFIG VPN(50) VR(10.10.10.1)# **display packet** PACKET-FILTER DENY EVENT-LOG: DISABLE PACKET-FILTER(s): PACKET-FILTER | ACTION | DIRECTION | ACL | EVENT-LOG | COUNTER 1 | PERMIT | INBOUND | 1 | DISABLE | 0 2 | DENY | INBOUND | 2 | DISABLE | 0 3 | FIREWALL | INBOUND | 3 | COUNTER | 0

Syntax	*Example*
[NO] PACKET-FILTER <filterId> {PERMIT | DENY | FIREWALL} {INBOUND | OUTBOUND} <genAclId> [COUNTER | EVENT-LOG]	IPNOS : CONFIG VPN(50) VR(10.10.10.1)# **packet-filter 4 deny outbound 4 event-log** VR(3.3.3.3): Object successfully set PACKET-FILTER 4: Object successfully set IPNOS: CONFIG VPN(50) VR(10.10.10.1)# **display packet** PACKET-FILTER DENY EVENT-LOG: DISABLE PACKET-FILTER(s): PACKET-FILTER | ACTION | DIRECTION | ACL | EVENT-LOG | COUNTER 1 | PERMIT | INBOUND | 1 | DISABLE | 0 2 | DENY | INBOUND | 2 | DISABLE | 0 3 | FIREWALL | INBOUND | 3 | COUNTER | 0 4 | DENY | OUTBOUND | 4 | EVENT-LOG | 0
[NO] PACKET-FILTER <filterId> {PERMIT | DENY | FIREWALL} {INBOUND | OUTBOUND} <genAclId> [COUNTER | EVENT-LOG]	IPNOS : CONFIG VPN(50) VR(10.10.10.1)# **packet-filter 5 firewall inbound 5 event-log** VR(3.3.3.3): Object successfully set PACKET-FILTER 5: Object successfully set IPNOS: CONFIG VPN(50) VR(10.10.10.1)# **display packet** PACKET-FILTER DENY EVENT-LOG: DISABLE PACKET-FILTER(s): PACKET-FILTER | ACTION | DIRECTION | ACL | EVENT-LOG | COUNTER 1 | PERMIT | INBOUND | 1 | DISABLE | 0 2 | DENY | INBOUND | 2 | DISABLE | 0 3 | FIREWALL | INBOUND | 3 | COUNTER | 0 4 | DENY | OUTBOUND | 4 | EVENT-LOG | 0 5 | FIREWALL | INBOUND | 5 | EVENT-LOG | 0
[NO] PACKET-FILTER <filterId> {PERMIT | DENY | FIREWALL} {INBOUND | OUTBOUND} <genAclId> [COUNTER | EVENT-LOG]	IPNOS : CONFIG VPN(50) VR(10.10.10.1)# **no packet-filter 5** VR(3.3.3.3): Object successfully set PACKET-FILTER 5: Object successfully deleted IPNOS: CONFIG VPN(50) VR(10.10.10.1)# **display packet** PACKET-FILTER DENY EVENT-LOG: DISABLE PACKET-FILTER(s): PACKET-FILTER | ACTION | DIRECTION | ACL | EVENT-LOG | COUNTER 1 | PERMIT | INBOUND | 1 | DISABLE | 0 2 | DENY | INBOUND | 2 | DISABLE | 0 3 | FIREWALL | INBOUND | 3 | COUNTER | 0 4 | DENY | OUTBOUND | 4 | EVENT-LOG | 0

Syntax	*Example*

Enable event logs for all VR and VI deny packet filters: [NO] PACKET-FILTER DENY EVENT-LOG	`IPNOS : CONFIG VPN(50) VR(10.10.10.1)# no packet-filter deny event-log`
	`VR(3.3.3.3): Object successfully set`
	`PACKET-FILTER 5: Object successfully set`
	
	`IPNOS: CONFIG VPN(50) VR(10.10.10.1)# display packet`
	`PACKET-FILTER DENY EVENT-LOG: ENABLE`
	
	`PACKET-FILTER(s):`
	`PACKET-FILTER | ACTION | DIRECTION | ACL | EVENT-LOG | COUNTER`
	`1 | PERMIT | INBOUND | 1 | DISABLE | 0`
	`2 | DENY | INBOUND | 2 | EVENT-LOG | 0`
	`3 | FIREWALL | INBOUND | 3 | COUNTER | 0`
	`4 | DENY | OUTBOUND | 4 | EVENT-LOG | 0`

Default Value

None

Command Level

VR

Related Commands

packet-filter (VI command)

ping

Check the availability of a specified host. Table 16-18 describes available parameters.

Table 16-18 Ping Parameters

Parameter	*Description*
address	IP address of the destination server.
TIMEOUT	Length of time in seconds before the ping fails. The range is 0–4,294,967 seconds. The default value is 5 seconds.
LENGTH	Length of the test packets. The range is 0–65,535 bytes. The default value is 64 bytes.
COUNT	Number of ping attempts. The range is 0–65,535. The default value is 4 attempts.
TTL	Time to live (TTL) for the ping activity. The range is 0–255 hops. The default value is 30 hops.
SOURCE	IP address of the source server.

Syntax	*Example*

`PING <address> [TIMEOUT <timeout-in-sec>] [LENGTH <send-data-length>] [COUNT <number-of-pings>] [TTL <ttl>] [SOURCE <source-address>]`	`IPNOS : CONFIG VPN(50) VR(10.1.1.1)#` **`ping 192.168.10.10`**
	`vpn 10 : [max tries 4, delay 5.000, data length 64 bytes, ttl 255]`
	`ping #1 timed out`
	`ping #2 timed out`
	`ping #3 timed out`
	`ping #4 timed out`
	`[finished]`

Default Value

None

Command Level

VR

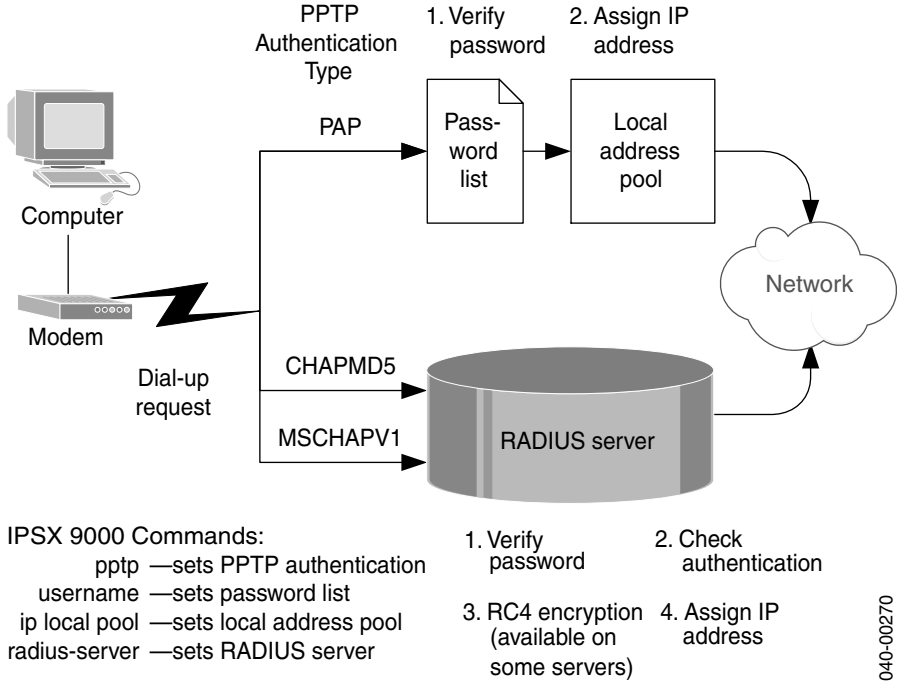
Related Commands

traceroute

pptp

Go to the Microsoft Point-to-Point Tunneling Protocol (PPTP) command level. Up to 128 PCs can dial up to a VR using PPTP. Up to five dialup sessions are supported per PC. Figure 16-6 illustrates the PPTP process.

Figure 16-6 PPTP Process



PPTP does not work on a control blade. The following commands are available.

- "display" on page 16-114
- "pptp" on page 16-115

Syntax	*Example*
PPTP SERVER	IPNOS : CONFIG VPN(50) VR(2.2.2.2)# **pptp server** VR(2.2.2.2): Object successfully set IPNOS : CONFIG VPN(50) VR(2.2.2.2) SERVER#

Default Value

None

Command Level

VR

Related Commands

ip
l2tp
radius-server
username

display

View the current status of the PPTP server.

Syntax	*Example*

DISPLAY	`IPNOS : CONFIG VPN(50) VR(2.2.2.2) SERVER# display`
	`VPN : 50`
	`VR : 2.2.2.2`
	`PPTP : ENABLE`
	`IP-POOL : testpool`
	`ENCRYPTION : NONE`
	`AUTHENTICATION : CHAPMD5`
	`IDLE-TIMEOUT : 300`
	`MAX-CONN : 1024`
	
	`PPTP CONNECTION(s):`
	`FAR |LOCAL |AGE`
	`192.128.55.5 |192.128.55.4 |14`
	`192.128.55.11 |192.128.55.10 |5`

Default Value

None

Command Level

pptp server

Related Commands

show

pptp

Configure elements of the PPTP server. Table 16-19 describes parameters of the **pptp** command. Use the **no** option to reset the command to its default value.

Table 16-19 PPTP Server Parameters

Parameter	*Description*
Authentication	Authentication algorithm used by the PPTP server. The following options are available: • None—No authentication and no encryption (for testing purposes). • Password Authentication Protocol (PAP)—Simple authentication for local address pools. No encryption. • Challenge-Handshake Authentication Protocol using Message Digest 5 authentication (CHAPMD5)— This option requires a RADIUS server. When set to CHAPMD5, the PPTP server automatically forwards client authentication requests to the RADIUS server for authentication. No encryption. • Microsoft CHAP (MSCHAPV1)— An enhanced version of CHAP using RC4 encryption. You must select strong or weak encryption. (Available on some RADIUS servers).
Connection	Clear a PPTP connection. State the far end IP address and then the local IP address. *Note:* You need to set VI1 to loopback before enabling PPTP.
Encryption	Encryption setting for connections. The following settings are available: • Strong (128-bit encryption) • Weak (40-bit encryption)
Idle-timeout	The amount of time with no network activity that elapses before a PPTP server will disconnect a PPTP client.
IP-pool	IP address pool for dial-up connections. This parameter is required. Refer to “ip” on page 16-68 for more information. If this parameter is not used, PPTP goes to the RADIUS server for IP addresses. Refer to “radius-server” on page 16-122 for more information.
Max-conn	The maximum number of simultaneous PPTP connections allowed by the server (a maximum of 1024).

Syntax	*Example*
Enable PAP authentication: PPTP AUTHENTICATION {PAP | CHAPMD5 | MSCHAPV1}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) SERVER# **pptp authentication pap** IPNOS : CONFIG VPN(50) VR(2.2.2.2) SERVER# **display** VPN : 50 VR : 2.2.2.2 PPTP : ENABLE IP-POOL : ENCRYPTION : NONE AUTHENTICATION : PAP IDLE-TIMEOUT : 300 MAX-CONN : 1024 PPTP CONNECTION(s): List is empty
Disable authentication: NO PPTP AUTHENTICATION	IPNOS : CONFIG VPN(50) VR(2.2.2.2) SERVER# **no pptp authentication** IPNOS : CONFIG VPN(50) VR(2.2.2.2) SERVER# **display** VPN : 50 VR : 2.2.2.2 PPTP : ENABLE IP-POOL : ENCRYPTION : NONE AUTHENTICATION : NONE IDLE-TIMEOUT : 300 MAX-CONN : 1024 PPTP CONNECTION(s): List is empty
NO PPTP CONNECTION <far-address> <local-address>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) SERVER# **no pptp connection 2.2.2.2 1.1.1.1**
PPTP ENCRYPTION {STRONG | WEAK} -or- NO PPTP ENCRYPTION	IPNOS : CONFIG VPN(50) VR(2.2.2.2) SERVER# **pptp encryption strong**
PPTP IDLE-TIMEOUT <value-in-secs> -or- NO PPTP IDLE-TIMEOUT	IPNOS : CONFIG VPN(50) VR(2.2.2.2) SERVER# **pptp idle-timeout 20**
PPTP IP-POOL <name> -or- NO PPTP IP-POOL	IPNOS : CONFIG VPN(50) VR(2.2.2.2) SERVER# **pptp ip-pool 2**
PPTP MAX-CONN <value> -or- NO PPTP MAX-CONN	IPNOS : CONFIG VPN(50) VR(2.2.2.2) SERVER# **pptp max-conn 100**

Default Value

None

Command Level

VR

Related Commands

ip
l2tp (l2tp commands)
ppp (ppp commands)
radius-server

qos

Go to the quality of service (QoS) command level. The following commands are available.

- "display" on page 16-118
- "ds-byte-filter" on page 16-119

Syntax	*Example*

QOS	`IPNOS : CONFIG VPN(50) VR(2.2.2.2)# qos`
	`QOS`
	`VR(2.2.2.2): Object successfully set`
	
	`IPNOS : CONFIG VPN(50) VR(2.2.2.2) QOS#`

Default Value

None

Command Level

VR

Related Commands

None

display

View the current status of QoS features.

Syntax	*Example*

DISPLAY	`IPNOS : CONFIG VPN(50) VR(2.2.2.2) QOS# display`
	
	`DS-BYTE-FILTER(s):`
	`DS-BYTE-FILTER | ACTION | VALUE | DIRECTION | ACL`
	`1 | SETVALUE | 32 | OUTBOUND | 1`

Default Value

None

Command Level

QoS

Related Commands

None

ds-byte-filter

Set the differentiated services (DS) field in an IP header for a packet leaving the VR. Similar in function to the **ds-byte** and **packet-filter** commands, the **ds-byte-filter** command extends the granularity of setting a packet's DS bits. For each entry, one of two actions must be selected: SET the DS bits field to the specified value or IGNORE the DS bits field and leave it at its original value. Use RFC 1349 and 2474 to calculate the appropriate byte value. Use the **no** option to disable the command. Table 16-20 describes the **ds-byte-filter** parameters.

Table 16-20 ds-byte-filter Parameters

Parameter	*Description*
FilterID	ID of the packet filter.
Setvalue	Define the values to be placed in the Type Of Service octet of each IP packet sent out or forwarded by the VR. The valid range covers both 7-bit configurations recommended in RFC 1349 (Precedence and TOS bits) and 6-bit configurations recommended in RFC 2474 (Differentiated Services). If the value selected is 256, the ds-byte-filter is not changed and packets are forwarded with their existing Type Of Service byte value. Table 16-21 describes the pre-set values used for compatibility with non-CoSine routers:
Inbound | Outbound	Direction of packets on which this filter is applied. Currently only outbound packets are filtered.
genACLId	ACL ID. All packets transmitted or received are acted upon by this filter.

Table 16-21 Possible Values for IP Precedence Compatibility

Value	*Description*
0	Routine
32	Priority
64	Immediate
96	Flash
128	Flash Override
160	Critical/ECP
192	Internet Control
224	Network Control

Syntax	*Example*
DS-BYTE-FILTER <filterId> {SETVALUE <value> | IGNORE} {INBOUND | OUTBOUND} <genAclId>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) QOS# **ds-byte-filter 1 setvalue 32 outbound 1** DS-BYTE-FILTER 1: Object successfully set IPNOS : CONFIG VPN(50) VR(2.2.2.2) QOS# **display** DS-BYTE-FILTER(s): DS-BYTE-FILTER | ACTION | VALUE | DIRECTION | ACL 1 | SETVALUE | 32 | OUTBOUND | 1
DS-BYTE-FILTER <filterId> {SETVALUE <value> | IGNORE} {INBOUND | OUTBOUND} <genAclId>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) QOS# **ds-byte-filter 2 ignore outbound 2** DS-BYTE-FILTER 2: Object successfully set IPNOS : CONFIG VPN(50) VR(2.2.2.2) QOS# **display** DS-BYTE-FILTER(s): DS-BYTE-FILTER | ACTION | VALUE | DIRECTION | ACL 1 | SETVALUE | 32 | OUTBOUND | 1 2 | IGNORE | - | OUTBOUND | 2

Syntax	*Example*
NO DS-BYTE-FILTER <filterId>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) QOS# **no ds-byte-filter 2** DS-BYTE-FILTER 2: Object successfully deleted IPNOS : CONFIG VPN(50) VR(2.2.2.2) QOS# **display** DS-BYTE-FILTER(s): DS-BYTE-FILTER | ACTION | VALUE | DIRECTION | ACL 1 | SETVALUE | 32 | OUTBOUND | 1

Default Value

Disabled

Command Level

QoS

Related Commands

ds-byte (VR command)
ds-byte-filter (VI QOS command)
packet-filter (VR command)
packet-filter (VI command)

radius-server

Connect to a remote authentication dial-in user service (RADIUS) server. The RADIUS protocol adds an extra level of authentication security for several IPNOS features such as L2TP, PPTP and IPSec.

NOTE: The RADIUS server connection must be enabled in the same VR as the PPTP, L2TP, or IPSec connection.

Table 16-22 Radius-Server Parameters

Parameter	*Description*
List	Connect the RADIUS server to the VR. Use the **no** option to delete the server from the list. The following parameters are available: • Priority—Set the priority level for the RADIUS server. This sets the order to be called upon by the VR. The lower the number, the higher the priority level. • Address—The IP address of the RADIUS server. • Key—Type a secret key. It is an ASCII password value defined by the administrator. The key can be a maximum of 80 characters in length. It is used as the key value to enable encryption during the RADIUS exchange process. This value is never seen by the client user. • ACCT—The server is an accounting server. You can connect up to six servers of this type. • AUTH—The server is an authentication server. You can connect up to six servers of this type. This is the default value if a server type is not specified. • Port—Type an optional port number. The default server port is used if you do not choose a specific port. The default value for authentication is 1812. The default value for accounting is 1813.
Retransmit	Set the number of retransmission attempts to the RADIUS server. Used to verify that the RADIUS server is alive. The default value is 3. Use the **no** option to reset the command to its default value.
Timeout	Set the time interval (in seconds) between retransmission attempts. The default value is 5. Use the **no** option to reset the command to its default value.

Syntax	Example
Add server with different port: RADIUS-SERVER LIST <priority> <address> <key> [ACCT | AUTH] [<port>]	IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **radius list 1 23.23.23.2 cosine1 auth 2222** VR(1.1.1.1): Object successfully set RADIUS-SERVER LIST 1 AUTH 2222: Object successfully set IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **dis rad** RADIUS-SERVER CONFIG: RETRANSMIT : 3 TIMEOUT : 5 RADIUS-AUTHENTICATION-SERVER(s): PRIORITY | ADDRESS | PORT | KEY 1 | 23.23.23.2 | 2222 | cosine1 RADIUS-ACCOUNTING-SERVER(s): List is empty
Add server with default port: RADIUS-SERVER LIST <priority> <address> <key> [ACCT | AUTH] [<port>]	IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **radius list 2 24.24.24.2 cosine2 auth** VR(1.1.1.1): Object successfully set RADIUS-SERVER LIST 2 AUTH: Object successfully set IPNOS: CONFIG VPN(50) VR(1.1.1.1)# dis rad RADIUS-SERVER CONFIG: RETRANSMIT : 3 TIMEOUT : 5 RADIUS-AUTHENTICATION-SERVER(s): PRIORITY | ADDRESS | PORT | KEY 1 | 23.23.23.2 | 2222 | cosine1 2 | 24.24.24.2 | 1812 | cosine2 RADIUS-ACCOUNTING-SERVER(s): List is empty

Syntax	*Example*
Add accounting server: RADIUS-SERVER LIST <priority> <address> <key> [ACCT | AUTH] [<port>]	IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **radius list 2 45.45.45.3 cosine3 acc** VR(1.1.1.1): Object successfully set RADIUS-SERVER LIST 3 ACCT: Object successfully set IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **dis rad** RADIUS-SERVER CONFIG: RETRANSMIT : 3 TIMEOUT : 5 RADIUS-AUTHENTICATION-SERVER(s): PRIORITY | ADDRESS | PORT | KEY 1 | 23.23.23.2 | 2222 | cosine1 2 | 24.24.24.2 | 1812 | cosine2 RADIUS-ACCOUNTING-SERVER(s): PRIORITY | ADDRESS | PORT | KEY 3 | 45.45.45.3 | 1813 | cosine3
NO RADIUS-SERVER LIST <priority> [ACCT | AUTH]	IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **no radius list 2 acct** VR(1.1.1.1): Object successfully set RADIUS-SERVER LIST 2 AUTH: Object successfully deleted IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **display radius** RADIUS-SERVER CONFIG: RETRANSMIT : 3 TIMEOUT : 5 RADIUS-AUTHENTICATION-SERVER(s): PRIORITY | ADDRESS | PORT | KEY 1 | 23.23.23.2 | 2222 | cosine1 RADIUS-ACCOUNTING-SERVER(s): PRIORITY | ADDRESS | PORT | KEY 3 | 45.45.45.3 | 1813 | cosine3

Syntax	Example
RADIUS-SERVER RETRANSMIT <value>	IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **radius-server retransmit 4** VR(1.1.1.1): Object successfully set RADIUS-SERVER RETRANSMIT 4: Object successfully set IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **dis rad** RADIUS-SERVER CONFIG: RETRANSMIT : 4 TIMEOUT : 5 RADIUS-AUTHENTICATION-SERVER(s): PRIORITY | ADDRESS | PORT | KEY 1 | 23.23.23.1 | 2222 | cosine1 RADIUS-ACCOUNTING-SERVER(s): PRIORITY | ADDRESS | PORT | KEY 3 | 45.45.45.3 | 1813 | cosine3
NO RADIUS-SERVER RETRANSMIT	IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **no radius-server retransmit** VR(1.1.1.1): Object successfully set RADIUS-SERVER RETRANSMIT: Object successfully set IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **dis rad** RADIUS-SERVER CONFIG: RETRANSMIT : 3 TIMEOUT : 5 RADIUS-AUTHENTICATION-SERVER(s): PRIORITY | ADDRESS | PORT | KEY 1 | 23.23.23.1 | 2222 | cosine1 RADIUS-ACCOUNTING-SERVER(s): PRIORITY | ADDRESS | PORT | KEY 3 | 45.45.45.3 | 1813 | cosine3

Syntax	*Example*
RADIUS-SERVER TIMEOUT <value>	IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **radius-server timeout 10** VR(1.1.1.1): Object successfully set RADIUS-SERVER TIMEOUT 10: Object successfully set IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **dis rad** RADIUS-SERVER CONFIG: RETRANSMIT : 3 TIMEOUT : 10 RADIUS-AUTHENTICATION-SERVER(s): PRIORITY | ADDRESS | PORT | KEY 1 | 23.23.23.1 | 2222 | cosine1 RADIUS-ACCOUNTING-SERVER(s): PRIORITY | ADDRESS | PORT | KEY 3 | 45.45.45.3 | 1813 | cosine3
NO RADIUS-SERVER TIMEOUT	IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **no radius-server timeout** VR(1.1.1.1): Object successfully set RADIUS-SERVER TIMEOUT: Object successfully set IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **dis rad** RADIUS-SERVER CONFIG: RETRANSMIT : 3 TIMEOUT : 5 RADIUS-AUTHENTICATION-SERVER(s): PRIORITY | ADDRESS | PORT | KEY 1 | 23.23.23.1 | 2222 | cosine1 RADIUS-ACCOUNTING-SERVER(s): PRIORITY | ADDRESS | PORT | KEY 3 | 45.45.45.3 | 1813 | cosine3

Default Value

Refer to “Radius-Server Parameters” on page 16-122

Command Level

VR

Related Commands

crypto
ip
l2tp
pptp

reassembly

Define the IP packet reassembly timeout. This is the length of time the VR waits for IP packet fragments. The timeout value is from 1 to 255 seconds.

Syntax	*Example*

`REASSEMBLY <time>`	`IPNOS: CONFIG VPN(50) VR(10.10.10.1)# reassembly 45`
	
	`IPNOS: CONFIG VPN(50) VR(10.10.10.1)# commit`
	`Commit configuration`
	`VR(10.10.10.1): Object successfully set`
	
	`IPNOS: CONFIG VPN(50)# vr 10.10.10.1`
	`VPN(50): Object successfully set`
	
	`IPNOS: CONFIG VPN(50) VR(10.10.10.1)# dis reass`
	`REASSEMBLY  : 45`

Default Value

30 seconds

Command Level

VR

Related Commands

None

router

Go to the router command level. See Chapter 20, “Routing Commands” for a list of commands.

Syntax	*Example*
[NO] ROUTER {RIP | ISIS | OSPF | BGP as-number | MPLS | RSVP}	IPNOS : CONFIG VPN(50) VR(1.1.1.1)# **router bgp 1** VR(1.1.1.1): Object successfully set IPNOS : CONFIG VPN(50) VR(1.1.1.1) BGP(1)#
[NO] ROUTER {RIP | ISIS | OSPF | BGP as-number | MPLS | RSVP}	IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **router isis** VR(1.1.1.1): Object successfully set IPNOS: CONFIG VPN(50) VR(1.1.1.1) ISIS#
[NO] ROUTER {RIP | ISIS | OSPF | BGP as-number | MPLS | RSVP}	IPNOS : CONFIG VPN(50) VR(1.1.1.1)# **router ospf** VR(1.1.1.1): Object successfully set IPNOS : CONFIG VPN(50) VR(1.1.1.1) OSPF#
[NO] ROUTER {RIP | ISIS | OSPF | BGP as-number | MPLS | RSVP}	IPNOS : CONFIG VPN(50) VR(1.1.1.1)# **router rip** VR(1.1.1.1): Object successfully set IPNOS : CONFIG VPN(50) VR(1.1.1.1) RIP#
[NO] ROUTER {RIP | ISIS | OSPF | BGP as-number | MPLS | RSVP}	IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **router mpls** VR(1.1.1.1): Object successfully set IPNOS: CONFIG VPN(50) VR(1.1.1.1) MPLS#
[NO] ROUTER {RIP | ISIS | OSPF | BGP as-number | MPLS | RSVP}	IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **router rsvp** VR(1.1.1.1): Object successfully set IPNOS: CONFIG VPN(50) VR(1.1.1.1) RSVP#

Default Value

None

Command Level

VR

Related Commands

None

service-selection

Go to the service selection command level. This optional feature, based on L2TP tunnel-switching, allows you to forward dialup sessions from different interfaces to different IP addresses. This can be useful for managing multiple ISPs and for QoS. See "l2tp" on page 16-78. Table 16-23 lists the supported Layer 2 encapsulation.

Table 16-23 Layer 2 Encapsulation Supported by Service Selection

Encapsulation	*Supported?*
PPP	Yes
Frame Relay	Yes
HDLC	No
Ethernet	No
PPP over ATM	Yes
PPP over Frame RElay	Yes
PPP over Ethernet	Yes

The following commands are available.

- "connection-profile" on page 16-131
- "display" on page 16-135
- "interface" on page 16-136
- "service-selection-template" on page 16-136

Syntax	*Example*

`SERVICE-SELECTION`	`IPNOS: CONFIG VPN(50) VR(10.10.10.1)#` **`service-selection`**
	`VR(1.1.1.1): Object successfully set`
	
	`IPNOS: CONFIG VPN(50) VR(10.10.10.1) SERVICE-SELECTION#`

Default Value

None

Command Level

VR

Related Commands

None

connection-profile

Go to the connection-profile command level. A connection profile chooses an "access VR" for authenticating a dialup session, the LNS address for termination, and a local IP pool for giving the session an address. The following commands are available.

- "display" on page 16-132
- "ip-local-pool-name" on page 16-132
- "lns-address" on page 16-133
- "vpn" on page 16-134
- "vr" on page 16-134

Syntax	*Example*
[NO] CONNECTION-PROFILE profile-name	IPNOS: CONFIG VPN(50) VR(10.10.10.1) SERVICE-SELECTION# **connection cosine1** IPNOS: CONFIG VPN(50) VR(10.10.10.1) SERVICE-SELECTION CONNECTION-PROFILE(cosine1)# **display** VPN : 50 VR : 10.10.10.1 NAME : cosine1 SERVICE-VPN : * SERVICE-VR : * LNS-ADDRESS : * IP POOL NAME : *

Default Value

None

Command Level

Service-selection

Related Commands

None

display

View the connection profile configuration.

Syntax	*Example*

DISPLAY	`IPNOS: CONFIG VPN(50) VR(10.10.10.1) SERVICE-SELECTION CONNECTION-PROFILE(cosine1)# display`
	`VPN : 50`
	`VR : 10.10.10.1`
	`NAME : cosine1`
	
	`SERVICE-VPN : 50`
	`SERVICE-VR : 10.10.10.1`
	`LNS-ADDRESS : 192.168.24.1`
	`IP POOL NAME : cosine`

Default Value

None

Command Level

Connection-profile

Related Commands

None

ip-local-pool-name

Assign an IP address pool to the connection profile. See “ip” on page 16-68 for details. If left empty, the L2TP IP pool configuration is used. See “l2tp” on page 16-82. The following lists the order of precedence for finding the local IP address pool.

1. Connection profile. See “connection-profile” on page 16-131.
2. L2TP configuration. See “l2tp” on page 16-82.

Syntax	Example
IP-LOCAL-POOL-NAME pool-name	IPNOS: CONFIG VPN(50) VR(10.10.10.1) SERVICE-SELECTION CONNECTION-PROFILE(cosine1)# **ip-local-pool cosine** IPNOS: CONFIG VPN(50) VR(10.10.10.1) SERVICE-SELECTION CONNECTION-PROFILE(cosine1)# **display** VPN : 50 VR : 10.10.10.1 NAME : cosine1 SERVICE-VPN : * SERVICE-VR : * LNS-ADDRESS : * IP POOL NAME : cosine

Default Value

None

Command Level

Connection-profile

Related Commands

ip

lns-address

Assign the IP address of the L2TP LNS. This will be the termination of the dialup connection. The following lists the order of precedence for finding the LNS IP address.

1. RADIUS server. See “radius-server” on page 16-122.
2. Connection profile. See “connection-profile” on page 16-131.
3. Tunnel switching table. See “tunnel-switch” on page 16-99.

Syntax	Example
LNS-ADDRESS ip-address	IPNOS: CONFIG VPN(50) VR(10.10.10.1) SERVICE-SELECTION CONNECTION-PROFILE(cosine1)# **lns-address 192.168.24.1**

Default Value

None

Command Level

Connection-profile

Related Commands

l2tp lns

vpn

Assign the VPN ID of the access VR for authenticating a dialup session. This command is optional. If empty, then the current VPN is used.

Syntax	*Example*
`VPN vpn-id`	`IPNOS: CONFIG VPN(50) VR(10.10.10.1) SERVICE-SELECTION CONNECTION-PROFILE(cosine1)# vpn 50`

Default Value

None

Command Level

Connection-profile

Related Commands

None

vr

Assign the VR ID of the access VR for authenticating a dialup session. This command is optional. If empty, then this VR authenticates the dialup connection.

Syntax	*Example*
`VR vr-id`	`IPNOS: CONFIG VPN(50) VR(10.10.10.1) SERVICE-SELECTION CONNECTION-PROFILE(cosine1)# vr 10.10.10.1`

Default Value

None

Command Level

Connection-profile

Related Commands

None

display

View the current service selection configuration.

Syntax	*Example*
DISPLAY	IPNOS: CONFIG VPN(50) VR(1.1.1.1) SERVICE-SELECTION# **display** VPN : 50 VR : 1.1.1.1 MANAGED INTERFACE(s): List is empty CONNECTION-PROFILE(s): NAME | SERVICE-VPN | SERVICE-VR | LNS-ADDRESS | IP-LOCAL-POOL-NAME cosine1 | 50 | 10.10.10.1 | 192.168.24.1| cosine SERVICE-SELECTION-TEMPLATE(s): List is empty IPNOS: CONFIG VPN(50) VR(1.1.1.1) SERVICE-SELECTION#

Default Value

None

Command Level

Service-selection

Related Commands

None

interface

Assign an interface to manage. When a user's dialup request goes through this interface, the interface checks the VR listed in the connection profile (not necessarily the current VR) for authentication. The VR will either check with a RADIUS server, the connection profile, or both for the LNS IP address. If both have an LNS address, the RADIUS server takes precedence. If there is a PPPoE session, the interface checks the service-selection-template for a list of connection profiles. See "service-selection-template" on page 16-136.

Syntax	*Example*
`INTERFACE {slot/port[:channel[:subif]]}`	`IPNOS: CONFIG VPN(50) VR(10.10.10.1) SERVICE-SELECTION#` **`interface 2/1`** `IPNOS: CONFIG VPN(50) VR(10.10.10.1) SERVICE-SELECTION#`

Default Value

None

Command Level

Service-selection

Related Commands

None

service-selection-template

Go to the service-selection-template command level. Currently only available for PPPoE, this template contains a list of connection profiles that a dialup user can choose from. See "pppoe" on page 10-45. The following commands are available.

- "connection-profile" on page 16-137
- "display" on page 16-138

Syntax	Example
[NO] SERVICE-SELECTION-TEMPLATE name	IPNOS: CONFIG VPN(50) VR(10.10.10.1) SERVICE-SELECTION# **service-selection-template globalnet** IPNOS: CONFIG VPN(50) VR(10.10.10.1) SERVICE-SELECTION SERVICE-SELECTION-TEMPLATE(globalnet)#

Default Value

None

Command Level

Service-selection

Related Commands

None

connection-profile

Assign or remove a connection profile to the template.

Syntax	Example
[NO] SERVICE-SELECTION-TEMPLATE name	IPNOS: CONFIG VPN(50) VR(10.10.10.1) SERVICE-SELECTION SERVICE-SELECTION-TEMPLATE(globalnet)# **connection-profile cosine1** IPNOS: CONFIG VPN(50) VR(10.10.10.1) SERVICE-SELECTION SERVICE-SELECTION-TEMPLATE(globalnet)# **display** VPN : 50 VR : 10.10.10.1 NAME : globalnet CONNECTION-PROFILE(s): cosine1

Default Value

None

Command Level

Service-selection-template

Related Commands

connection-profile

display

View the current service-selection-template configuration.

Syntax	*Example*

`[NO] SERVICE-SELECTION-TEMPLATE name`	`IPNOS: CONFIG VPN(50) VR(10.10.10.1) SERVICE-SELECTION SERVICE-SELECTION-TEMPLATE(globalnet)# display`
	`VPN : 50`
	`VR : 10.10.10.1`
	`NAME : globalnet`
	
	`CONNECTION-PROFILE(s):`
	`cosine1`
	`cosine2`
	`cosine3`

Default Value

None

Command Level

Service-selection-template

Related Commands

None

show

View the current configuration of the VR or specific routing protocols.

Syntax	*Example*
SHOW CONFIGURATION	IPNOS: CONFIG VPN(10) VR(15.0.0.1)# show config CREATE VPN 10 CREATE VR 15.0.0.1 2 RADIUS-SERVER RETRANSMIT 5 COMMIT VR 15.0.0.1 VI 1 IP ADDRESS 15.0.0.1 255.255.255.255 IP BROADCAST 0 COMMIT # VI 1 CREATE VI 2 IP ADDRESS 192.16.2.1 255.255.255.252 COMMIT # VI 2 CREATE VI 3 IP ADDRESS 10.2.1.1 255.255.255.252 COMMIT # VI 3 CREATE VI 4 IP ADDRESS 10.2.1.5 255.255.255.252 COMMIT # VI 4 VI 2 CONNECTED-TO DISCONNECTED COMMIT # VI 2 VI 3 CONNECTED-TO VIRTUAL 2/192.168.36.1/2 COMMIT # VI 3 VI 4 CONNECTED-TO VIRTUAL 2/192.168.34.1/2 COMMIT # VI 4 TELNET COMMIT QOS COMMIT L2TP COMMIT # L2TP L2TP COMMIT # LNS L2TP COMMIT # LAC L2TP

Syntax	*Example*
SHOW CONFIGURATION (continued)	COMMIT # TUNNEL-SWITCH COMMIT # VR 15.0.0.1 CREATE VPN 10 CREATE VR 15.0.0.1 2 VI 1 COMMIT # VI 1 VI 2 COMMIT # VI 2 VI 3 COMMIT # VI 3 VI 4 COMMIT # VI 4 IP ROUTE DEFAULT 10.2.1.1 IP ROUTE DEFAULT 10.2.1.5 COMMIT # VR 15.0.0.1 ROUTING VI 2 COMMIT # VI 2 VI 3 COMMIT # VI 3 VI 4 COMMIT # VI 4
SHOW BGP {CONFIGURATION | NEIGHBORS | SUMMARY | COMMUNITY | DAMPENED-PATHS}	IPNOS: CONFIG VPN(50) VR(4.4.4.4)# **show bgp config** ROUTER BGP 6942 BGP DAMPENING 600 2 3 20 NEIGHBOR cosine PEER-GROUP NEIGHBOR cosine REMOTE-AS 333 NEIGHBOR 172.16.24.1 REMOTE-AS 333 NEIGHBOR 172.16.24.1 PEER-GROUP cosine NEIGHBOR 172.16.24.1 EXPORT 100 COMMIT # BGP
SHOW BGP {CONFIGURATION | NEIGHBORS | SUMMARY | COMMUNITY | DAMPENED-PATHS}	IPNOS: CONFIG VPN(50) VR(4.4.4.4)# **show bgp neighbors** peer-group cosine flags <> peer 172.16.24.1 version 4 lcladdr 172.16.24.2 gateway (null) remote router ID 3.3.3.3 flags 0x0 state 0x3 <Active> options 0x440000 <NoV4ASLoop MED> metric_out -1 preference 170 preference2 0 recv buffer size 0 send buffer size 0 messages in 0 (updates 0, not updates 0) 0 octets messages out 34 (updates 0, not updates 34) 816 octets

Syntax	*Example*
SHOW BGP {CONFIGURATION | NEIGHBORS | SUMMARY | COMMUNITY | DAMPENED-PATHS}	IPNOS: CONFIG VPN(50) VR(4.4.4.4)# **show bgp summary** Neighbor V AS MsgRcvd MsgSent State PfxRcvd PfxSent 172.16.24.1 4 333 0 34 Active 0 0 BGP summary, 1 groups, 1 peers
SHOW BGP {CONFIGURATION | NEIGHBORS | SUMMARY | COMMUNITY | DAMPENED-PATHS}	IPNOS: CONFIG VPN(50) VR(4.4.4.4)# **show bgp dampened-paths** IPNOS: CONFIG VPN(50) VR(4.4.4.4)#
SHOW ISIS {ADJ-LOG | ADJACENCY | ATTRIBUTES | CIRCUIT | CONFIGURATION | DATABASE | SPF | STATISTICS}	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis adj-log** IS-IS Adjacency Log
SHOW ISIS {ADJ-LOG | ADJACENCY | ATTRIBUTES | CIRCUIT | CONFIGURATION | DATABASE | SPF | STATISTICS}	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis adjacency** System-ID Interface SNPA State Hold-Time Type
SHOW ISIS {ADJ-LOG | ADJACENCY | ATTRIBUTES | CIRCUIT | CONFIGURATION | DATABASE | SPF | STATISTICS}	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis circuit** CIRCUIT-NAME LOCAL-ID I/F-INDEX TYPE LEVEL STATE

Syntax	*Example*
SHOW ISIS {ADJ-LOG | ADJACENCY | ATTRIBUTES | CIRCUIT | CONFIGURATION | DATABASE | SPF | STATISTICS}	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config** ROUTER ISIS SYSTEM-ID 0200.0505.0505 EVENT-LOG AREA ADJACENCY EVENT-LOG AREA LSP EVENT-LOG AREA SNP EVENT-LOG AREA SPF EVENT-LOG AREA ALL AREA-ADDRESS 0101.0101 COMMIT # ISIS
SHOW ISIS {ADJ-LOG | ADJACENCY | ATTRIBUTES | CIRCUIT | CONFIGURATION | DATABASE | SPF | STATISTICS}	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis database** Level 1 LSP database LSPID Seq Num Checksum Holdtime ATT/P/OL 0200.0505.0505.00-00* 0x2 0xc5c1 714 0/0/0 Level 2 LSP database LSPID Seq Num Checksum Holdtime ATT/P/OL 0200.0505.0505.00-00* 0x3 0xc3c2 714 0/0/0
SHOW ISIS {ADJ-LOG | ADJACENCY | ATTRIBUTES | CIRCUIT | CONFIGURATION | DATABASE | SPF | STATISTICS}	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis spf** SPF [LOG | RESULTS]
SHOW RIP {CONFIGURATION | NEIGHBORS | STATISTICS}	IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **show rip config** CREATE VPN 50 CREATE VR 1.1.1.1 ROUTER RIP COMMIT # RIP VI 1 COMMIT # VI 1 VI 2 ROUTER RIP COMMIT # VI 2 COMMIT # VR 1.1.1.1 ROUTING
SHOW RIP {CONFIGURATION | NEIGHBORS | STATISTICS}	IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **show rip neighbor** Targets: 192.168.24.1 -> 224.0.0.9 Interface: 192.168.24.1 VI2

Syntax	*Example*
SHOW RIP {CONFIGURATION | NEIGHBORS | STATISTICS}	IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **show rip stat** Route Changes: 0 VI: 2 Bad Packets: 0 Bad Routes: 0 Triggered Updates: 0
SHOW ISIS {ADJ-LOG | ADJACENCY | ATTRIBUTES | CIRCUIT | CONFIGURATION | DATABASE | SPF | STATISTICS}	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis statistics** PDU type Received Drops Sent Rexmit LSP 0 0 0 0 IIH 0 0 0 0 PSNP 0 0 0 0 CSNP 0 0 0 0 Unknown 0 0 0 0 Total 0 0 0 0 Corrupted LSPs : 0 L1 Database Overload : 0 L2 Database Overload : 0 Area Address Dropped : 0 Area Mismatches : 0 Max Area Mismatches : 0 Own LSP Purges : 0 ID Length Mismatches : 0 Authentication Failures : 0 Sequence Number Skipped : 0 Max Sequence Number Exceeded : 0

Default Value

None

Command Level

VR

Related Commands

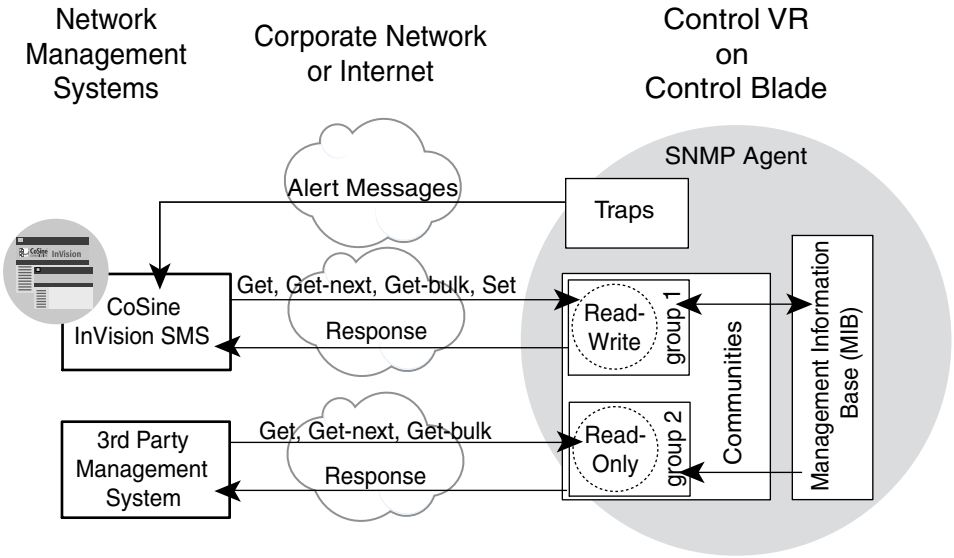
None

snmp-server

Configure the Simple Network Management Protocol (SNMP) agent. Use to connect to network management systems (NMSs) such as CoSine's InVision.

Refer to RFCs 1902 and 1905 for more information. Enable **snmp-server** in the control VR 165.165.100.100 in the system VPN 1000000 on the control blade. Once enabled, all traps and MIB events will be sent to and received from an NMS. Figure 16-7 describes the IPNOS SNMP process.

Figure 16-7 SNMP Process



IPNOS Commands:

snmp-server enable— Enables/disables SNMP agent

snmp-server enable traps— Enables/disables traps

snmp-server community—
1. Enables/disables SNMP agent
2. Adds/deletes communities
3. Creates a community of network hosts allowed to receive management information

snmp-server host—
1. Enables/disables traps
2. Adds/deletes network hosts to receive traps (alerts)
3. Sets the type of traps (SNMPv1, SNMPv2c, or both)

040-00280

Table 16-24 describes the parameters available.

Table 16-24 SNMP Server Parameters

Parameter	*Description*
Community	Create a list of hosts with Network Management Systems allowed to send get or set requests for network management information. When you create a community, you also automatically enable the SNMP agent. Use the **no** option to delete a community. The following parameters are available: • String—Community string a network host uses to request for management information. Use alphanumeric characters. • RO|RW—Read-only or read-write access to the management information base (MIB), which defines specific network management variables and their meanings. • IP address—IP address of the specific network host allowed to receive management information. If the IP address is 0.0.0.0, then any NMS with the correct community string can receive management information.
Enable [traps]	Enables or disables the SNMP agent. The default value is disable. You can also enable or disable trap messages by IPNOS software. Traps are network events, such as a router shutting down or starting up. Use the **no** option to disable the agent. The default value is disable.
Host	List of network hosts to receive trap messages. Use the **no** option to delete a host. The following parameters need to be configured: • IP address—The network host to receive traps. • Community String—Sets the community string included in trap messages. Use alphanumeric characters. • Port—The UDP port from which the trap messages are sent. The default value is port 162. • Type—The type of SNMP messages the host will receive: 1. SNMPv1 2. SNMPv2c 3. Both SNMPv1 and SNMPv2c The default value is 2.

Syntax	*Example*
[NO] SNMP-SERVER ENABLE [TRAPS]	IPNOS : CONFIG VPN(100000) VR(165.165.100.100)# **snmp-server enable** VR(165.165.100.100): Object successfully set SNMP Research SNMP Agent Resident Module Version 14.2.1.7 Copyright 1989, 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997 SNMP Research, Inc. IPNOS : CONFIG VPN(100000) VR(165.165.100.100)# **display snmp** SNMP-SERVER : ENABLE TRAPS : ENABLE COMMUNITY(s): COMMUNITY | PRIVILEGE | IP public | RO | 0.0.0.0 private | RW | 0.0.0.0 TRAPS HOST(s): List is empty IPNOS : CONFIG VPN(100000) VR(165.165.100.100)# **commit** Commit configuration VR(165.165.100.100): Object successfully set IPNOS : CONFIG VPN(100000)#
[NO] SNMP-SERVER ENABLE [TRAPS]	IPNOS : CONFIG VPN(100000) VR(165.165.100.100)# **snmp-server enable traps** VR(165.165.100.100): Object successfully set IPNOS : CONFIG VPN(100000) VR(165.165.100.100)#

Syntax	Example
[NO] SNMP-SERVER COMMUNITY <string> [RO|RW] <address>	IPNOS : CONFIG VPN(100000) VR(165.165.100.100)# **snmp-server community sam123 rw 128.192.34.52** VR(165.165.100.100): Object successfully set COMMUNITY sam: Object successfully set IPNOS : CONFIG VPN(100000) VR(165.165.100.100)# **display snmp** SNMP-SERVER : ENABLE TRAPS : ENABLE COMMUNITY(s): COMMUNITY | PRIVILEGE | IP sam123 | RW | 128.192.34.52 public | RO | 0.0.0.0 private | RW | 0.0.0.0 TRAPS HOST(s): COMMUNITY | IP | PORT | TYPE sam | 128.192.34.3 | 162 | 2 IPNOS : CONFIG VPN(100000) VR(165.165.100.100)#
[NO] SNMP-SERVER HOST <address> <community-string> [PORT <trap-udp-port>] [TYPE {1 | 2 | 3}]	IPNOS : CONFIG VPN(100000) VR(165.165.100.100)# **snmp-server host 128.192.34.3 sam** VR(165.165.100.100): Object successfully set IPNOS : CONFIG VPN(100000) VR(165.165.100.100)# **display snmp** SNMP-SERVER : ENABLE TRAPS : ENABLE COMMUNITY(s): COMMUNITY | PRIVILEGE | IP sam123 | RW | 128.192.34.52 public | RO | 0.0.0.0 private | RW | 0.0.0.0 TRAPS HOST(s): COMMUNITY | IP | PORT | TYPE sam | 128.192.34.3 | 162 | 2

Syntax	*Example*
[NO] SNMP-SERVER ENABLE [TRAPS]	IPNOS : CONFIG VPN(100000) VR(165.165.100.100)# **no snmp enable traps** VR(165.165.100.100): Object successfully set SNMP-SERVER TRAPS: Disabled successfully IPNOS : CONFIG VPN(100000) VR(165.165.100.100)# **display snmp** SNMP-SERVER : ENABLE TRAPS : DISABLE COMMUNITY(s): COMMUNITY | PRIVILEGE | IP sam123 | RW | 128.192.34.52 public | RO | 0.0.0.0 private | RW | 0.0.0.0 TRAPS HOST(s): COMMUNITY | IP | PORT | TYPE sam | 128.192.34.3 | 162 | 2 private | 192.168.34.56 | 162 | 2 IPNOS : CONFIG VPN(100000) VR(165.165.100.100)#
[NO] SNMP-SERVER ENABLE [TRAPS]	IPNOS : CONFIG VPN(100000) VR(165.165.100.100)# **no snmp enable** VR(165.165.100.100): Object successfully set SNMP-SERVER: Disabled successfully IPNOS : CONFIG VPN(100000) VR(165.165.100.100)# **display snmp** SNMP-SERVER : DISABLE TRAPS : DISABLE COMMUNITY(s): List is empty TRAPS HOST(s): List is empty IPNOS : CONFIG VPN(100000) VR(165.165.100.100)#

Syntax	Example
[NO] SNMP-SERVER HOST <address> <community-string>	IPNOS : CONFIG VPN(100000) VR(165.165.100.100)# **display snmp** SNMP-SERVER : ENABLE TRAPS : DISABLE COMMUNITY(s): COMMUNITY | PRIVILEGE | IP sam123 | RW | 128.192.34.52 public | RO | 0.0.0.0 string | RO | 128.192.34.50 private | RW | 0.0.0.0 TRAPS HOST(s): COMMUNITY | IP | PORT | TYPE ro | 128.192.34.2 | 162 | 2 sam | 128.192.34.3 | 162 | 2 private | 192.168.34.56 | 162 | 2 IPNOS : CONFIG VPN(100000) VR(165.165.100.100)# **no snmp-server host 128.192.34.2 ro** VR(165.165.100.100): Object successfully set SNMP-SERVER HOST 128.192.34.2 ro: Object successfully deleted IPNOS : CONFIG VPN(100000) VR(165.165.100.100)# **display snmp** SNMP-SERVER : ENABLE TRAPS : DISABLE COMMUNITY(s): COMMUNITY | PRIVILEGE | IP sam123 | RW | 128.192.34.52 public | RO | 0.0.0.0 string | RO | 128.192.34.50 private | RW | 0.0.0.0 TRAPS HOST(s): COMMUNITY | IP | PORT | TYPE sam | 128.192.34.3 | 162 | 2 private | 192.168.34.56 | 162 | 2 IPNOS : CONFIG VPN(100000) VR(165.165.100.100)#

Syntax	*Example*
NO SNMP-SERVER COMMUNITY <string>	IPNOS : CONFIG VPN(100000) VR(165.165.100.100)# **display snmp** SNMP-SERVER : ENABLE TRAPS : ENABLE COMMUNITY(s): COMMUNITY | PRIVILEGE | IP sam123 | RW | 128.192.34.52 public | RO | 0.0.0.0 string | RO | 128.192.34.50 private | RW | 0.0.0.0 woodywoodward | RW | 192.168.34.53 TRAPS HOST(s): COMMUNITY | IP | PORT | TYPE sam | 128.192.34.3 | 162 | 2 private | 192.168.34.56 | 162 | 2 IPNOS : CONFIG VPN(100000) VR(165.165.100.100)# **no snmp-server community sam** VR(165.165.100.100): Object successfully deleted IPNOS : CONFIG VPN(100000) VR(165.165.100.100)#

Default Value

None

Command Level

VR

Related Commands

None

syslogd

Map event log messages to an external system log daemon (syslogd) server. IPNOS collects event messages from the different blades and temporarily stores them on the control blade. When you enable **syslogd**, the event messages are forwarded to an external UNIX or PC syslogd server.

Facilities and levels correspond to IPNOS's areas and levels. IPNOS areas represent a group of related features. For example, link layer areas represent ATM, Frame Relay, HDLC and PPP messages. IPNOS levels represent priority levels of a message. For example, emergency level represents major system problems such as an inoperative blade.

Enable **syslogd** in the control VR 165.165.100.100 in the system VPN 1000000 on the control blade. Once enabled, all messages generated from IPNOS will be forwarded to the specified syslogd server. Use the **event-log** command at each command level to enable or disable a sub-set of the messages for monitoring a specific feature.

NOTE: Use the event log commands sparingly. The more event log messages you enable can slow down the performance of IPNOS.

Table 16-25 describes the parameters available.

Table 16-25 Syslogd Parameters

Parameter	*Description*
Enable	Enable system logging. Use the **no** option to disable the command.
Facility	Map an IPNOS area event to a UNIX syslogd facility. UNIX provides 24 facilities while IPNOS provides 11. Facilities 0 through 15 are UNIX function-specific (mail, daemon, UUCP, CRON, etc.) and are not relevant to IPNOS. Facilities 16 through 23 represent user-definable UNIX facilities (local0 through local7). By default, all IPNOS area events are mapped to 19 (local3). The following lists the facility number and the IPNOS event area it represents. 1—Trap messages 2—Emergency messages such as a loss of a blade 3—General system messages 4—Authentication messages 5—Application messages (such as PPTP, RADIUS) 6—Protocol messages (except IPSec and layer 2) 7—IPSec, IKE and ISAKMP messages 8—Link layer messages (such as ATM, Frame Relay, HDLC, PPP) 9—Configuration messages 10—Packet filter messages 11—Firewall messages (such as FTP, HTTP proxies)
Level	Map an IPNOS severity level to a UNIX priority level. The following lists the IPNOS severity levels. 1—Critical 2—Major 3—Minor 4—Inform 5—Debug By default, all IPNOS severity levels are mapped to UNIX priority level 5 (warning). Use the following list of UNIX priority levels to customize for your monitoring needs. 1—Emergency: Panic condition 2—Alert: Problem that must be corrected immediately 3—Critical Condition: problems such as hard drive errors 4—Errors: General application errors 5—Warning messages 6—Log Notice: Notifications of states in the system that may require handling 7—Informational messages
Host	Set the IP address of the host for system logging.

Syntax	*Example*
[NO] SYSLOGD ENABLE	IPNOS: CONFIG VPN(1000000) VR(165.165.100.100)# **syslogd enable** VR(165.165.100.100): Object successfully set IPNOS: CONFIG VPN(1000000) VR(165.165.100.100)# **display syslogd** SYSLOGD : ENABLE SYSLOGD HOST : 0.0.0.0 FACILITY 1 : 19 FACILITY 2 : 19 FACILITY 3 : 19 FACILITY 4 : 19 FACILITY 5 : 19 FACILITY 6 : 19 FACILITY 7 : 19 FACILITY 8 : 19 FACILITY 9 : 19 FACILITY 10 : 19 FACILITY 11 : 19 LEVEL 1 : 5 LEVEL 2 : 5 LEVEL 3 : 5 LEVEL 4 : 5 LEVEL 5 : 5

Syntax	*Example*
SYSLOGD HOST <ip-address>	IPNOS: CONFIG VPN(1000000) VR(165.165.100.100)# **syslogd host 172.20.12.236** VR(165.165.100.100): Object successfully set IPNOS: CONFIG VPN(1000000) VR(165.165.100.100)# **display syslogd** SYSLOGD : ENABLE SYSLOGD HOST : 172.20.12.236 FACILITY 1 : 19 FACILITY 2 : 19 FACILITY 3 : 19 FACILITY 4 : 19 FACILITY 5 : 19 FACILITY 6 : 19 FACILITY 7 : 19 FACILITY 8 : 19 FACILITY 9 : 19 FACILITY 10 : 19 FACILITY 11 : 19 LEVEL 1 : 5 LEVEL 2 : 5 LEVEL 3 : 5 LEVEL 4 : 5 LEVEL 5 : 5

Syntax	*Example*

Map IPNOS level 1 (critical) to UNIX priority level 3 (critical condition):
`SYSLOGD LEVEL [1-5] [1-7]`
	`IPNOS: CONFIG VPN(1000000) VR(165.165.100.100)#` **`syslogd level 1 3`**
	`VR(165.165.100.100): Object successfully set`
	`SYSLOGD: Object successfully set`
	
	`IPNOS: CONFIG VPN(1000000) VR(165.165.100.100)#` **`display syslogd`**
	`SYSLOGD : ENABLE`
	`SYSLOGD HOST : 172.20.12.236`
	`FACILITY 1 : 19`
	`FACILITY 2 : 19`
	`FACILITY 3 : 19`
	`FACILITY 4 : 19`
	`FACILITY 5 : 19`
	`FACILITY 6 : 19`
	`FACILITY 7 : 19`
	`FACILITY 8 : 19`
	`FACILITY 9 : 19`
	`FACILITY 10 : 19`
	`FACILITY 11 : 19`
	`LEVEL 1 : 3`
	`LEVEL 2 : 5`
	`LEVEL 3 : 5`
	`LEVEL 4 : 5`
	`LEVEL 5 : 5`

Syntax	*Example*

Map IPNOS area event 1 (traps) to syslogd facility 18 (local2):
`SYSLOGD FACILITY [1-11] [1-24]`
	`IPNOS: CONFIG VPN(1000000) VR(165.165.100.100)#` **`syslogd facility 1 18`**
	`VR(165.165.100.100): Object successfully set`
	`SYSLOGD: Object successfully set`
	
	`IPNOS: CONFIG VPN(1000000) VR(165.165.100.100)#` **`display syslogd`**
	`SYSLOGD : ENABLE`
	`SYSLOGD HOST : 172.20.12.236`
	`FACILITY 1 : 18`
	`FACILITY 2 : 19`
	`FACILITY 3 : 19`
	`FACILITY 4 : 19`
	`FACILITY 5 : 19`
	`FACILITY 6 : 19`
	`FACILITY 7 : 19`
	`FACILITY 8 : 19`
	`FACILITY 9 : 19`
	`FACILITY 10 : 19`
	`FACILITY 11 : 19`
	`LEVEL 1 : 3`
	`LEVEL 2 : 5`
	`LEVEL 3 : 5`
	`LEVEL 4 : 5`
	`LEVEL 5 : 5`

Default Value

Disabled

Command Level

VR

Related Commands

None

telnet

Go to the telnet command level. Enable **telnet** in the control VR 165.165.100.100 in the system VPN 1000000 on the control blade. Once enabled, you can log into the IPSX switch via a telnet session. Up to 8 sessions to the same IPSX switch are possible. The following commands are available.

- display
- enable
- kill
- show
- timeout

Syntax	*Example*
`TELNET`	`IPNOS : CONFIG VPN(1000000) VR(165.165.100.100)# telnet` `IPNOS : CONFIG VPN(1000000) VR(165.165.100.100) TELNET#`

Default Value

None

Command Level

VR

Related Commands

None

display

View the current telnet configuration and the the currently running telnet sessions.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG VPN(1000000) VR(165.165.100.100) TELNET# **display** STATUS : ENABLE TIMEOUT : 30 ACTIVE TELNET SESSION(s) : SESSION ID| REMOTE IP ADDRESS | REMOTE PORT 1 | 172.20.12.160 | 1342 2 | 172.20.12.28 | 45551

Default Value

None

Command Level

Telnet

Related Commands

None

enable

Enable or disable telnet access to the VR.

Syntax	*Example*
[NO] ENABLE	IPNOS : CONFIG VPN(1000000) VR(165.165.100.100) TELNET# **telnet enable** VR(165.165.100.100): Object successfully set TELNET: Object successfully set

Default Value

Disabled

Command Level

Telnet

Related Commands

None

kill

Stop a telnet session on the VR.

Syntax	*Example*
KILL <session_id>	IPNOS : CONFIG VPN(1000000) VR(165.165.100.100) TELNET# **display** STATUS : ENABLE TIMEOUT : 30 ACTIVE TELNET SESSION(s) : SESSION ID| REMOTE IP ADDRESS | REMOTE PORT 1 | 172.20.12.160 | 1342 2 | 172.20.12.28 | 45551 IPNOS : CONFIG VPN(1000000) VR(165.165.100.100) TELNET# **kill 2** VR(2.2.2.2): Object successfully set TELNET: Object successfully deleted IPNOS : CONFIG VPN(1000000) VR(165.165.100.100) TELNET# **display** STATUS : ENABLE TIMEOUT : 30 ACTIVE TELNET SESSION(s) : SESSION ID| REMOTE IP ADDRESS | REMOTE PORT 1 | 172.20.12.160 | 1342

Default Value

None

Command Level

Telnet

Related Commands

None

show

View the current telnet configuration, plus the the currently running telnet sessions.

Syntax	*Example*

SHOW	`IPNOS : CONFIG VPN(1000000) VR(165.165.100.100) TELNET#` **`show`**
	
	`STATUS : ENABLE`
	`TIMEOUT : 30`
	
	`ACTIVE TELNET SESSION(s) :`
	
	`SESSION ID| REMOTE IP ADDRESS | REMOTE PORT`
	`1 | 172.20.12.160 | 1342`
	`2 | 172.20.12.28 | 45551`

Default Value

None

Command Level

Telnet

Related Commands

None

timeout

Set the time out for inactive telnet sessions, in minutes.

Syntax	Example
TIMEOUT <minutes>	IPNOS : CONFIG VPN(1000000) VR(165.165.100.100) TELNET# **timeout 45** VR(2.2.2.2): Object successfully set TELNET: Object successfully set IPNOS : CONFIG VPN(1000000) VR(165.165.100.100) TELNET# **display** STATUS : ENABLE TIMEOUT : 45 ACTIVE TELNET SESSION(s) : SESSION ID| REMOTE IP ADDRESS | REMOTE PORT 1 | 172.20.12.160 | 1342 2 | 172.20.12.28 | 45551
NO TIMEOUT	IPNOS : CONFIG VPN(1000000) VR(165.165.100.100) TELNET# **no timeout** VR(2.2.2.2): Object successfully set TELNET: Object successfully set IPNOS : CONFIG VPN(1000000) VR(165.165.100.100) TELNET# **display** STATUS : ENABLE TIMEOUT : 30 ACTIVE TELNET SESSION(s) : SESSION ID| REMOTE IP ADDRESS | REMOTE PORT 1 | 172.20.12.160 | 1342 2 | 172.20.12.28 | 45551

Default Value

30 minutes

Command Level

Telnet

Related Commands

None

traceroute

Trace the path taken by a packet to its destination. Table 16-26 describes available parameters.

Table 16-26 Traceroute Parameters

Parameter	*Description*
address	IP address of the destination server
HOPCOUNT	Maximum number of hops allowed
SOURCE	IP address of the source server

Syntax	*Example*
TRACEROUTE <address> [HOPCOUNT <hop-count>] [SOURCE <source-address>]	IPNOS : CONFIG VPN(50) VR(10.1.1.1)# **traceroute 192.168.4.1** trace route to 192.168.4.1 over a maximum of 30 hops: 1 0 ms 0 ms 0 ms [172.16.16.1] 2 0 ms 0 ms 0 ms [172.16.8.1] 3 0 ms 0 ms 0 ms [10.1.0.1] 4 0 ms 0 ms 0 ms [192.168.4.1] Packet trace complete.

Default Value

30 hops

Command Level

VR

Related Commands

ping

ttl

Set a default time-to-live (TTL) value for the virtual router. The TTL counter specifies the maximum number of hops that an IP data packet can traverse in the network before an intermediate router discards the packet, thus preventing packets from looping endlessly through the network. Use a value between 1 and 255.

Syntax	*Example*
`TTL <value>`	`IPNOS : CONFIG VPN(50) VR(127.0.0.1)# ttl 110`

Default Value

30 hops

Command Level

VR

Related Commands

None

type

Assign the VR type. Table 16-27 describes different types of VRs available.

Table 16-27 VR Types

VR Type	*Description*
Customer	Owned and run by the customer
Control	Control VR for network testing and maintenance
Service provider	Maintained by the Internet service provider (ISP)

Syntax	*Example*
`TYPE {CUSTOMER | CONTROL | SERVICE-PROVIDER}`	`IPNOS : CONFIG VPN(50) VR(1.1.1.1)#` **`type service-provider`**

Default Value

None

Command Level

VR

Related Commands

None

username

Assign a username and password for the following protocols:

- Level 2 Tunneling Protocol (L2TP)
- Point-to-Point Protocol (PPP)
- Point-to-Point Tunneling Protocol (PPTP)

PPP, PPTP, and L2TP use the username list for a valid list of users. If a RADIUS server is installed, then it is used instead of the username list.

L2TP requires the LAC server's Internet hostname in the username list. The password is a shared secret between the LAC and the LNS. The shared secret is used if the LAC or LNS requests tunnel authentication. Even if you use a RADIUS server, you must put the LAC host name and shared secret in the username list.

Refer to "pptp" on page 16-113 for more information on CHAP and PAP. Refer to "l2tp" on page 16-78 for more information on L2TP. Refer to "hostname" on page 16-60 for more information on setting a host name to a remote VR. Use the **no** option to delete a username.

NOTE: This command has no relationship to the command "name" on page 16-106.

Syntax	*Example*

Conventional user name and password:
`[NO] USERNAME <name> PASSWORD <password>`
	`IPNOS: CONFIG VPN(50) VR(2.2.2.2)# username fred password abcdefg`
	`VR(2.2.2.2): Object successfully set`
	`USERNAME fred: Object successfully set`
	
	`IPNOS: CONFIG VPN(50) VR(2.2.2.2)# display ppp username`
	
	`PPP USERNAME(s):`
	`USERNAME | PASSWORD`
	`fred | ********`

Syntax	*Example*

L2TP LAC Internet host name and shared secret:
`[NO] USERNAME <name> PASSWORD <password>`
	`IPNOS: CONFIG VPN(50) VR(2.2.2.2)# **username lac.bigcompany.com password 12345**`
	`VR(2.2.2.2): Object successfully set`
	`USERNAME lac.bigcompany.com: Object successfully set`
	
	`IPNOS: CONFIG VPN(50) VR(2.2.2.2)# **display ppp username**`
	
	`PPP USERNAME(s):`
	`USERNAME | PASSWORD`
	`fred | ********`
	`lac.bigcompany.com| ********`

Default Value

None

Command Level

VR

Related Commands

hostname
l2tp
pptp

vi

Go to the virtual interface (VI) command level. See "VI Commands" in Chapter 21 for a list of commands.

Syntax	*Example*
[CREATE | DELETE] VI <viId>	IPNOS : CONFIG VPN (1234) VR(1.1.1.1)# **vi 5** VR(1.1.1.1) Object successfully set IPNOS : CONFIG VPN(1234) VR(1.1.1.1) VI(5)#
[CREATE | DELETE] VI <viId>	IPNOS : CONFIG VPN(1234) VR(1.1.1.1)# **create vi 5** VR(1.1.1.1) Object successfully set IPNOS : CONFIG VPN(1234) VR(1.1.1.1) VI(5)#
[CREATE | DELETE] VI <viId>	IPNOS : CONFIG VPN(1234) VR(1.1.1.1)# **delete vi 5** VR(1.1.1.1) Object successfully set VI(5): Object successfully deleted IPNOS : CONFIG VPN(1234) VR(1.1.1.1)#

Default Value

None

Command Level

VR

Related Commands

vpn
vr

wins

Use the **wins** command to set the primary and secondary domain name server for Microsoft Windows name service (WINS). Also known as the Net BIOS name service (NBNS), the name server maintains a list of domain names in a network. Refer to "dns" on page 16-43 for the non-Microsoft version of the domain name service feature.

Syntax	*Example*
WINS {PRIMARY | SECONDARY} address	IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **wins primary 192.167.24.1** VR(1.1.1.1): Object successfully set WINS SERVER 192.167.24.1: Object successfully set IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **display WINS** WINS PRIMARY : 192.167.24.1 WINS SECONDARY : 0.0.0.0 IPNOS: CONFIG VPN(50) VR(1.1.1.1)#
WINS {PRIMARY | SECONDARY} address	IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **wins secondary 192.167.24.2** VR(1.1.1.1): Object successfully set NBNS SERVER 192.167.24.2: Object successfully set IPNOS: CONFIG VPN(50) VR(1.1.1.1)# **display wins** WINS PRIMARY : 192.167.24.1 WINS SECONDARY : 192.167.24.2 IPNOS: CONFIG VPN(50) VR(1.1.1.1)#

Default Value

None

Command Level

VR

Related Commands

dns
nbns

CHAPTER

17 Crypto Commands

Use these commands to configure Internet Protocol Security (IPSec). This chapter describes the following commands:

Command	*Page*
ipsec	17-3
transform-set	17-3
display	17-4
initvector	17-5
mode	17-6
isakmp	17-8
parameter	17-8
display	17-9
expire-timer	17-10
flag	17-10
local-secret-size	17-14
max-retry-timer	17-14
max-sa-count	17-15
retry-limit	17-16
retry-timer-sec	17-17
retry-timer-usec	17-18
secret-recreate	17-19
token-hash-type	17-20
policy	17-20
authentication	17-21

Command	Page
display	17-23
encryption	17-24
group	17-25
hash	17-26
lifetime	17-27
map	17-29
ipsec-dialup	17-30
display	17-30
match	17-32
set	17-33
ipsec-isakmp	17-39
display	17-39
match	17-40
set	17-40
ipsec-manual	17-46
display	17-47
match	17-47
set	17-48

ipsec

Go to the IPSec command level. Currently, only transform-set is available.

transform-set

Go to the transform set command level. Configure the transform set's encryption and authentication type when you create a transform set. Used during ISAKMP Phase II, it encrypts user data packets that go across the IPSec tunnel. Table 17-1 lists the encryption and authentication types.

Table 17-1 IPSec Transform Set Encryption and Authentication Types

Type	*Description*
esp-des	Encapsulated Security Protocol (ESP) using 56-bit Data Encryption Standard (DES) and no authentication.
esp-md5-hmac	ESP using hashed message authentication codes (HMACs) and message digest 5 (MD5) authentication.
esp-sha-hmac	ESP using 56-bit DES and HMAC with the secure hash algorithm (SHA-1) authentication.
esp-3des	ESP using 168-bit Data Encryption Standard (Triple-DES) and no authentication.
esp3-md5-hmac	ESP using Triple-DES and HMAC-MD5 authentication.
esp3-sha-hmac	ESP using Triple-DES and HMAC-SHA-1 authentication.
ah-md5-hmac	Authentication header (AH) mode with HMAC-MD5 authentication.
ah-sha-hmac	AH mode with HMAC-SHA-1 authentication.
ah-rfc1828	AH mode used in software emulation without crypto hardware (based on RFC 1828). *Note:* This mode is available in the ipsec-manual crypto map only.
esp-rfc1829	ESP using 56-bit DES. Used in software emulation without crypto hardware (based on RFC 1829).

The following commands are available at this level.

- "display" on page 17-4
- "initvector" on page 17-5
- "mode" on page 17-6

Syntax	*Example*
[NO] CRYPTO IPSEC TRANSFORM-SET <name> [<transform1>] [<transform2>] [<transform3>]	IPNOS : CONFIG VPN(50) VR(2.2.2.2)# **crypto ipsec transform-set set1 esp-3des** VR(2.2.2.2): Object successfully set IPNOS : CONFIG VPN(50) VR(2.2.2.2) TRANSFORM-SET(set1)# **dis** VPN : 1 VR : 10.1.1.1 NAME : set1 TRANSFORM-SET : esp-3des INIT-VECTOR : 8 MODE : TUNNEL
[NO] CRYPTO IPSEC TRANSFORM-SET <name> [<transform1>] [<transform2>] [<transform3>]	IPNOS : CONFIG VPN(50) VR(2.2.2.2)# **no crypto IPSec transform-set trans1 ah-md5-hmac** VR(2.2.2.2): Object successfully set TRANSFORM-SET trans1: Object successfully deleted IPNOS : CONFIG VPN(50) VR(2.2.2.2)#

Default Value

None

Command Level

transform-set

Related Commands

isakmp
map

display

View the current status of the IPSec transform set.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG VPN(50) VR(2.2.2.2) TRANSFORM-SET(trans1)# **display** VPN : 1 VR : 10.1.1.1 NAME : trans1 TRANSFORM-SET : esp-des INIT-VECTOR : 8 MODE : TUNNEL
DISPLAY <argument>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) TRANSFORM-SET(trans1)# **display mode** MODE : TUNNEL

Default Value

None

Command Level

transform-set

Related Commands

show

initvector

Set the bytes for cryptographic synchronization. ESP transform sets require an 8-byte initialization vector string, except for the software-only transform esp-rfc1829. AH mode and manual keying (pre-shared keys, or passwords) do not use the init-vector setting, so the value is ignored. Choose 4 or 8 bytes.

Syntax	*Example*
INIT-VECTOR {4 | 8}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) TRANSFORM-SET(trans1)# **init-vector 8** IPNOS : CONFIG VPN(50) VR(2.2.2.2) TRANSFORM-SET(trans1)# **display** VPN : 1 VR : 10.1.1.1 NAME : trans1 TRANSFORM-SET : esp-des INIT-VECTOR : 8 MODE : TUNNEL

Default Value

8 bytes

Command Level

transform-set

Related Commands

None

mode

Set the transport mode.

Syntax	*Example*
MODE {TUNNEL | TRANSPORT}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) TRANSFORM-SET(trans1)# **mode tunnel** IPNOS : CONFIG VPN(50) VR(2.2.2.2) TRANSFORM-SET(trans1)# **display mode** MODE : TUNNEL
MODE {TUNNEL | TRANSPORT}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) TRANSFORM-SET(trans1)# **mode transport** IPNOS : CONFIG VPN(50) VR(2.2.2.2) TRANSFORM-SET(trans1)# **display mode** MODE : TRANSPORT

Default Value

Tunnel

Command Level

transform-set

Related Commands

None

isakmp

Go the Internet Security Association Key Management Protocol (ISAKMP) command levels. Choose one of two levels to enter:

- "parameter" on page 17-8
- "policy" on page 17-20

parameter

Go to the ISAKMP parameter command level. Issue the command to activate ISAKMP/IKE Phase I server. The default parameter values should work with most gateways, but they can be modified to be compatible with older devices. The following is a list of the commands available.

- "display" on page 17-9
- "expire-timer" on page 17-10
- "flag" on page 17-10
- "local-secret-size" on page 17-14
- "max-retry-timer" on page 17-14
- "max-retry-timer" on page 17-14
- "max-sa-count" on page 17-15
- "retry-limit" on page 17-16
- "retry-timer-sec" on page 17-17
- "retry-timer-usec" on page 17-18
- "secret-recreate" on page 17-19
- "token-hash-type" on page 17-20

Syntax	*Example*

`CRYPTO ISAKMP PARAMETER`	`IPNOS : CONFIG VPN(50) VR(2.2.2.2)# crypto isakmp parameter`
	`VR(2.2.2.2): Object successfully set`
	
	`IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER#`

Default Value

None

Command Level

crypto ISAKMP

Related Commands

ipsec
map

display

View the current ISAKMP server configuration.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display** VPN : 1 VR : 10.1.1.1 NAME : default ADDRESS : 0.0.0.0 PORT : 500 RETRY-LIMIT : 10 RETRY-TIMER-SEC : 1 RETRY-TIMER-USEC: 200 MAX-RETRY-TIMER : 30 EXPIRE-TIMER : 300 SECRET-RECREATE : 14400 SPI-SIZE : 8 MAX-KEY-LEN : 64 MAX-SA-COUNT : 512 LOCAL-SECRET-SIZE: 32 TOKEN-HASH-TYPE : MD5 FLAG(s): List is empty
DISPLAY <argument>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display token** TOKEN-HASH-TYPE : MD5

Default Value

None

Command Level

ISAKMP parameter

Related Commands

show

expire-timer

Set the ISAKMP server time limit for Phase I negotiations.

Syntax	*Example*
EXPIRE-TIMER <value>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **expire-timer 400** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display expire** EXPIRE-TIMER : 400
NO EXPIRE-TIMER	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **no expire-timer** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display expire** EXPIRE-TIMER : 300

Default Value

300 seconds

Command Level

ISAKMP parameter

Related Commands

None

flag

Set backward compatibility flags. Table 17-2 describes each flag.

Table 17-2 ISAKMP Parameter Flags

Flag	*Description*
No	Disables a flag or reverts it to a default setting.

Table 17-2 ISAKMP Parameter Flags (Continued)

Flag	*Description*
Old-pk-encryption	Enables IPSec encapsulation according to the older IETF RFCs governing IPSec from 1995. By default, this flag should not be set.
Old-payload-padding	Allows the older packet payload padding scheme from previous versions of ISAKMP/IPSec.
Old-message-padding	Allows an older message padding scheme.
Old-da-padding	Allows an older DA padding scheme.
Old-id-in-hash	Allows an older ID in hashing scheme.
Old-cr-format	Allows an older version of the certificate request format.
Use-zero-spi	Allows padding zeros in security parameter indicies (SPIs). When enabled, the SPI field is filled with zeroes in the ISAKMP SA. If this value is set to FALSE (the default), the SPI is filled with ISAKMP SA cookies.
Ignore-cr-payload	Ignores all certificate requests payloads.

Syntax	*Example*
FLAG OLD-PK-ENCRYPTION	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **flag old-pk** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display flag** FLAG(s): OLD-PK-ENCRYPTION
FLAG OLD-PAYLOAD-PADDING	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **flag old-payload** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display flag** FLAG(s): OLD-PK-ENCRYPTION OLD-PAYLOAD_PADDING
FLAG OLD-MESSAGE-PADDING	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **flag old-message** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display flag** FLAG(s): OLD-PK-ENCRYPTION OLD-PAYLOAD_PADDING OLD-MESSAGE-PADDING

Syntax	*Example*
FLAG OLD-DA-PADDING	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **flag old-da** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display flag** FLAG(s): OLD-PK-ENCRYPTION OLD-PAYLOAD_PADDING OLD-MESSAGE-PADDING OLD-DA-PADDING
FLAG OLD-ID-IN-HASH	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **flag old-id** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display flag** FLAG(s): OLD-PK-ENCRYPTION OLD-PAYLOAD_PADDING OLD-MESSAGE-PADDING OLD-DA-PADDING OLD-ID-IN-HASH
FLAG OLD-CR-FORMAT	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **flag old-cr** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display flag** FLAG(s): OLD-PK-ENCRYPTION OLD-PAYLOAD_PADDING OLD-MESSAGE-PADDING OLD-DA-PADDING OLD-ID-IN-HASH IGNORE-CR-PAYLOAD
FLAG OLD-CR-PAYLOAD	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **flag ignore-cr** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display flag** FLAG(s): OLD-PK-ENCRYPTION OLD-PAYLOAD_PADDING OLD-MESSAGE-PADDING OLD-DA-PADDING OLD-ID-IN-HASH IGNORE-CR-PAYLOAD OLD-CR-FORMAT

Syntax	*Example*
FLAG USE-ZERO-SPI	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **flag use-zero** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display flag** FLAG(s): OLD-PK-ENCRYPTION OLD-PAYLOAD_PADDING OLD-MESSAGE-PADDING OLD-DA-PADDING OLD-ID-IN-HASH IGNORE-CR-PAYLOAD USE-ZERO-SPI
[NO] FLAG {[OLD-PK-ENCRYPTION] | [OLD-PAYLOAD-PADDING] | [OLD-MESSAGE-PADDING] | [OLD-DA-PADDING] | [OLD-ID-IN-HASH] | [OLD-CR-FORMAT] | [USE-ZERO-SPI] | [IGNORE-CR-PAYLOAD]}	PNOS: CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **no flag old-payload** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display flag** FLAG(s): OLD-PK-ENCRYPTION OLD-MESSAGE-PADDING OLD-DA-PADDING OLD-ID-IN-HASH IGNORE-CR-PAYLOAD OLD-CR-FORMAT USE-ZERO-SPI

Default Value

All flags are disabled.

Command Level

ISAKMP parameter

Related Commands

None

local-secret-size

Set the length of the local secret. Used to prevent denial-of-service attacks, local secrets create anti-clogging tokens. If you set the value to zero, the default value is used.

Syntax	*Example*
LOCAL-SECRET-SIZE <value>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **local-secret-size 256** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display local secret** LOCAL-SECRET-SIZE: 256 SECRET-RECREATE : 14400
NO LOCAL-SECRET-SIZE	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **no local-secret-size** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display local** LOCAL-SECRET-SIZE: 32

Default Value

256 bits (32 bytes)

Command Level

ISAKMP parameter

Related Commands

secret-recreate
max-retry-timer
token-hash-type

max-retry-timer

Set the maximum value of the retry timer.

Syntax	*Example*
MAX-RETRY-TIMER <seconds>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **max-retry-timer 55** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display max** MAX-RETRY-TIMER : 55 MAX-KEY-LEN : 64 MAX-SA-COUNT : 512
NO MAX-RETRY-TIMER	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **no max-retry-timer** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display max** MAX-RETRY-TIMER : 30 MAX-KEY-LEN : 64 MAX-SA-COUNT : 512

Default Value

30 seconds

Command Level

ISAKMP parameter

Related Commands

None

max-sa-count

Limit the number of ISAKMP Security Associations (SAs). If this limit is reached, the next SAs due to expire are disconnected to make room for new ones.

Syntax	*Example*
MAX-SA-COUNT <value>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **max-sa-count 256** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display max-sa** MAX-SA-COUNT : 256
NO MAX-SA-COUNT	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **no max-sa-count** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display max-sa** MAX-SA-COUNT : 512

Default Value

512 entries

Command Level

ISAKMP parameter

Related Commands

None

retry-limit

Limit resending ISAKMP packets. The server will retry a transmission if it does not receive a response in the number of seconds specified by **retry-timer-sec**. See "retry-timer-sec" on page 17-17.

Syntax	*Example*
RETRY-LIMIT <number_of_retries>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **retry-limit 5** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display retry** RETRY-LIMIT : 5 RETRY-TIMER-SEC : 1 RETRY-TIMER-USEC: 200

Syntax	Example
NO RETRY-LIMIT	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **no retry-limit** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display retry** RETRY-LIMIT : 10 RETRY-TIMER-SEC : 1 RETRY-TIMER-USEC: 200

Default Value

10 retry attempts

Command Level

ISAKMP parameter

Related Commands

None

retry-timer-sec

Set the time ISAKMP server gives the remote server to reply to a sent request. The packet is re-sent when the time is up.

Syntax	Example
RETRY-TIMER-SEC <seconds>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **retry-timer-sec 5** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display retry** RETRY-LIMIT : 10 RETRY-TIMER-SEC : 5 RETRY-TIMER-USEC: 200
NO RETRY-TIMER-SEC	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **no retry-timer-sec** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display retry** RETRY-LIMIT : 10 RETRY-TIMER-SEC : 1 RETRY-TIMER-USEC: 200

Default Value

1 second

Command Level

ISAKMP parameter

Related Commands

None

retry-timer-usec

Set the time (in milliseconds) the ISAKMP server gives the remote server to reply to a sent request. The packet is re-sent when the time is up. This parameter is useful for very fast SA peers. It does not need to be set if the **retry-timer-sec** value is specified.

Syntax	*Example*
RETRY-TIMER-USEC <micro_seconds>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **retry-timer-usec 250** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display retry** RETRY-LIMIT : 10 RETRY-TIMER-SEC : 1 RETRY-TIMER-USEC: 250
NO RETRY-TIMER-USEC	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **no retry-timer-usec** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display retry** RETRY-LIMIT : 10 RETRY-TIMER-SEC : 1 RETRY-TIMER-USEC: 200

Default Value

200 milliseconds

Command Level

ISAKMP parameter

Related Commands

None

secret-recreate

Set the timer for re-creating the local secret token. The local secret is periodically regenerated, and all subsequent anti-clogging tokens use the new secret. (Anti-clogging is a technique used to stop denial-of-service attacks, in which attempts are made to occupy the server's entire bandwidth with spurious requests for service). The maximum lifetime of an ISAKMP SA is twice the lifetime of the **secret-recreate** timer. If the value is kept at zero or null, the default value is used.

Syntax	*Example*
SECRET-RECREATE <seconds>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **secret-recreate 7200** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display secret** SECRET-RECREATE : 7200
NO SECRET-RECREATE	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **no secret-recreate** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display secret** SECRET-RECREATE : 14400

Default Value

14400 seconds (4 hours)

Command Level

ISAKMP parameter

Related Commands

local-secret-size
max-retry-timer
token-hash-type

token-hash-type

Define the hashing algorithm for the anti-clogging token. (The anti-clogging token is used to stop denial-of-service attacks.) Choose between message digest 5 (MD5) and secure hash algorithm (SHA1).

Syntax	*Example*
TOKEN-HASH-TYPE {SHA | MD5}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **token-hash-type sha** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display token** TOKEN-HASH-TYPE : SHA
TOKEN-HASH-TYPE {SHA | MD5}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **token-hash-type md5** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display token** TOKEN-HASH-TYPE : MD5
NO TOKEN-HASH-TYPE	IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **no token-hash-type** IPNOS : CONFIG VPN(50) VR(2.2.2.2) ISAKMP PARAMETER# **display token** TOKEN-HASH-TYPE : MD5

Default Value

MD5

Command Level

ISAKMP parameter

Related Commands

local-secret-size
max-retry-timer
secret-recreate

policy

Create or go to an ISAKMP Phase I policy. This policy sets the rules for initiating a Security Association (SA) between two devices. If you have multiple policies in operation, the priority number sets the order of

importance. This command will also put you in the ISAKMP policy level. The following commands are available.

- "authentication" on page 17-21
- "display" on page 17-23
- "encryption" on page 17-24
- "group" on page 17-25
- "hash" on page 17-26
- "lifetime" on page 17-27

Syntax	*Example*
[NO] CRYPTO ISAKMP POLICY <policy_name> <priority_number>	IPNOS : CONFIG VPN(50) VR(2.2.2.2)# **crypto isakmp policy joenet 1 1** VR(2.2.2.2): Object successfully set IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1 1)#
[NO] CRYPTO ISAKMP POLICY <policy_name> <priority_number>	IPNOS : CONFIG VPN(50) VR(2.2.2.2)# **no crypto isakmp policy joenet 1 1** VR(2.2.2.2): Object successfully set POLICY joenet 1 1: Object successfully deleted IPNOS : CONFIG VPN(50) VR(2.2.2.2)#

Default Value

None

Command Level

Crypto ISAKMP

Related Commands

None

authentication

Set a digital signature or a pre-shared key for ISAKMP Phase I authentication. Certificates are signed with a digital signature to verify their authenticity. Pre-shared key is a simple arbitrary hexadecimal value which is shared by both gateways. Refer to **ipsec-dialup** "set" on page 17-33 or **ipsec-isakmp**"set" on page 17-48 for setting a pre-shared key. Refer to "certificate" on page 16-17 for

setting up a certificate for the digital signature. Table 17-3 describes the options available.

Table 17-3 ISAKMP Policy Options for Certificate and Pre-Shared Key

Algorithm	*Description*
Pre-share	Pre-shared key. The actual key is set in the crypto map.
Dss-sig	Digital Signature Standard (DSS), which has no user packet encryption. Requires a certificate.
Rsa-sig	Rivest-Shamir-Adleman (RSA) digital signature. Requires a certificate.
Xauth-init-preshared	Pre-shared key with RADIUS server authentication. The actual key is set in the crypto map. Refer to "radius-server" on page 16-122 to configure the RADIUS server. Use only for isakmp-dialup crypto maps.
Xauth-init-rsa-sig	RSA with RADIUS server authentication. Refer to "radius-server" on page 16-122 to configure the RADIUS server. Use only for isakmp-dialup crypto maps.

Syntax	*Example*
[NO] AUTHENTICATION {PRE-SHARE | DSS-SIG | RSA-SIG | XAUTH-INIT-PRESHARED | XAUTH-INIT-RSA-SIG}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **authentication pre-share** IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **display auth** AUTHENTICATION : PRE-SHARE
[NO] AUTHENTICATION {PRE-SHARE | DSS-SIG | RSA-SIG | XAUTH-INIT-PRESHARED | XAUTH-INIT-RSA-SIG}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **authentication dss-sig** IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **display auth** AUTHENTICATION : DSS-SIG
[NO] AUTHENTICATION {PRE-SHARE | DSS-SIG | RSA-SIG | XAUTH-INIT-PRESHARED | XAUTH-INIT-RSA-SIG}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **authentication rsa-sig** IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **display auth** AUTHENTICATION : RSA-SIG
[NO] AUTHENTICATION {PRE-SHARE | DSS-SIG | RSA-SIG | XAUTH-INIT-PRESHARED | XAUTH-INIT-RSA-SIG}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **authentication auth-init-preshared** IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **display auth** AUTHENTICATION : XAUTH-INIT-PRESHARED

Syntax	Example
[NO] AUTHENTICATION {PRE-SHARE | DSS-SIG | RSA-SIG | XAUTH-INIT-PRESHARED | XAUTH-INIT-RSA-SIG}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **authentication xauth-init-rsa-sig** IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **display auth** AUTHENTICATION : XAUTH-INIT-RSA-SIG
[NO] AUTHENTICATION {PRE-SHARE | DSS-SIG | RSA-SIG | XAUTH-INIT-PRESHARED | XAUTH-INIT-RSA-SIG}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **no authentication** IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **display auth** AUTHENTICATION : RSA-SIG

Default Value

RSA-SIG (RSA signature authentication)

Command Level

ISAKMP policy

Related Commands

None

display

View the current ISAKMP policy configuration.

Syntax	Example
DISPLAY	IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **display** VPN : 1 VR : 10.1.1.1 NAME : joenet PRIORITY : 1 HASH : SHA ENCRYPTION : 3DES AUTHENTICATION : RSA-SIG GROUP : 1 SOFT-SECONDS : 10800 SOFT-KBYTES : 4096 HARD-SECONDS : 10800 HARD-KBYTES : 4096

Syntax	*Example*
DISPLAY <argument>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **display** hash HASH : SHA

Default Value

None

Command Level

ISAKMP policy

Related Commands

None

encryption

Set the encryption depth of the ISAKMP Phase I exchange. The 128-bit triple-DES is the default, but 56-bit DES is also available.

Syntax	*Example*
ENCRYPTION {DES | 3DES}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **encryption des** IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **display encryption** ENCRYPTION : DES
ENCRYPTION {DES | 3DES}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **encryption 3des** IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **display encryption** ENCRYPTION : 3DES
NO ENCRYPTION	IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **no encryption** IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **display encryption** ENCRYPTION : 3DES

Default Value

3DES

Command Level

ISAKMP policy

Related Commands

None

group

Set the Diffie-Hellman Group key exchange system for ISAKMP Phase I. Group 1 supports 768-bit depth. Group 2 supports 1024-bit depth. Group 5 supports 1536-bit depth.

Syntax	*Example*
GROUP {1 | 2 | 5}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **group 1** IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **display group** GROUP : 1
GROUP {1 | 2 | 5}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **group 2** IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **display group** GROUP : 2
GROUP {1 | 2 | 5}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **group 5** IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **display group** GROUP : 5
NO GROUP	IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **no group** IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **display group** GROUP : 1

Default Value

Group 1

Command Level

ISAKMP policy

Related Commands

None

hash

Set the hashing engine for performing complex integrity checking for ISAKMP Phase I. MD5 and SHA-1 are available. The ISAKMP/IKE hash setting may differ from that used by the IPSec tunnel in Phase II. Refer to "transform-set" on page 17-3 for more information on Phase II hashing.

Syntax	*Example*
HASH {MD5 | SHA}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **hash md5** IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **display hash** HASH : MD5
HASH {MD5 | SHA}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **hash sha** IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **display hash** HASH : SHA
NO HASH	IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **no hash** IPNOS : CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet 1)# **display hash** HASH : SHA

Default Value

None

Command Level

ISAKMP policy

Related Commands

None

lifetime

Set the lifetime of an ISAKMP/IKE Phase I key before it expires. If 90% of the lifetime limit is reached, the ISAKMP/IKE Phase I SA is deleted. Currently existing Phase II SAs negotiated under that Phase I SA will stay active. The Phase I SA rekey initiates when a Phase II rekey initiates. If both gateways involved with the SA have different lifetime settings, the shorter lifetime setting will be used for Phase I. Table 17-4 describes the parameters available.

Table 17-4 ISAKMP Lifetime Settings

Setting	*Definition*
Hard-seconds	Seconds before the SA is completely deleted. The possible values are 0, and 60 to 2,147,483,647 (2 billion) seconds. The default value is 10,800 seconds (3 hours).
Hard-kilobytes	Kilobytes transferred before the SA is completely deleted. The possible values are 0, and 256 to 2,147,483,647 (2 billion) kbytes. The default value is 4,096 kilobytes.

NOTE: Cisco and Nortel routers currently do not accept hard-kilobyte lifetime limits for ISAKMP/IKE Phase I. For compatibility, set the hard-kilobytes lifetime to the 0 kilobytes.

Syntax	*Example*

`LIFETIME hard-secs [hard-kbytes]`	`IPNOS: CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet1 1)#`
	`lifetime 1260`
	
	`IPNOS: CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet1 1)#`
	`display life`
	`HARD-SECONDS  : 1260`
	`HARD-KBYTES  : 4096`

Reset to defaults:
`NO LIFETIME`
	`IPNOS: CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet1 1)#`
	`no lifetime`
	
	`IPNOS: CONFIG VPN(50) VR(2.2.2.2) POLICY(joenet1 1)#`
	`display life`
	`HARD-SECONDS  : 10800`
	`HARD-KBYTES  : 4096`

Default Value

See Table 17-4

Command Level

ISAKMP policy

Related Commands

set security-association lifetime (IPSec-dialup crypto map)
set security-association lifetime (IPSec-ISAKMP crypto map)

map

Go to the crypto map command levels. Choose from three crypto map types:

- "ipsec-dialup" on page 17-30
- "ipsec-isakmp" on page 17-39
- "ipsec-manual" on page 17-46

An IPSEC-ISAKMP crypto map links ACLs, transform sets and ISAKMP policies together to define a particular SA. An IPSEC-dialup crypto map is similar in function to the IPSEC-ISAKMP crypto map. IPSEC-dialup supports remote VPN clients (such as a laptop PC using a modem) by setting up a secure gateway capable of sending them address and configuration information.

An IPSEC-Manual crypto map is a manually defined SA, usually used for testing IPSec connections. Each map has a name and a priority number. Refer to "acl" on page 16-7, "policy" on page 17-20, and "transform-set" on page 17-3 for more information. Use the **no** option to delete a crypto map.

Syntax	*Example*
[NO] CRYPTO MAP <name> <priority> {IPSEC-DIALUP | IPSEC-MANUAL | IPSEC-ISAKMP}	IPNOS : CONFIG VPN(50) VR(2.2.2.2)# **crypto map iremap1 3 ipsec-dialup** VR(2.2.2.2): Object successfully set IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)#
[NO] CRYPTO MAP <name> <priority> {IPSEC-DIALUP | IPSEC-MANUAL | IPSEC-ISAKMP}	IPNOS : CONFIG VPN(50) VR(2.2.2.2)# **crypto map joemap1 1 ipsec-isakmp** VR(2.2.2.2): Object successfully set IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)#
[NO] CRYPTO MAP <name> <priority> {IPSEC-DIALUP | IPSEC-MANUAL | IPSEC-ISAKMP}	IPNOS : CONFIG VPN(50) VR(2.2.2.2)# **crypto map marymap1 2 ipsec-manual** VR(2.2.2.2): Object successfully set IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-MANUAL(marymap1 2)#
[NO] CRYPTO MAP <name> <priority> {IPSEC-DIALUP | IPSEC-MANUAL | IPSEC-ISAKMP}	IPNOS : CONFIG VPN(50) VR(2.2.2.2)# **no crypto map marymap1 2 ipsec-manual** VR(2.2.2.2): Object successfully set IPSEC-MANUAL(marymap1 2): Object successfully deleted IPNOS : CONFIG VPN(50) VR(2.2.2.2)#

Default Value

None

Command Level

VR crypto

Related Commands

None

ipsec-dialup

Define a crypto map for an SA for a remote dialup access connection. This command puts you in the **ipsec-dialup** command level. The following commands are available.

- "display" on page 17-39
- "match" on page 17-40
- "set" on page 17-40

display

Use the **display** command to show the current status of the ipsec-dialup crypto map. You can also limit the display to only the items you want to see.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display** VPN : 1 VR : 10.1.1.1 NAME : iremap1 NUMBER : 1 MATCH ADDRESS : 1 PFS-GROUP : 1 TRANSFORM-SET : set1 IKE-KEY : IKE-POLICY : joenet1 LOCAL-ISAKMP-IDENTITY: C=US, O=Cosine Communications, CN=qa-1 HARD-SECONDS : 28800 HARD-KBYTES : 4096 IDLE-TIMEOUT : 1800 FLAG(s): SA-PER-NET MAIN-MODE
DISPLAY <argument>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display name** NAME : iremap1 NUMBER : 1

Default Value

None

Command Level

ipsec-dialup

Related Commands

show
ipsec-manual display
ipsec-isakmp display

match

Assign an access control list (ACL). The ACL is used as the ID for ISAKMP Phase II exchanges. See "acl" on page 16-7.

Syntax	*Example*
MATCH ADDRESS <acl_Id>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap13)# **match address 1** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap13)# **display match** MATCH ADDRESS : 1

Default Value

None

Command Level

ipsec-dialup

Related Commands

ipsec-manual match
ipsec-isakmp match

set

Set crypto map parameters. Table 17-5 describes the options available.

Table 17-5 IPSEC Dialup Set Options

Parameter	*Description*
Flag	Configure the following flags: • PFS—Optional. Request perfect forward secrecy (PFS) key exchange for each network or host during ISAKMP Phase II. This flag requires you to set the PFS-Group parameter for Diffie-Hellman key exchange. • Identity-PFS—Optional. Request a complete renegotiation of the entire SA. The ISKAMP Phase I Key and ID exchange are renegotiated. • SA-Per-Net—Allow one ISKAMP Phase II exchange per network. All matching connections are combined into the single SA, including any host-to-host or host-to-net communications. This flag is enabled by default. • Aggressive-Mode—Force the ISAKMP Phase I negotiations to take place in fewer packets. This increases the speed of the exchange, but the host ID is not protected. If aggressive-mode is set, main-mode is automatically disabled. • Main-Mode—Allow a standard ISAKMP Phase I six-packet exchange to take place. This exchange is slower, but the host ID is protected. If Main mode is set, aggressive-mode is automatically disabled. This flag is on by default.
Idle-Timeout	Timeout in seconds for an idle dialup connection. Setting this object to 0 will turn the timer off. The minimum valid timeout value for a dialup connection is 60 seconds. The default value is 1800 seconds.
IKE-Key	You can manually type in a hexadecimal pre-shared key for ISAKMP Phase I instead of a digital signature. Refer to “authentication” on page 17-21 for information on configuring the ISAKMP policy for a digital signature or a pre-shared key. By default, this field is left blank.
IKE-Policy	ISAKMP policy used for ISAKMP Phase I. Multiple policies may be loaded for successive querying with peers.

Table 17-5 IPSEC Dialup Set Options (Continued)

Parameter	*Description*
Local-ISAKMP-Identity	Enter an identity. If you use pre-shared keys, use either an Internet Protocol version 4 IP address or a fully qualified domain name (FQDN). Use the following formats: • IPV4 <x.x.x.x> • FQDN <company.com> • DIST-NAME C=<country>, O=<organization>, OU=<organization unit>, DNQ=<domain name qualifier>, ST=<state or province>, CN=<common name> If you use digital certificates, use the digital certificate's X509 distinguished name for the ISAKMP Phase I Initiator ID (to identify the certificate for authentication.) The name ties the crypto map to the certificate for the SA connection. Refer to "certificate" on page 16-17. The following distinguished name fields are supported: • C—Country of origin. This is the only required field. • O—Organization name • OU—Organizational unit such as department or division • DNQ—Domain name qualifer is any value used to distinguish between multiple certificates if all the other fields are the same. • ST—State or province • CN—Common Name such as the system administrator's name
PFS-Group	Diffie-Hellman group for ISAKMP Phase II key exchanges. Required if the PFS flag is set. Possible values are 1, 2, or 5. The default value is 1.
Security-Association	Lifetime of an IPSec Phase II key before it expires. The Security Association lifetimes are expressed as "Hard" limits. When the time or traffic volume limits are reached, the IPSec Phase II SA is deleted. No further traffic is accepted or sent protected under that SA. Also, if 90% of the lifetime limit is reached, the IPSec Phase II SA rekey initiates. Other existing Phase II SAs are not affected. If the parent Phase I SA does not exist when Phase II SA rekeys, the Phase I is also rekeyed. If the hard-seconds or hard-kilobytes are set to zero, then the default values are used for rekeying. The default values are 28,800 hard-seconds and 4,095 hard-kilobytes. Refer to "ISAKMP Lifetime Settings" on page 17-27 for details. *Note:* Phase II lifetime should be less than the Phase I lifetime.
Transform-Set	Name of the IPSec transform set for ISAKMP Phase II. It is used for user data packet encryption and authentication. Multiple sets may also be loaded.

Syntax	*Example*
[NO] SET FLAG {[PFS | IDENTITY-PFS] [SA-PER-HOST |SA-PER-NET] [AGGRESSIVE-MODE | MAIN-MODE]}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **set flag pfs** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display flag** FLAG(s): PFS SA-PER-NET MAIN-MODE
[NO] SET FLAG {[PFS | IDENTITY-PFS] [SA-PER-HOST |SA-PER-NET] [AGGRESSIVE-MODE | MAIN-MODE]}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **set flag sa-per-net** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display flag** FLAG(s): PFS SA-PER-NET MAIN-MODE
[NO] SET FLAG {[PFS | IDENTITY-PFS] [SA-PER-HOST |SA-PER-NET] [AGGRESSIVE-MODE | MAIN-MODE]}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **set flag main-mode** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display flag** FLAG(s): PFS SA-PER-NET MAIN-MODE
[NO] SET FLAG {[PFS | IDENTITY-PFS] [SA-PER-HOST |SA-PER-NET] [AGGRESSIVE-MODE | MAIN-MODE]}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **no set flag pfs** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display flag** FLAG(s): SA-PER-NET MAIN-MODE
[NO] SET IDLE-TIMEOUT <value-in-secs>	IPNOS: CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **set idle-timeout 1200** IPNOS: CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display idle-timeout** IDLE-TIMEOUT : 1200

Syntax	*Example*
[NO] SET IDLE-TIMEOUT	IPNOS: CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **no set idle-timeout** IPNOS: CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display idle-timeout** IDLE-TIMEOUT : 1800
SET IKE-KEY <hex_key>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **set ike-key 1a2b3c4d5e** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display ike** IKE-KEY : 1a2b3c4d5e IKE-POLICY :
NO SET IKE-KEY	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **no set ike-key** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display ike** IKE-KEY : IKE-POLICY : joenet1
SET IKE-POLICY <name>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **set ike-policy joenet1** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display ike** IKE-KEY : 1a2b3c4d5e IKE-POLICY : joenet1
NO SET IKE-POLICY	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **no set ike-policy** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display ike** IKE-KEY : IKE-POLICY :
SET LOCAL-ISAKMP-IDENTITY {IPV4 <w.x.y.z> | FQDN <company.com>| DIST-NAME C=<country>, [O=<organization>,] [OU=<organization unit>,] [DNQ=<domain name qualifier>,] [ST=<state or province>,] [CN=<common name>]}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **set local-isakmp-identity ipv4 192.128.53.21** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display local** LOCAL-ISAKMP-IDENTITY: 192.128.53.21

Syntax	Example
SET LOCAL-ISAKMP-IDENTITY {IPV4 <w.x.y.z> | FQDN <company.com>| DIST-NAME C=<country>, [O=<organization>,] [OU=<organization unit>,] [DNQ=<domain name qualifier>,] [ST=<state or province>,] [CN=<common name>]}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **set local-isakmp-identity fqdn cosinecom.com** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display local** LOCAL-ISAKMP-IDENTITY: cosinecom.com
SET LOCAL-ISAKMP-IDENTITY {IPV4 <w.x.y.z> | FQDN <company.com>| DIST-NAME C=<country>, [O=<organization>,] [OU=<organization unit>,] [DNQ=<domain name qualifier>,] [ST=<state or province>,] [CN=<common name>]}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **set local-isakmp-identity DIST-NAME C=USA, O=CoSine Communications, OU=engineering, DNQ=level1, ST=California, CN=Joe Schmoe** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display local** LOCAL-ISAKMP-IDENTITY: C=USA, O=CoSine Communications, OU=engineering, DNQ=level1, ST=California, CN=Joe Schmoe
NO SET LOCAL-ISAKMP-IDENTITY	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **no set local-isakmp-identity** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display local** LOCAL-ISAKMP-IDENTITY:
SET PFS-GROUP {1 | 2 | 5}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **set pfs-group 2** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display pfs** PFS-GROUP : 2
NO SET PFS-GROUP	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **no set pfs-group** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display pfs-group** PFS-GROUP : 1

Syntax	*Example*
SET SECURITY-ASSOCIATION LIFETIME hard-secs [hard-kbytes]	IPNOS: CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **set security-association lifetime 14400** IPNOS: CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display security** HARD-SECONDS : 14400 HARD-KBYTES : 4096
NO SET SECURITY-ASSOCIATION LIFETIME	IPNOS: CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 1)# **no set security-association lifetime** IPNOS: CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 1)# **display security** HARD-SECONDS : 28800 HARD-KBYTES : 4096
SET TRANSFORM-SET <transform_name1> [<transform_name2> ... <transform_name6>]	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **set transform-set set1** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display transform-set** TRANSFORM-SET : set1
SET TRANSFORM-SET <transform_name1> [<transform_name2> ... <transform_name6>]	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **set transform-set set1 set2** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display transform-set** TRANSFORM-SET : set1 set2
NO SET TRANSFORM-SET	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **no set transform-set** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display transform-set** TRANSFORM-SET :

Default Value

None

Command Level

ipsec-dialup

Related Commands

ipsec-manual set
ipsec-isakmp set

ipsec-isakmp

Define a crypto map for connections between network gateways (such as the IPSX switch). This command puts you in the **ipsec-isakmp** command level. The following commands are available.

- "display" on page 17-39
- "match" on page 17-40
- "set" on page 17-40

display

View the current ipsec-isakmp crypto map configuration.

Syntax	*Example*
DISPLAY	IPNOS: CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(map1 1)# **display** VPN : 50 VR : 2.2.2.2 NAME : map1 NUMBER : 1 MATCH ADDRESS : 0 PFS-GROUP : 1 TRANSFORM-SET : IKE-KEY : IKE-POLICY : LOCAL-ISAKMP-IDENTITY: HARD-SECONDS : 28800 HARD-KBYTES : 4096 FLAG(s): SA-PER-NET MAIN-MODE
DISPLAY <argument>	IPNOS: CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(map1 1)# **display name** NAME : map1 NUMBER : 1

Default Value

None

Command Level

ipsec-isakmp

Related Commands

show

match

Assign an access control list (ACL). The ACL is used as the ID for ISAKMP Phase II exchanges. See "acl" on page 16-46.

Syntax	*Example*
MATCH ADDRESS <acl_Id>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **match address 1** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **display match** MATCH ADDRESS : 1

Default Value

None

Command Level

ipsec-isakmp

Related Commands

ipsec-manual match

set

Set the ipsec-isakmp crypto map parameters. Table 17-6 describes the parameters available.

Table 17-6 IPSEC ISAKMP Set Parameters

Parameter	*Description*
Flag	Configures the following security association (SA) flags: • PFS—Optional. Request perfect forward secrecy (PFS) key exchange for each network or host during ISAKMP Phase II. This flag requires you to set the PFS-Group parameter for Diffie-Hellman key exchange. • Identity-PFS—Optional. Request a complete renegotiation of the entire SA. The ISKAMP Phase I Key and ID exchange are renegotiated. • SA-Per-Host—Use a separate ISAKMP Phase II exchange for each host on the network that is part of the SA. This adds more overhead to the network, but ensures better protection. This flag is disabled by default. • SA-Per-Net—Allow one ISKAMP Phase II exchange per network. All matching connections are combined into the single SA, including any host-to-host or host-to-net communications. This flag is enabled by default. • Aggressive-Mode—Force the ISAKMP Phase I negotiations to take place in fewer packets. This increases the speed of the exchange, but the host ID is not protected. If aggressive-mode is set, main-mode is automatically disabled. • Main-Mode—Allow a standard ISAKMP phase I six-packet exchange to take place. This exchange is slower, but the host ID is protected. If Main mode is set, aggressive-mode is automatically disabled. This flag is on by default.
IKE-Key	Manually type in a hexadecimal pre-shared key for ISAKMP Phase I instead of a digital signature. Refer to “authentication” on page 17-21 for information on configuring the ISAKMP policy for a digital signature or a pre-shared key. By default, this field is left blank.
IKE-Policy	ISAKMP policy used for ISAKMP Phase I. Multiple policies may be loaded for successive querying with peers.

Table 17-6 IPSEC ISAKMP Set Parameters (Continued)

Parameter	*Description*
Local-ISAKMP-Identity	Enter an identity. If you use pre-shared keys, use either an Internet Protocol version 4 IP address or a fully qualified domain name (FQDN). Use the following formats: • ipv4 x.x.x.x • fqdn company.com • DIST-NAME C=<country>, O=<organization>, OU=<organization unit>, DNQ=<domain name qualifier>, ST=<state or province>, CN=<common name> Digital certificate require an X509 distinguished name for the ISAKMP Phase I Initiator ID (to identify the certificate for authentication.) The name ties the crypto map to the certificate for the SA connection. See “certificate” on page 16-17. The following distinguished name fields are supported: • C—Country of origin. This is the only required field. • O—Organization name • OU—Organizational unit such as department or division • DNQ—Domain name qualifer is any value used to distinguish between multiple certificates if all the other fields are the same. • ST—State or province • CN—Common Name such as the system administrator’s name
PFS-Group	Diffie-Hellman group for ISAKMP Phase II key exchanges. Required if the PFS flag is set. Possible values are 1, 2, or 5. The default value is 1.
Security-Association	The lifetime of an IPSec Phase II key before it expires. The Security Association lifetimes are expressed as "Hard" limits. When the time or traffic volume limits are reached, the IPSec Phase II SA is deleted. No further traffic is accepted or sent protected under that SA. Also, if 90% of the lifetime limit is reached, the IPSec Phase II SA rekey initiates. Other existing Phase II SAs are not affected. If the parent Phase I SA does not exist when Phase II SA rekeys, the Phase I is also rekeyed. If the hard-seconds or hard-kilobytes are set to zero, then the default values are used for rekeying. The default values are 28,800 hard-seconds and 4,095 hard-kilobytes. Refer to “ISAKMP Lifetime Settings” on page 17-27 for details. *Note:* Phase II lifetime should be less than the Phase I lifetime.
Transform-Set	Name of the IPSec transform set for ISAKMP Phase II. It is used for user data packet encryption and authentication. Multiple sets may also be loaded.

Syntax	*Example*
[NO] SET FLAG {[PFS | IDENTITY-PFS] [SA-PER-HOST |SA-PER-NET] [AGGRESSIVE-MODE | MAIN-MODE]}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **set flag pfs** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **display flag** FLAG(s): PFS SA-PER-NET MAIN-MODE
[NO] SET FLAG {[PFS | IDENTITY-PFS] [SA-PER-HOST |SA-PER-NET] [AGGRESSIVE-MODE | MAIN-MODE]}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **set flag sa-per-net** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **display flag** FLAG(s): PFS SA-PER-NET MAIN-MODE
[NO] SET FLAG {[PFS | IDENTITY-PFS] [SA-PER-HOST |SA-PER-NET] [AGGRESSIVE-MODE | MAIN-MODE]}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **set flag main-mode** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **display flag** FLAG(s): PFS SA-PER-NET MAIN-MODE
[NO] SET FLAG {[PFS | IDENTITY-PFS] [SA-PER-HOST |SA-PER-NET] [AGGRESSIVE-MODE | MAIN-MODE]}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **no set flag pfs** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **display flag** FLAG(s): SA-PER-NET MAIN-MODE
SET IKE-KEY <hex_key>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **set ike-key 1a2b3c4d5e** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **display ike** IKE-KEY : 1a2b3c4d5e IKE-POLICY :

Syntax	*Example*
NO SET IKE-KEY	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **no set ike-key** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **display ike** IKE-KEY : IKE-POLICY : joenet1
SET IKE-POLICY <name>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **set ike-policy joenet1** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **display ike** IKE-KEY : 1a2b3c4d5e IKE-POLICY : joenet1
NO SET IKE-POLICY	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **no set ike-policy** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **display ike** IKE-KEY : IKE-POLICY :
SET LOCAL-ISAKMP-IDENTITY {IPV4 <w.x.y.z> | FQDN <company.com>| DIST-NAME C=<country>, [O=<organization>,] [OU=<organization unit>,] [DNQ=<domain name qualifier>,] [ST=<state or province>,] [CN=<common name>]}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(joemap1 3)# **set local-isakmp-identity ipv4 192.128.53.21** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display local** LOCAL-ISAKMP-IDENTITY: 192.128.53.21
SET LOCAL-ISAKMP-IDENTITY {IPV4 <w.x.y.z> | FQDN <company.com>| DIST-NAME C=<country>, [O=<organization>,] [OU=<organization unit>,] [DNQ=<domain name qualifier>,] [ST=<state or province>,] [CN=<common name>]}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(joemap1 3)# **set local-isakmp-identity fqdn cosinecom.com** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display local** LOCAL-ISAKMP-IDENTITY: cosinecom.com

Syntax	Example
SET LOCAL-ISAKMP-IDENTITY {IPV4 <w.x.y.z> | FQDN <company.com>| DIST-NAME C=<country>, [O=<organization>,] [OU=<organization unit>,] [DNQ=<domain name qualifier>,] [ST=<state or province>,] [CN=<common name>]}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(joemap1 3)# **set local-isakmp-identity DIST-NAME C=USA, O=CoSine Communications, OU=engineering, DNQ=level1, ST=California, CN=Joe Schmoe** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-DIALUP(iremap1 3)# **display local** LOCAL-ISAKMP-IDENTITY: C=USA, O=CoSine Communications, OU=engineering, DNQ=level1, ST=California, CN=Joe Schmoe
NO SET LOCAL-ISAKMP-IDENTITY	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **no set local-isakmp-identity** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **display local** LOCAL-ISAKMP-IDENTITY:
SET PFS-GROUP {1 | 2}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **set pfs-group 2** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **display pfs** PFS-GROUP : 2
NO SET PFS-GROUP	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **no set pfs-group** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **display pfs-group** PFS-GROUP : 1
SET SECURITY-ASSOCIATION LIFETIME hard-secs [hard-kbytes]	IPNOS: CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **set security-association lifetime 14400** IPNOS: CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **display security** HARD-SECONDS : 14400 HARD-KBYTES : 4096
NO SET SECURITY-ASSOCIATION LIFETIME	IPNOS: CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **no set security-association lifetime** IPNOS: CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **display security** HARD-SECONDS : 28800 HARD-KBYTES : 4096

Syntax	*Example*
SET TRANSFORM-SET <transform_name1> [<transform_name2> ... <transform_name6>]	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **set transform-set set1** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **display transform-set** TRANSFORM-SET : set1
SET TRANSFORM-SET <transform_name1> [<transform_name2> ... <transform_name6>]	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **set transform-set set1 set2** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **display transform-set** TRANSFORM-SET : set1 set2
NO SET TRANSFORM-SET	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **no set transform-set** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-ISAKMP(joemap1 1)# **display transform-set** TRANSFORM-SET :

Default Value

None

Command Level

ipsec-isakmp

Related Commands

ipsec-manual set

ipsec-manual

Manually define an SA connection between each gateway (such as an IPSX switch). This command puts you in the **ipsec-manual** command level. This level is generally used only for testing purposes. The following commands are available.

- "display" on page 17-47
- "match" on page 17-47
- "set" on page 17-48

display

View the current ipsec-manual crypto map configuration.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-MANUAL(marymap1 2)# **display** VPN : 1 VR : 10.1.1.1 NAME : marymap1 NUMBER : 2 MATCH ADDRESS : 0 TRANSFORM-SET : INBOUND AH SPI : 0 INBOUND AH KEY : OUTBOUND AH SPI : 0 OUTBOUND AH KEY : INBOUND ESP SPI : 0 INBOUND ESP CIPHER: INBOUND ESP AUTHENTICATOR: OUTBOUND ESP SPI: 0 OUTBOUND ESP CIPHER: OUTBOUND ESP AUTHENTICATOR:
DISPLAY <argument>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-MANUAL(marymap1 2)# **display name** NAME : marymap1 NUMBER : 2

Default Value

None

Command Level

ipsec-manual

Related Commands

show

match

Assign an access control list (ACL). The ACL is used as the ID for ISAKMP Phase II exchanges. See “acl” on page 16-46.

Syntax	*Example*
MATCH ADDRESS <acl_Id>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-MANUAL(marymap1 2)# **match address 1** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-MANUAL(marymap1 2)# **display match** MATCH ADDRESS : 1

Default Value

None

Command Level

ipsec-manual

Related Commands

ipsec-isakmp match

set

Configure session keys and transform sets. Session keys are hexadecimal values shared between IPSec peers in an SA, which include authentication and encryption keys and SPIs for inbound and outbound traffic. Table 17-7 describes the settings available.

Table 17-7 IPSec-Manual Settings for Session Keys and Transform Sets

Setting	*Definition*
INBOUND AH SPI	Operator-defined security parameter index value for the inbound AH security association, if AH is being used.
INBOUND AH KEY	Operator-defined authentication key value for the inbound AH security association, if AH is being used.
OUTBOUND AH SPI	Operator-defined security parameter index value for the outbound AH security association, if AH is being used.
OUTBOUND AH KEY	Operator-defined session key value for the outbound AH security association, if AH is being used.
INBOUND ESP SPI	Operator-defined security parameter index value for the inbound ESP security association, if ESP is being used.

Table 17-7 IPSec-Manual Settings for Session Keys and Transform Sets (Continued)

Setting	*Definition*
INBOUND ESP CIPHER	Encryption key for the inbound ESP data encryption session. This key enables the peer to decrypt incoming encrypted packets.
INBOUND ESP AUTHENTICATOR	Separate hex key value used as an authentication key for the inbound ESP session. If no value is placed here, HMAC authentication will not be used in the SA.
OUTBOUND ESP SPI	Operator-defined security parameter index value for the outbound ESP security association.
OUTBOUND ESP CIPHER	Session key for the outbound ESP data encryption session. This key enables the peer to encrypt and send incoming encrypted packets.
OUTBOUND ESP AUTHENTICATOR	Separate hex key value used as an authentication key for the outbound ESP session. If no value is placed here, HMAC authentication will not be used in the SA.
TRANSFORM-SET	Name of the IPSec transform set. You can load only one transform set. Refer to "ipsec" on page 17-3 for more information.

The length of the keys can vary widely. Table 17-8 defines key sizes, in bytes, that can be inserted into manual crypto maps. For more information on how to use the keys, refer to the *CoSine IPNOS CLI Configuration Guide.*

Table 17-8 IPSec Session Key Sizes (in bytes)

Transform-Set	*Cipher Key Size*	*AH Key Size*	*Authenticator Key Size*	*Init-Vector Size*
ah-rfc1828	N/A	16	N/A	N/A
esp-des-rfc1829	8	N/A	N/A	4 or 8
ah-md5-hmac	N/A	16	N/A	N/A
ah-sha-hmac	N/A	20	N/A	N/A
esp-des	8	N/A	N/A	8
esp-3des	24	N/A	N/A	8
esp-md5-hmac	8	N/A	16	8
esp-sha-hmac	8	N/A	20	8
esp3-md5-hmac	24	N/A	16	8
esp3-sha-hmac	24	N/A	20	8

Syntax	*Example*
SET SESSION-KEY {INBOUND | OUTBOUND} AH <spi_number> <hex_key>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-MANUAL(marymap1 2)# **set session-key inbound ah 1234 a4d3f2c1** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-MANUAL(marymap1 2)# **display session** INBOUND AH SPI : 1234 INBOUND AH KEY : a4d3f2c1 OUTBOUND AH SPI : 0 OUTBOUND AH KEY : INBOUND ESP SPI : 1234 INBOUND ESP CIPHER: a12b3c4d INBOUND ESP AUTHENTICATOR: 5f2d33ab OUTBOUND ESP SPI: 4321 OUTBOUND ESP CIPHER: d4c3b21a OUTBOUND ESP AUTHENTICATOR: ba33d2f5
SET SESSION-KEY {INBOUND | OUTBOUND} AH <spi_number> <hex_key>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-MANUAL(marymap1 2)# **set session-key outbound ah 4321 1c2f3d4a** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-MANUAL(marymap1 2)# **display session** INBOUND AH SPI : 1234 INBOUND AH KEY : a4d3f2c1 OUTBOUND AH SPI : 4321 OUTBOUND AH KEY : 1c2f3d4a INBOUND ESP SPI : 1234 INBOUND ESP CIPHER: a12b3c4d INBOUND ESP AUTHENTICATOR: 5f2d33ab OUTBOUND ESP SPI: 4321 OUTBOUND ESP CIPHER: d4c3b21a OUTBOUND ESP AUTHENTICATOR: ba33d2f5

Syntax	*Example*
SET SESSION-KEY {INBOUND | OUTBOUND} ESP <spi_number> CIPHER <hex_key> [AUTHENTICATOR <hex_key>]	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-MANUAL(marymap1 2)# **set session-key inbound esp 1234 cipher a12b3c4d** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-MANUAL(marymap1 2)# **display session** INBOUND AH SPI : 0 INBOUND AH KEY : OUTBOUND AH SPI : 0 OUTBOUND AH KEY : INBOUND ESP SPI : 1234 INBOUND ESP CIPHER: a12b3c4d INBOUND ESP AUTHENTICATOR: OUTBOUND ESP SPI: 0 OUTBOUND ESP CIPHER: OUTBOUND ESP AUTHENTICATOR:
SET SESSION-KEY {INBOUND | OUTBOUND} ESP <spi_number> CIPHER <hex_key> [AUTHENTICATOR <hex_key>]	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-MANUAL(marymap1 2)# **set session-key inbound esp 1234 cipher a12b3c4d authenticator 5f2d33ab** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-MANUAL(marymap1 2)# **display session** INBOUND AH SPI : 0 INBOUND AH KEY : OUTBOUND AH SPI : 0 OUTBOUND AH KEY : INBOUND ESP SPI : 1234 INBOUND ESP CIPHER: a12b3c4d INBOUND ESP AUTHENTICATOR: 5f2d33ab OUTBOUND ESP SPI: 0 OUTBOUND ESP CIPHER: OUTBOUND ESP AUTHENTICATOR:

Syntax	*Example*
SET SESSION-KEY {INBOUND | OUTBOUND} ESP <spi_number> CIPHER <hex_key> [AUTHENTICATOR <hex_key>]	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-MANUAL(marymap1 2)# **set session-key outbound esp 4321 cipher d4c3b21a** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-MANUAL(marymap1 2)# **display session** INBOUND AH SPI : 0 INBOUND AH KEY : OUTBOUND AH SPI : 0 OUTBOUND AH KEY : INBOUND ESP SPI : 1234 INBOUND ESP CIPHER: a12b3c4d INBOUND ESP AUTHENTICATOR: 5f2d33ab OUTBOUND ESP SPI: 4321 OUTBOUND ESP CIPHER: d4c3b21a OUTBOUND ESP AUTHENTICATOR:
SET SESSION-KEY {INBOUND | OUTBOUND} ESP <spi_number> CIPHER <hex_key> [AUTHENTICATOR <hex_key>]	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-MANUAL(marymap1 2)# **set session-key outbound esp 4321 cipher d4c3b21a authenticator ba33d2f5** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-MANUAL(marymap1 2)# **display session** INBOUND AH SPI : 0 INBOUND AH KEY : OUTBOUND AH SPI : 0 OUTBOUND AH KEY : INBOUND ESP SPI : 1234 INBOUND ESP CIPHER: a12b3c4d INBOUND ESP AUTHENTICATOR: 5f2d33ab OUTBOUND ESP SPI: 4321 OUTBOUND ESP CIPHER: d4c3b21a OUTBOUND ESP AUTHENTICATOR: ba33d2f5
SET TRANSFORM-SET <name>	IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-MANUAL(marymap1 2)# **set transform-set set1** IPNOS : CONFIG VPN(50) VR(2.2.2.2) MAP IPSEC-MANUAL(marymap1 2)# **display transform** TRANSFORM-SET : set1

Default Value

None

Command Level

ipsec-manual

Related Commands

ipsec-isakmp set

CHAPTER

18 Firewall Commands

Use these commands to configure firewall proxies. This chapter describes the following commands.

Command	*Page*
access	18-5
action	18-5
destination	18-6
display	18-7
policy	18-7
port	18-8
source	18-8
aol-proxy	18-10
cs-proxy	18-11
event-log	18-12
display	18-12
level	18-13
ftp-proxy	18-15
child-limit	18-16
display	18-16
listen	18-17
policy	18-18
display	18-19
dynamic-pkt-filter	18-21
function	18-22

Command	Page
handoff	18-25
logged	18-26
timeout	18-29
http-proxy	18-31
child-limit	18-31
display	18-33
listen	18-34
policy	18-35
display	18-36
feature	18-37
function	18-38
handoff	18-40
header	18-41
logged	18-43
ourname	18-45
timeout	18-46
ldap-proxy	18-47
lnotes-proxy	18-48
netm-proxy	18-49
nntp-proxy	18-50
plug-proxy	18-51
child-limit	18-52
display	18-53
listen	18-54
policy	18-54
display	18-55
remote-port	18-56
remote-server	18-57
timeout	18-57
pop3-proxy	18-59
child-limit	18-59
display	18-60
listen	18-61

Command	Page
policy	18-63
display	18-63
pop3-server	18-64
timeout	18-65
rsh-proxy	18-66
child-limit	18-67
display	18-67
listen	18-68
policy	18-69
display	18-70
remote-server	18-71
timeout	18-71
smtp-proxy	18-73
child-limit	18-73
display	18-74
listen	18-75
policy	18-76
display	18-78
local-domain	18-78
log-relay	18-79
log-spam	18-80
logging	18-80
mail-hub	18-81
max-bytes	18-81
max-recip	18-82
mime-check	18-82
relay-host	18-83
relay-reply	18-84
spam-host	18-84
spam-reply	18-85
timeout	18-85
sql-proxy	18-87
child-limit	18-87

Command	Page
display	18-88
listen	18-89
policy	18-90
display	18-91
remote-port	18-92
remote-server	18-92
timeout	18-93
telnet-proxy	18-95
child-limit	18-96
display	18-96
listen	18-97
policy	18-98
display	18-99
timeout	18-100

access

Go to the firewall access policy command level. The access policy is similar to an access control list (ACL) policy. The following commands are available at the access level:

- "action" on page 18-5
- "destination" on page 18-6
- "display" on page 18-7
- "policy" on page 18-7
- "port" on page 18-8
- "source" on page 18-9

Syntax	*Example*

`FW ACCESS`	`IPNOS : CONFIG VPN(1) VR(10.1.1.1)# fw access 1`
	`VR(10.1.1.1): Object successfully set`
	
	`IPNOS : CONFIG VPN(1) VR(10.1.1.1) FW ACCESS(1)#`

Default Value

None

Command Level

fw

Related Commands

None

action

Permit or deny a packet's access through a firewall.

Syntax	*Example*
ACTION {PERMIT | DENY}	IPNOS : CONFIG VPN(1) VR(10.1.1.1) FW ACCESS(1)# **action permit** IPNOS : CONFIG VPN(1) VR(10.1.1.1) FW ACCESS(1)# **display action** ACTION : PERMIT
ACTION {PERMIT | DENY}	IPNOS : CONFIG VPN(1) VR(10.1.1.1) FW ACCESS(1)# **action deny** IPNOS : CONFIG VPN(1) VR(10.1.1.1) FW ACCESS(1)# **display action** ACTION : DENY

Default Value

deny

Command Level

access

Related Commands

None

destination

Set the destination server IP address and subnet mask.

Syntax	*Example*
DESTINATION <address> <netmask>	IPNOS : CONFIG VPN(1) VR(10.1.1.1) FW ACCESS(1)# **destination 192.172.16.1 255.255.255.0** IPNOS : CONFIG VPN(1) VR(10.1.1.1) FW ACCESS(1)# **display destination** DESTINATION : 192.172.16.1|255.255.255.0

Default Value

0.0.0.0 | 0.0.0.0

Command Level

access

Related Command

access source

display

View the current firewall configuration.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG VPN(1) VR(10.1.1.1) FW ACCESS(1)# **display** VPN : 1 VR : 10.1.1.1 PRIORITY : 1 SOURCE : 192.168.18.1|255.255.255.0 ACTION : PERMIT DESTINATION : 192.172.16.1|255.255.255.0 PORT : 21 POLICY : fwpolicy1
DISPLAY <argument>	IPNOS : CONFIG VPN(1) VR(10.1.1.1) FW ACCESS(1)# **display source** SOURCE : 192.168.18.1|255.255.255.0

Default Value

None

Command Level

access

Related Commands

None

policy

Set access policy name. This name is referenced by the proxy server of the appropriate type.

Syntax	*Example*
POLICY <name>	IPNOS : CONFIG VPN(1) VR(10.1.1.1) FW ACCESS(1)# **policy fwpolicy1** IPNOS : CONFIG VPN(1) VR(10.1.1.1) FW ACCESS(1)# **display policy** POLICY : fwpolicy1

Default Value

None

Command Level

access

Related Commands

None

port

Set the port number the source is trying to connect to.

Syntax	*Example*
PORT <number>	IPNOS : CONFIG VPN(1) VR(10.1.1.1) FW ACCESS(1)# **port 21** IPNOS : CONFIG VPN(1) VR(10.1.1.1) FW ACCESS(1)# **display port** PORT : 21

Default Value

The industry standard port value for the proxy (for example, 21 for FTP)

Command Level

access

Related Commands

None

source

Set the source server IP address and subnet mask.

Syntax	*Example*
SOURCE <address> <netmask>	IPNOS : CONFIG VPN(1) VR(10.1.1.1) FW ACCESS(1)# **source 192.168.18.1 255.255.255.0** IPNOS : CONFIG VPN(1) VR(10.1.1.1) FW ACCESS(1)# **display source** SOURCE : 192.168.18.1|255.255.255.0

Default Value

0.0.0.0|0.0.0.0

Command Level

access

Related Command

access destination

aol-proxy

Go to the America Online (AOL) proxy command level. See "plug-proxy" on page 18-51 for a list of commands.

Syntax	*Example*

`FW AOL-PROXY`	`IPNOS : CONFIG VPN(1) VR(10.1.1.1)# fw aol-proxy`
	`VR(10.1.1.1): Object successfully set`
	
	`IPNOS : CONFIG VPN(1) VR(10.1.1.1) AOL-PROXY#`

Default Value

None

Command Level

fw

Related Commands

cs-proxy
ftp-proxy
http-proxy
ldap-proxy
lnotes-proxy
netm-proxy
nntp-proxy
plug-proxy
pop3-proxy
rsh-proxy
smtp-proxy
sql-proxy
telnet-proxy

cs-proxy

Go to the CompuServe proxy command level. See "plug-proxy" on page 18-51 for a list of commands.

Syntax	*Example*

`FW CS-PROXY`	`IPNOS : CONFIG VPN(1) VR(10.1.1.1)# fw cs-proxy`
	`VR(10.1.1.1): Object successfully set`
	
	`IPNOS : CONFIG VPN(1) VR(10.1.1.1) CS-PROXY#`

Default Value

None

Command Level

fw

Related Commands

aol-proxy
ftp-proxy
http-proxy
ldap-proxy
lnotes-proxy
netm-proxy
nntp-proxy
plug-proxy
pop3-proxy
rsh-proxy
smtp-proxy
sql-proxy
telnet-proxy

event-log

Go to the event-log command level. The following commands are available.

- display
- level

Syntax	*Example*
FW EVENT-LOG	IPNOS : CONFIG VPN(50) VR(192.128.32.1)# **fw event-log** VR(192.128.32.1): Object successfully set IPNOS : CONFIG VPN(50) VR(192.128.32.1) FW EVENT-LOG#
NO FW EVENT-LOG	IPNOS : CONFIG VPN(50) VR(192.128.32.1)# **no fw event-log** VR(192.128.32.1): Object successfully set VR: FW Event log disabled

Default Value

None

Command Level

VR

Related Command

syslogd

display

View the current event-log configuration.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG VPN(50) VR(192.128.32.1) FW EVENT-LOG# **display** VPN : 50 VR : 192.128.32.1 LEVEL : MAJOR

Default Value

None

Command Level

Firewall event-log

Related Commands

None

level

Display event messages by severity level. As you choose a lower level, the higher level events will also be displayed, which will slow the IPSX switch's performance. Use the **no** command to reset the command to its default value. Use the **no fw event-log** command to disable displaying all event messages. Table 18-1 describes the severity levels.

Table 18-1 IPSec Event Message Severity Levels

Level	*Description*
Critical	Critical system problems that affect normal IPSX operations, such as system failure. This is the highest level. Only critical level messages will be displayed.
Major	Major problems, such as proxy failure. This is the default level.
Minor	Minor errors, such as an invalid parameter.
Inform	General information, such as a new proxy is enabled.
Debug	Debug codes, such as programming messages. This is the lowest level. All levels of messages will be displayed. Note: IPSX switch system performance will drop dramatically because of all the log messages it has to display at this level.

Syntax	*Example*

`LEVEL { CRITICAL | MAJOR | MINOR | INFORM | DEBUG }`	`IPNOS : CONFIG VPN(50) VR(192.128.32.1) FW EVENT-LOG#` **`level critical`**
	
	`IPNOS : CONFIG VPN(50) VR(192.128.32.1) FW EVENT-LOG#` **`display`**
	`VPN : 50`
	`VR : 192.128.32.1`
	`LEVEL : CRITICAL`

Syntax	*Example*
LEVEL { CRITICAL | MAJOR | MINOR | INFORM | DEBUG }	IPNOS : CONFIG VPN(50) VR(192.128.32.1) FW EVENT-LOG# **no level** IPNOS : CONFIG VPN(50) VR(192.128.32.1) FW EVENT-LOG# **display** VPN : 50 VR : 192.128.32.1 LEVEL : MAJOR

Default Value

Major

Command Level

Firewall event-log

Related Commands

None

ftp-proxy

Go to the File Transfer Protocol (FTP) proxy command level.. The following commands are available.

- "child-limit" on page 18-16
- "display" on page 18-16
- "listen" on page 18-17
- "policy" on page 18-18

Syntax	*Example*

`FW FTP-PROXY`	`IPNOS : CONFIG VPN(1) VR(10.1.1.1)# fw ftp-proxy`
	`VR(10.1.1.1): Object successfully set`
	
	`IPNOS : CONFIG VPN(1) VR(10.1.1.1) FTP-PROXY#`

Default Value

None

Command Level

fw

Related Commands

aol-proxy
cs-proxy
http-proxy
ldap-proxy
lnotes-proxy
netm-proxy
nntp-proxy
plug-proxy
pop3-proxy
rsh-proxy
smtp-proxy
sql-proxy
telnet-proxy

child-limit

Set the maximum concurrent FTP sessions supported by the proxy. Use the **no** option to reset the command to its default value.

Syntax	*Example*
CHILD-LIMIT <value>	IPNOS: CONFIG VPN(50) VR(1.1.1.1) FTP-PROXY# **child-limit 32** IPNOS: CONFIG VPN(50) VR(1.1.1.1) FTP-PROXY# **display child** CHILD-LIMIT : 32
NO CHILD-LIMIT	IPNOS: CONFIG VPN(50) VR(1.1.1.1) FTP-PROXY# **no child-limit** IPNOS: CONFIG VPN(50) VR(1.1.1.1) FTP-PROXY# **display child-limit** CHILD-LIMIT : 64

Default Value

64 sessions

Command Level

ftp-proxy

Related Commands

http-proxy child-limit
plug-proxy child-limit
pop3-proxy child-limit
rsh-proxy child-limit
smtp-proxy child-limit
sql-proxy child-limit
telnet-proxy child-limit

display

View the current status of the FTP proxy.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY# **display** VPN : 50 VR : 10.0.16.1 PROXY : FTP-PROXY STATUS : ENABLE CHILD-LIMIT : 64 LISTEN(s): List is empty POLICY(s): List is empty
DISPLAY <argument>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY# **display proxy** PROXY : FTP-PROXY

Default Value

None

Command Level

ftp-proxy

Related Commands

None

listen

Set the port that the proxy will listen to. Use the value 21 for FTP proxy, for example. Use the **no** option to delete a port.

Syntax	*Example*
[NO] LISTEN <port>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY# **listen 21** LISTEN 21: Object successfully set IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY# **display** VPN : 50 VR : 10.0.16.1 PROXY : FTP-PROXY STATUS : DISABLE CHILD-LIMIT : 64 LISTEN(s): 21 POLICY(s): List is empty

Default Value

None

Command Level

ftp-proxy

Related Commands

http-proxy listen
plug-proxy listen
pop3-proxy listen
rsh-proxy listen
smtp-proxy listen
sql-proxy listen
telnet-proxy listen

policy

Set the name of the proxy policy. The policy is an index to a particular group of proxy settings. The policy name is alphanumeric and can include "_" but no blank spaces. This command also puts you into the FTP policy command level. Use the **no** option to delete the policy. The following commands are available at this level:

- "display" on page 18-19

- "dynamic-pkt-filter" on page 18-21
- "function" on page 18-22
- "handoff" on page 18-25
- "logged" on page 18-26
- "timeout" on page 18-29

Syntax	*Example*

`[NO] POLICY <name>`	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY# policy 1`
	`FTP-PROXY: Object successfully set`
	
	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)#`

Default Value

None

Command Level

ftp-proxy

Related Commands

http-proxy policy
plug-proxy policy
pop3-proxy policy
rsh-proxy policy
smtp-proxy policy
sql-proxy policy
telnet-proxy policy

display

View the current FTP policy configuration.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)# **display** VPN : 50 VR : 10.0.16.1 POLICY : 1 TIMEOUT : 0 HANDOFF : DYNAMIC-PKT-FILTER: OFF LOGGED(s): !ABOR !ACCT !ALLO !APPE !CDUP !CWD !DELE !HELP !LIST !MKD !MODE !NLST !NOOP !PASS !PASV !PORT !PWD !QUIT !REIN !REST !RETR !RMD !RNFR !RNTO !SITE !SIZE !SMNT !STAT !STOR !STOU !STRU !SYST !TYPE !USER !XCUP !XCWD !XMKD !XPWD !XRMD FUNC(s): ABOR ACCT ALLO APPE CDUP CWD DELE HELP LIST MKD MODE NLST NOOP PASS PASV PORT PWD QUIT REIN REST RETR RMD RNFR RNTO SITE SIZE SMNT STAT STOR STOU STRU SYST TYPE USER XCUP XCWD XMKD XPWD XRMD

Default Value

None

Command Level

ftp-policy

Related Command

None

dynamic-pkt-filter

Turn on or off dynamic packet filtering.

Syntax	*Example*
DYNAMIC-PKT-FILTER {ON | OFF}	IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)# dynamic-pkt-filter on IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)# dis VPN : 50 VR : 10.0.16.1 POLICY : 1 TIMEOUT : 0 HANDOFF : DYNAMIC-PKT-FILTER: ON LOGGED(s): !ABOR !ACCT !ALLO !APPE !CDUP !CWD !DELE !HELP !LIST !MKD !MODE !NLST !NOOP !PASS !PASV !PORT !PWD !QUIT !REIN !REST !RETR !RMD !RNFR !RNTO !SITE !SIZE !SMNT !STAT !STOR !STOU !STRU !SYST !TYPE !USER !XCUP !XCWD !XMKD !XPWD !XRMD FUNC(s): ABOR ACCT ALLO APPE CDUP CWD DELE HELP LIST MKD MODE NLST NOOP PASS PASV PORT PWD QUIT REIN REST RETR RMD RNFR RNTO SITE SIZE SMNT STAT STOR STOU STRU SYST TYPE USER XCUP XCWD XMKD XPWD XRMD

Default Value

Off

Command Level

ftp-policy

Related Command

None

function

Permit or deny various FTP functions. By default, all FTP functions are allowed. Table 18-2 lists FTP functions that can be controlled. Use the **no** option to deny a function. Run the command **no func all** to disable the default setting. Then you can enable individual functions.

Table 18-2 FTP Functions

Function	*Description*
ABOR	Abort
ACCT	Account
ALLO	Allocate
APPE	Append
CDUP	Change to parent directory
CWD	Change working directory
DELE	Delete
HELP	Help
LIST	List
MKD	Make directory
MODE	Transfer mode
NLST	Name list
NOOP	No operation
PASS	Password
PASV	Passive
PORT	Data port
PWD	Print directory
QUIT	Logout
REIN	Full user terminate
REST	Restart an incomplete transfer

Table 18-2 FTP Functions (Continued)

Function	*Description*
RETR	Retrieve
RMD	Remove directory
RNFR	Rename from
RNTO	Rename to
SITE	Non-standard commands
SIZE	Return the size of the file
SMNT	Structure mount
STAT	Return the status of the server
STOR	Store
STOU	Store unique
STRU	File structure
SYST	Show operating system server type
TYPE	Representation type
USER	User name
XCUP	Change to parent directory
XCWD	Change working directory
XMKD	Make directory
XPWD	Print working directory
XRMD	Remove working directory

Syntax	*Example*
[NO] FUNC ALL	IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)# **func all** IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)# **display func** FUNC(s): ABOR ACCT ALLO APPE CDUP CWD DELE HELP LIST MKD MODE NLST NOOP PASS PASV PORT PWD QUIT REIN REST RETR RMD RNFR RNTO SITE SIZE SMNT STAT STOR STOU STRU SYST TYPE USER XCUP XCWD XMKD XPWD XRMD
[NO] FUNC ALL	IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)# **no func all** IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)# **display func** FUNC(s): !ABOR !ACCT !ALLO !APPE !CDUP !CWD !DELE !HELP !LIST !MKD !MODE !NLST !NOOP !PASS !PASV !PORT !PWD !QUIT !REIN !REST !RETR !RMD !RNFR !RNTO !SITE !SIZE !SMNT !STAT !STOR !STOU !STRU !SYST !TYPE !USER !XCUP !XCWD !XMKD !XPWD !XRMD

Syntax	Example
FUNC <function>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)# **func abor** IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)# **display func** FUNC(s): ABOR !ACCT !ALLO !APPE !CDUP !CWD !DELE !HELP !LIST !MKD !MODE !NLST !NOOP !PASS !PASV !PORT !PWD !QUIT !REIN !REST !RETR !RMD !RNFR !RNTO !SITE !SIZE !SMNT !STAT !STOR !STOU !STRU !SYST !TYPE !USER !XCUP !XCWD !XMKD !XPWD !XRMD

Default Value

All FTP functions are enabled.

Command Level

ftp-policy

Related Commands

http-policy function

handoff

Set the name of the host to which the ftp-proxy hands the proxy request. The value specified here can be an IP address, a host name, or a network entity name resolved by a DNS server. You can also bind the host to a specific port. If no bind port is specified, the proxy uses port 21 by default.

Syntax	Example
HANDOFF <host>[:<port>]	IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)# **handoff 128.158.36.2** IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)# **display handoff** HANDOFF : 128.158.36.2

Syntax	*Example*
HANDOFF <host>[:<port>]	IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)# **handoff 128.158.36.2:21** IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)# **display handoff** HANDOFF : 128.158.36.2:21

Default Value

None

Command Level

ftp-policy

Related Command

None

logged

Log a specific FTP transaction to the Syslogd server. See "syslogd" on page 16-151. Disabled log functions are indicated by an exclamation mark (!) before the item. Run the command **logged all** to log all transactions. You can run the command **no logged all** to disable logging all transactions. Table 18-3 lists the FTP transactions that can be logged.

Table 18-3 FTP Transactions

Transaction	*Description*
ABOR	Abort
ACCT	Account
ALLO	Allocate
APPE	Append
CDUP	Change to parent directory
CWD	Change working directory
DELE	Delete
HELP	Help
LIST	List
MKD	Make directory

Table 18-3 FTP Transactions (Continued)

Transaction	*Description*
MODE	Transfer mode
NLST	Name list
NOOP	No operation
PASS	Password
PASV	Passive
PORT	Data port
PWD	Print directory
QUIT	Logout
REIN	Full user terminate
REST	Restart an incomplete transfer
RETR	Retrieve
RMD	Remove directory
RNFR	Rename from
RNTO	Rename to
SITE	Non-standard commands
SIZE	Return the size of the file
SMNT	Structure mount
STAT	Return the status of the server
STOR	Store
STOU	Store unique
STRU	File structure
SYST	Show operating system server type
TYPE	Representation type
USER	User name
XCUP	Change to parent directory
XCWD	Change working directory
XMKD	Make directory
XPWD	Print working directory
XRMD	Remove working directory

Syntax	*Example*
[NO] LOGGED ALL	IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)# **no logged all** IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)# **display log** LOGGED(s): !ABOR !ACCT !ALLO !APPE !CDUP !CWD !DELE !HELP !LIST !MKD !MODE !NLST !NOOP !PASS !PASV !PORT !PWD !QUIT !REIN !REST !RETR !RMD !RNFR !RNTO !SITE !SIZE !SMNT !STAT !STOR !STOU !STRU !SYST !TYPE !USER !XCUP !XCWD !XMKD !XPWD !XRMD
[NO] LOGGED ALL	IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)# **logged all** IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)# **display log** LOGGED(s): ABOR ACCT ALLO APPE CDUP CWD DELE HELP LIST MKD MODE NLST NOOP PASS PASV PORT PWD QUIT REIN REST RETR RMD RNFR RNTO SITE SIZE SMNT STAT STOR STOU STRU SYST TYPE USER XCUP XCWD XMKD XPWD XRMD

Syntax	Example
LOGGED <function>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)# **logged abor** IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)# **display log** LOGGED(s): ABOR !ACCT !ALLO !APPE !CDUP !CWD !DELE !HELP !LIST !MKD !MODE !NLST !NOOP !PASS !PASV !PORT !PWD !QUIT !REIN !REST !RETR !RMD !RNFR !RNTO !SITE !SIZE !SMNT !STAT !STOR !STOU !STRU !SYST !TYPE !USER !XCUP !XCWD !XMKD !XPWD !XRMD

Default Value

No FTP transactions are logged.

Command Level

ftp-policy

Related Commands

logged

timeout

Set idle connection timeout.

Syntax	Example
TIMEOUT <value>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)# **timeout 2400** IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)# **display timeout** TIMEOUT : 2400

Syntax	*Example*
NO TIMEOUT	IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)# **no timeout** IPNOS : CONFIG VPN(50) VR(10.0.16.1) FTP-PROXY POLICY(1)# **display timeout** TIMEOUT : 7200

Default Value

7200 seconds (2 hours)

Command Level

ftp-policy

Related Commands

timeout

http-proxy

Go to Hypertext Transfer Protocol (HTTP) proxy command level. The following commands are available.

- "child-limit" on page 18-31
- "display" on page 18-33
- "listen" on page 18-34
- "policy" on page 18-35

Syntax	*Example*

`FW HTTP-PROXY`	`IPNOS : CONFIG VPN(1) VR(10.1.1.1)# fw http-proxy`
	`VR(10.1.1.1): Object successfully set`
	
	`IPNOS : CONFIG VPN(1) VR(10.1.1.1) HTTP-PROXY#`

Default Value

None

Command Level

fw

Related Commands

aol-proxy
cs-proxy
ftp-proxy
ldap-proxy
lnotes-proxy
netm-proxy
nntp-proxy
plug-proxy
pop3-proxy
rsh-proxy
smtp-proxy
sql-proxy
telnet-proxy

child-limit

Set the maximum number of concurrent HTTP sessions.

Syntax	*Example*
CHILD-LIMIT <value>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY# **child-limit 100** IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY# **display child-limit** CHILD-LIMIT : 100 LISTEN(s): List is empty POLICY(s): List is empty
NO CHILD-LIMIT	IPNOS : CONFIG VPN(1) VR(10.1.1.1) HTTP-PROXY# **no child-limit** IPNOS : CONFIG VPN(1) VR(10.1.1.1) HTTP-PROXY# **display** VPN : 1 VR : 10.1.1.1 PROXY : HTTP-PROXY STATUS : ENABLE CHILD-LIMIT : 64 LISTEN(s): 80 POLICY(s): List is empty

Default Value

64 sessions

Command Level

http-proxy

Related Commands

ftp-proxy child-limit
plug-proxy child-limit
pop3-proxy child-limit
rsh-proxy child-limit
smtp-proxy child-limit
sql-proxy child-limit
telnet-proxy child-limit

display

View the current HTTP proxy configuration.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY# **display** VPN : 50 VR : 10.0.16.1 PROXY : HTTP-PROXY STATUS : ENABLE CHILD-LIMIT : 64 LISTEN(s): List is empty POLICY(s): List is empty IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY#
DISPLAY <argument>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY# **display proxy** PROXY : HTTP-PROXY STATUS : ENABLE LISTEN(s): List is empty POLICY(s): List is empty IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY#

Default Value

None

Command Level

http-proxy

Related Commands

None

listen

Set the port the proxy will listen to.

Syntax	*Example*
[NO] LISTEN <port>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY# **listen 8080** LISTEN 80: Object successfully set IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY# **display listen** No match LISTEN(s): 80 8080 POLICY(s): List is empty
[NO] LISTEN <port>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY# **no listen 8080** LISTEN 8080: Object successfully deleted IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY# **display listen** No match LISTEN(s): 80 POLICY(s): List is empty

Default Value

Ports 80 and 8080

Command Level

http-proxy

Related Commands

ftp-proxy listen
plug-proxy listen
pop3-proxy listen
rsh-proxy listen
smtp-proxy listen
sql-proxy listen
telnet-proxy listen

policy

Set the name of the HTTP proxy policy. The policy is an index to a particular group of proxy settings. The policy name is alphanumeric and can include "_" but no blank spaces. This command also puts you into the FTP policy command level. Use the **no** option to delete a policy. The following commands are available at this level:

- "display" on page 18-36
- "feature" on page 18-37
- "function" on page 18-38
- "handoff" on page 18-40
- "header" on page 18-41
- "logged" on page 18-43
- "ourname" on page 18-45
- "timeout" on page 18-46

Syntax	*Example*

`[NO] POLICY <name>`	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY# policy 1`
	`HTTP-PROXY: Object successfully set`
	
	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)#`

Default Value

None

Command Level

http-proxy

Related Command

ftp-proxy policy
plug-proxy policy
pop3-proxy policy
rsh-proxy policy
smtp-proxy policy
sql-proxy policy
telnet-proxy policy

display

View the current HTTP policy configuration.

Syntax	*Example*

DISPLAY	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)#` **`display`**
	`VPN : 50`
	`VR : 10.0.16.1`
	`POLICY : 1`
	`TIMEOUT : 0`
	`HANDOFF :`
	`OURNAME :`
	
	`LOGGED(s):`
	`!BIN !DIR !EXEC !FTP`
	`!HREQ !HTTP !READ !RETR`
	`!STOR !TEXT !TYPE !WRTE`
	
	`FUNC(s):`
	`BIN DIR EXEC FTP`
	`HREQ HTTP READ RETR`
	`STOR TEXT TYPE WRTE`
	
	`FEATURE(s):`
	`ACTX FRAM HTM2 JAVA`
	`KANJ LEGK MPF SCRP`
	
	`HEADER(s):`
	`ACHR AENC ALNG ALOW`
	`CENC CLNG DATE EXP`
	`FROM IMOD LMOD LNK`
	`MIME PRGM REFR RETA`
	`SERV TITL URI USER`
	`WWW`

Default Value

None

Command Level

http-policy

Related Command

None

feature

Permit or deny web extensions or executable languages. Run the command **no feature all** to disable the default setting. Then you can enable individual features. Use the **no** option to disable a feature. Table 18-4 describes the features available.

Table 18-4 HTTP-Proxy Features

Feature	*Description*
ACTX	Microsoft ActiveX programming language
FRAM	Hyper Text Markup Language (HTML) Frames
HTM2	Enforce HTML 2.0 tags.
JAVA	SUN Microsystem's Java programming language
KANJ	Japanese Kanji language
LEGK	Legacy Kanji
MPF	Multi-Part Forms to be used with a CGI language, such as Perl
SCRP	Netscape's Java Script scripting language

Syntax	*Example*

[NO] FEATURE ALL	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)#` **`feature all`**
	
	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)#` **`display feature`**
	
	`FEATURE(s):`
	`ACTX FRAM HTM2 JAVA`
	`KANJ LEGK MPF SCRP`
	
	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)#`

Syntax	*Example*
[NO] FEATURE ALL	IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **no feature all** IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **display feature** FEATURE(s): !ACTX !FRAM !HTM2 !JAVA !KANJ !LEGK !MPF !SCRP
[NO] FEATURE [ACTX][FRAM][HTM2][JAVA][KANJ][LEGK][MPF][SCRP]	IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **feature fram** IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **display feature** FEATURE(s): !ACTX FRAM !HTM2 !JAVA !KANJ !LEGK !MPF !SCRP

Default Value

All HTTP features are enabled.

Command Level

http-policy

Related Commands

None

function

Permit or deny various HTTP functions. Run the command **no function all** to disable the default setting. Then you can enable individual functions. Use the **no** option to disable a function. Table 18-5 lists the HTTP functions that can be controlled.

Table 18-5 HTTP Functions

Function	*Description*
BIN	Binary
DIR	Directory

Table 18-5 HTTP Functions (Continued)

Function	*Description*
EXEC	Execute
FTP	File Transfer Protocol
HREQ	HTTP requests
HTTP	Hypertext Transfer Protocol
READ	Read
RETR	Retrieve
STOR	Store
TEXT	Text
TYPE	Type
WRTE	Write

Syntax	*Example*
[NO] FUNC ALL	IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **func all** IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **display func** FUNC(s): BIN DIR EXEC FTP HREQ HTTP READ RETR STOR TEXT TYPE WRTE
[NO] FUNC ALL	IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **no func all** IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **display func** FUNC(s): !BIN !DIR !EXEC !FTP !HREQ !HTTP !READ !RETR !STOR !TEXT !TYPE !WRTE

Syntax	*Example*
FUNC <function>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **func dir** IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **display func** FUNC(s): !BIN DIR !EXEC !FTP !HREQ !HTTP !READ !RETR !STOR !TEXT !TYPE !WRTE

Default Value

All HTTP functions are enabled.

Command Level

http-policy

Related Commands

None

handoff

Set the name of the host the policy hands the proxy request to. The host can be specified through a host name, or an IP address. You can also bind the host to a specific port.

Syntax	*Example*
HANDOFF <host>[:<port>]	IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **handoff http.server.com** IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **display handoff** HANDOFF : http.server.com

Default Value

None

Command Level

http-policy

Related Commands

None

header

Permit or deny various HTTP packet header types. This enables the proxy administrator to ensure that certain types of add-ons or plug-ins can't be used. Run the command **no header all** to disable the default setting. Then you can enable individual functions. Use the **no** option to disable a packet header. Table 18-6 describes the headers available.

Table 18-6 HTTP Packet Headers

Header	*Description*
ACHR	Accept character set
AENC	Accept encoding
ALNG	Accept language
ALOW	Allow
CENC	Content encoding
CLNG	Content language
DATE	Date
EXP	Expires
FROM	From
IMOD	If modified since
LMOD	Last modified
LNK	Link
MIME	Multipurpose internet mail extension (MIME) version
PRGM	Pragma
REFR	Referer
RETA	Retry after
SERV	Server
TITL	Title
URL	Uniform resource locator

Table 18-6 HTTP Packet Headers (Continued)

Header	*Description*
USER	User agent
WWW	World Wide Web authentication

Syntax	*Example*
[NO] HEADER ALL	IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **header all** IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **display header** HEADER(s): ACHR AENC ALNG ALOW CENC CLNG DATE EXP FROM IMOD LMOD LNK MIME PRGM REFR RETA SERV TITL URI USER WWW
[NO] HEADER ALL	IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **no header all** IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **display header** HEADER(s): !ACHR !AENC !ALNG !ALOW !CENC !CLNG !DATE !EXP !FROM !IMOD !LMOD !LNK !MIME !PRGM !REFR !RETA !SERV !TITL !URI !USER !WWW

Syntax	*Example*
HEADER <header>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **header www** IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **display header** HEADER(s): !ACHR !AENC !ALNG !ALOW !CENC !CLNG !DATE !EXP !FROM !IMOD !LMOD !LNK !MIME !PRGM !REFR !RETA !SERV !TITL !URI !USER WWW

Default Value

All packet header types are permitted.

Command Level

http-policy

Related Commands

None

logged

Log HTTP transactions to syslogd file. See "syslogd" on page 16-151. Run the command **logged all** to log all transactions. You can run the command **no logged all** to disable logging all transactions. Use the **no** option to disable logging a transaction. Table 18-7 lists the HTTP transactions that can be logged.

Table 18-7 HTTP Transactions

Transaction	*Description*
BIN	Binary
DIR	Directory
EXEC	Execute
FTP	File Transfer Protocol
HREQ	HTTP requests

Table 18-7 HTTP Transactions (Continued)

Transaction	*Description*
HTTP	HTTP
READ	Read
RETR	Retrieve
STOR	Store
TEXT	Text
TYPE	Type
WRTE	Write

Syntax	*Example*
[NO] LOGGED ALL	IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **logged all** IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **display log** LOGGED(s): BIN DIR EXEC FTP HREQ HTTP READ RETR STOR TEXT TYPE WRTE
[NO] LOGGED ALL	IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **no logged all** IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **display log** LOGGED(s): !BIN !DIR !EXEC !FTP !HREQ !HTTP !READ !RETR !STOR !TEXT !TYPE !WRTE

Syntax	Example
LOGGED <function>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **logged dir** IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **display log** LOGGED(s): !BIN DIR !EXEC !FTP !HREQ !HTTP !READ !RETR !STOR !TEXT !TYPE !WRTE

Default Value

No HTTP transactions are logged.

Command Level

http-policy

Related Commands

logged

ourname

Set the host and domain name that the HTTP proxy uses when putting its own name into non-transparent forwarded URLs.

Syntax	Example
OURNAME <name>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **ourname www.cosinecom.com** IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **display ourname** OURNAME : www.cosinecom.com

Default Value

None

Command Level

http-policy

Related Commands

None

timeout

Set the timeout limit for an idle connection.

Syntax	*Example*
TIMEOUT <value>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **timeout 4500** IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **display timeout** TIMEOUT : 4500
NO TIMEOUT	IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **no timeout** IPNOS : CONFIG VPN(50) VR(10.0.16.1) HTTP-PROXY POLICY(1)# **display timeout** TIMEOUT : 7200

Default Value

7200 seconds (2 hours)

Command Level

http-policy

Related Command

timeout

ldap-proxy

Go to the lightweight directory access protocol (LDAP) proxy command level. See "plug-proxy" on page 18-51 for a list of commands.

Syntax	*Example*

`FW LDAP-PROXY`	`IPNOS : CONFIG VPN(1) VR(10.1.1.1)# fw ldap-proxy`
	`VR(10.1.1.1): Object successfully set`
	
	`IPNOS : CONFIG VPN(1) VR(10.1.1.1) LDAP-PROXY#`

Default Value

None

Command Level

fw

Related Commands

aol-proxy
cs-proxy
ftp-proxy
http-proxy
lnotes-proxy
netm-proxy
nntp-proxy
plug-proxy
pop3-proxy
rsh-proxy
smtp-proxy
sql-proxy
telnet-proxy

lnotes-proxy

Go to the Lotus Notes proxy command level. See "plug-proxy" on page 18-51 for a list of commands.

Syntax	*Example*
FW LNOTES-PROXY	IPNOS : CONFIG VPN(1) VR(10.1.1.1)# **fw lnotes-proxy** VR(10.1.1.1): Object successfully set IPNOS : CONFIG VPN(1) VR(10.1.1.1) LNOTES-PROXY#

Default Value

None

Command Level

fw

Related Commands

aol-proxy
cs-proxy
ftp-proxy
http-proxy
ldap-proxy
netm-proxy
nntp-proxy
plug-proxy
pop3-proxy
rsh-proxy
smtp-proxy
sql-proxy
telnet-proxy

netm-proxy

Go to the NetMeeting proxy command level. See "plug-proxy" on page 18-51 for a list of commands.

NOTE: The NetMeeting proxy supports only audio and video. The electronic white-board and shared application features are not supported.

Syntax	*Example*

`FW NETM-PROXY`	`IPNOS : CONFIG VPN(1) VR(10.1.1.1)# fw netm-proxy`
	`VR(10.1.1.1): Object successfully set`
	
	`IPNOS : CONFIG VPN(1) VR(10.1.1.1) NETM-PROXY#`

Default Value

None

Command Level

fw

Related Commands

aol-proxy
cs-proxy
ftp-proxy
http-proxy
ldap-proxy
lnotes-proxy
nntp-proxy
plug-proxy
pop3-proxy
rsh-proxy
smtp-proxy
sql-proxy
telnet-proxy

nntp-proxy

Go to the Network News Transfer Protocol (NNTP) proxy command level. See "plug-proxy" on page 18-51 for a list of commands.

Syntax	*Example*

`FW NNTP-PROXY`	`IPNOS : CONFIG VPN(1) VR(10.1.1.1)# fw nntp-proxy`
	`VR(10.1.1.1): Object successfully set`
	
	`IPNOS : CONFIG VPN(1) VR(10.1.1.1) NNTP-PROXY#`

Default Value

None

Command Level

fw

Related Commands

aol-proxy
cs-proxy
ftp-proxy
http-proxy
ldap-proxy
lnotes-proxy
netm-proxy
plug-proxy
pop3-proxy
rsh-proxy
smtp-proxy
sql-proxy
telnet-proxy

plug-proxy

Go to the plug-proxy command level. These proxy commands apply for plug-ins and third-party network applications. The following commands are available.:

- "child-limit" on page 18-52
- "display" on page 18-53
- "listen" on page 18-54
- "policy" on page 18-54

Several third-party firewall proxies are available that are based on the plug proxy. Their CLI commands are exactly the same as the **plug-proxy**. The primary difference is the port they listen to.

- "aol-proxy" on page 18-10
- "cs-proxy" on page 18-11
- "ldap-proxy" on page 18-47
- "lnotes-proxy" on page 18-48
- "netm-proxy" on page 18-49
- "nntp-proxy" on page 18-50

Syntax	*Example*

`FW PLUG-PROXY`	`IPNOS : CONFIG VPN(1) VR(10.1.1.1)# fw plug-proxy`
	`VR(10.1.1.1): Object successfully set`
	
	`IPNOS : CONFIG VPN(1) VR(10.1.1.1) PLUG-PROXY#`

Default Value

None

Command Level

fw

Related Commands

aol-proxy
cs-proxy
ftp-proxy
http-proxy
ldap-proxy
lnotes-proxy
netm-proxy

nntp-proxy
pop3-proxy
rsh-proxy
smtp-proxy
sql-proxy
telnet-proxy

child-limit

Set the maximum number of concurrent plug sessions supported.

Syntax	*Example*

`CHILD-LIMIT <value>`	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) PLUG-PROXY#`
	`child-limit 64`
	
	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) PLUG-PROXY#`
	`display child-limit`
	`CHILD-LIMIT  : 64`
	
	`LISTEN(s):`
	`  List is empty`
	
	`POLICY(s):`
	`  List is empty`

Default Value

64 sessions

Command Level

plug-proxy

Related Commands

ftp-proxy child-limit
http-proxy child-limit
pop3-proxy child-limit
rsh-proxy child-limit
smtp-proxy child-limit
sql-proxy child-limit
telnet-proxy child-limit

display

View the current plug proxy configuration.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG VPN(50) VR(10.0.16.1) PLUG-PROXY# **display** VPN : 50 VR : 10.0.16.1 PROXY : PLUG-PROXY STATUS : ENABLE CHILD-LIMIT : 64 LISTEN(s): List is empty POLICY(s): List is empty IPNOS : CONFIG VPN(50) VR(10.0.16.1) PLUG-PROXY#
DISPLAY <argument>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) PLUG-PROXY# **display proxy** PROXY : PLUG-PROXY STATUS : ENABLE LISTEN(s): List is empty POLICY(s): List is empty IPNOS : CONFIG VPN(50) VR(10.0.16.1) PLUG-PROXY#

Default Value

None

Command Level

plug-proxy

Related Commands

None

listen

Set the port that the proxy will listen to. Choose from 1 to 65,535. Use the **no** option to delete a port from the list. Refer to the Internet Assigned Numbers Authority (IANA) at http://www.iana.org for the official port information.

Syntax	*Example*
[NO] LISTEN <port>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) PLUG-PROXY# **listen 80** LISTEN 80: Object successfully set IPNOS : CONFIG VPN(50) VR(10.0.16.1) PLUG-PROXY# **display listen** No match LISTEN(s): 80 POLICY(s): List is empty

Default Value

None

Command Level

plug-proxy

Related Commands

ftp-proxy listen
http-proxy listen
pop3-proxy listen
rsh-proxy listen
smtp-proxy listen
sql-proxy listen
telnet-proxy listen

policy

Set the name of the proxy policy. The policy is an index to a particular group of proxy settings. The policy name is alphanumeric and can include "_" but no blank spaces. This command also puts you into the plug policy command

level. Use the **no** option to delete a policy. The following commands are available at this level:

- "display" on page 18-55
- "remote-port" on page 18-56
- "remote-server" on page 18-57
- "timeout" on page 18-57

Syntax	*Example*

`[NO] POLICY <name>`	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) PLUG-PROXY# policy 1`
	`PLUG-PROXY: Object successfully set`
	
	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) PLUG-PROXY POLICY(1)#`

Default Value

None

Command Level

plug-proxy

Related Commands

ftp-proxy policy
http-proxy policy
pop3-proxy policy
rsh-proxy policy
smtp-proxy policy
sql-proxy policy
telnet-proxy policy

display

View the current plug policy configuration.

Syntax	Example
DISPLAY	IPNOS : CONFIG VPN(50) VR(10.0.16.1) plug-PROXY POLICY(1)# **display** VPN : 50 VR : 10.0.16.1 POLICY : 1 TIMEOUT : 0 REMOTE PORT : REMOTE SERVER :

Default Value

None

Command Level

plug-policy

Related Command

None

remote-port

Set the port to connect to on the remote server. Choose from 1 to 65,535. This command is not available for the remote shell (RSH) proxy. See "rsh-proxy" on page 18-66. Use the **no** option to reset the command to its default value. Refer to the Internet Assigned Numbers Authority (IANA) at http://www.iana.org for the official port information.

Syntax	Example
[NO] REMOTE-PORT <port>	IPNOS : CONFIG VPN(1) VR(10.1.1.1) PLUG-PROXY POLICY(1)# **remote-port 352** IPNOS : CONFIG VPN(1) VR(10.1.1.1) PLUG-PROXY POLICY(1)# **display remote** REMOTE-SERVER : 192.168.18.1 REMOTE-PORT : 352

Default Value

Plug is none, AOL is 5190, CompuServe is 4144, NNTP is 119, LotusNotes is 1352, LDAP is 389, NetMeeting is 389.

Command Level

plug-policy

Related Command

None

remote-server

Set the IP address of the remote server to plug into.

Syntax	*Example*
REMOTE-SERVER <host>	IPNOS : CONFIG VPN(1) VR(10.1.1.1) PLUG-PROXY POLICY(1)# **remote-server 192.168.18.1** IPNOS : CONFIG VPN(1) VR(10.1.1.1) PLUG-PROXY POLICY(1)# **display remote** REMOTE-SERVER : 192.168.18.1 REMOTE-PORT : 3600
[NO] REMOTE-SERVER	IPNOS : CONFIG VPN(1) VR(10.1.1.1) PLUG-PROXY POLICY(1)# **no remote-server** IPNOS : CONFIG VPN(1) VR(10.1.1.1) PLUG-PROXY POLICY(1)# **display remote** REMOTE-SERVER : REMOTE-PORT : 3600

Default Value

None

Command Level

plug-policy

Related Command

None

timeout

Set the timeout limit for an idle connection.

Syntax	*Example*
TIMEOUT <value>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) PLUG-PROXY POLICY(1)# **timeout 4500** IPNOS : CONFIG VPN(50) VR(10.0.16.1) PLUG-PROXY POLICY(1)# **display timeout** TIMEOUT : 4500
NO TIMEOUT	IPNOS : CONFIG VPN(50) VR(10.0.16.1) PLUG-PROXY POLICY(1)# **no timeout** IPNOS : CONFIG VPN(50) VR(10.0.16.1) PLUG-PROXY POLICY(1)# **display timeout** TIMEOUT : 7200

Default Value

7200 seconds (2 hours)

Command Level

plug-policy

Related Command

timeout

pop3-proxy

Go to the Post Office Protocol, Version 3 (POP3) proxy command level for incoming electronic mail. The following commands are available.

- "child-limit" on page 18-59
- "display" on page 18-60
- "listen" on page 18-61
- "policy" on page 18-63

Syntax	*Example*

`FW POP3-PROXY`	`IPNOS : CONFIG VPN(1) VR(10.1.1.1)# fw pop3-proxy`
	`VR(10.1.1.1): Object successfully set`
	
	`IPNOS : CONFIG VPN(1) VR(10.1.1.1) POP3-PROXY#`

Default Value

None

Command Level

fw

Related Commands

aol-proxy
cs-proxy
ftp-proxy
http-proxy
ldap-proxy
lnotes-proxy
netm-proxy
nntp-proxy
plug-proxy
rsh-proxy
smtp-proxy
sql-proxy
telnet-proxy

child-limit

Set the maximum number of concurrent pop3 sessions.

Syntax	*Example*
CHILD-LIMIT <value>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) POP3-PROXY# **child-limit 64** IPNOS : CONFIG VPN(50) VR(10.0.16.1) POP3-PROXY# **display child-limit** CHILD-LIMIT : 64 LISTEN(s): List is empty POLICY(s): List is empty

Default Value

64 sessions

Command Level

pop3-proxy

Related Commands

ftp-proxy child-limit
http-proxy child-limit
plug-proxy child-limit
rsh-proxy child-limit
smtp-proxy child-limit
sql-proxy child-limit
telnet-proxy child-limit

display

View the current POP3 proxy configuration.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG VPN(50) VR(10.0.16.1) POP3-PROXY# **display** VPN : 50 VR : 10.0.16.1 PROXY : POP3-PROXY STATUS : ENABLE CHILD-LIMIT : 64 LISTEN(s): List is empty POLICY(s): List is empty
DISPLAY <parameter>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) POP3-PROXY# **display proxy** PROXY : POP3-PROXY

Default Value

None

Command Level

pop3-proxy

Related Commands

None

listen

Set the port that the proxy will listen to. Use the **no** option to delete a port from the list.

Syntax	*Example*
[NO] LISTEN <port>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) POP3-PROXY# **listen 110** LISTEN 110: Object successfully set IPNOS : CONFIG VPN(50) VR(10.0.16.1) POP3-PROXY# **display listen** No match LISTEN(s): 110 POLICY(s): List is empty

Default Value

None

Command Level

pop3-proxy

Related Commands

ftp-proxy listen
http-proxy listen
plug-proxy listen
rsh-proxy listen
smtp-proxy listen
sql-proxy listen
telnet-proxy listen

policy

Set the name of the proxy policy. The policy is an index to a particular group of proxy settings. The policy name is alphanumeric and can include "_" but no blank spaces. This command also puts you into the POP3 policy command level. Use the **no** option to delete a policy. The following commands are available at this level:

- "display" on page 18-63
- "pop3-server" on page 18-64
- "timeout" on page 18-65

Syntax	*Example*

`[NO] POLICY <name>`	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) POP3-PROXY# policy 1`
	`HTTP-PROXY: Object successfully set`
	
	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) POP3-PROXY POLICY(1)#`

Default Value

None

Command Level

pop3-proxy

Related Command

ftp-proxy policy
http-proxy policy
plug-proxy policy
rsh-proxy policy
smtp-proxy policy
sql-proxy policy
telnet-proxy policy

display

View the current POP3 policy configuration.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG VPN(50) VR(10.0.16.1) POP3-PROXY POLICY(1)# **display** VPN : 50 VR : 10.0.16.1 POLICY : 1 AUTHENTICATE : POP SERVER : TIMEOUT : 0

Default Value

None

Command Level

pop3-policy

Related Commands

None

pop3-server

Set the IP address of the pop3 server host used for handoff.

Syntax	*Example*
POP3-SERVER <host>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) POP3-PROXY POLICY(1)# **pop3-server 128.159.16.16** IPNOS : CONFIG VPN(50) VR(10.0.16.1) POP3-PROXY POLICY(1)# **display pop3-server** POP3-SERVER : 128.159.16.16
[NO] POP3-SERVER	IPNOS : CONFIG VPN(50) VR(3.3.3.3) POP3-PROXY POLICY(1)# **no pop3-server** IPNOS : CONFIG VPN(50) VR(3.3.3.3) POP3-PROXY POLICY(1)# **display pop3-server** POP3-SERVER :

Default Value

None

Command Level

pop3-policy

Related Commands

None

timeout

Set the idle connection timeout limit.

Syntax	*Example*
TIMEOUT <value>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) POP3-PROXY POLICY(1)# **timeout 4500** IPNOS : CONFIG VPN(50) VR(10.0.16.1) POP3-PROXY POLICY(1)# **display timeout** TIMEOUT : 4500
NO TIMEOUT	IPNOS : CONFIG VPN(50) VR(10.0.16.1) POP3-PROXY POLICY(1)# **no timeout** IPNOS : CONFIG VPN(50) VR(10.0.16.1) POP3-PROXY POLICY(1)# **display timeout** TIMEOUT : 7200

Default Value

7200 seconds (2 hours)

Command Level

pop3-policy

Related Command

timeout

rsh-proxy

Go to the remote shell (RSH) proxy command level. The following commands are available.

- "child-limit" on page 18-67
- "display" on page 18-67
- "listen" on page 18-68
- "policy" on page 18-69

Syntax	*Example*

`FW RSH-PROXY`	`IPNOS : CONFIG VPN(1) VR(10.1.1.1)# fw rsh-proxy`
	`VR(10.1.1.1): Object successfully set`
	
	`IPNOS : CONFIG VPN(1) VR(10.1.1.1) RSH-PROXY#`

Default Value

None

Command Level

fw

Related Commands

aol-proxy
cs-proxy
ftp-proxy
http-proxy
ldap-proxy
lnotes-proxy
netm-proxy
nntp-proxy
plug-proxy
pop3-proxy
smtp-proxy
sql-proxy
telnet-proxy

child-limit

Set the maximum number of concurrent remote shell sessions supported.

Syntax	*Example*
CHILD-LIMIT <value>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) RSH-PROXY# **child-limit 64** IPNOS : CONFIG VPN(50) VR(10.0.16.1) RSH-PROXY# **display child-limit** CHILD-LIMIT : 64 LISTEN(s): List is empty POLICY(s): List is empty

Default Value

64 sessions

Command Level

rsh-proxy

Related Commands

ftp-proxy child-limit
http-proxy child-limit
plug-proxy child-limit
pop3-proxy child-limit
smtp-proxy child-limit
sql-proxy child-limit
telnet-proxy child-limit

display

View the current rsh proxy configuration.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG VPN(50) VR(10.0.16.1) RSH-PROXY# **display** VPN : 50 VR : 10.0.16.1 PROXY : RSH-PROXY STATUS : ENABLE CHILD-LIMIT : 64 LISTEN(s): List is empty POLICY(s): List is empty IPNOS : CONFIG VPN(50) VR(10.0.16.1) RSH-PROXY#
DISPLAY <argument>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) RSH-PROXY# **display proxy** PROXY : RSH-PROXY STATUS : ENABLE LISTEN(s): List is empty POLICY(s): List is empty IPNOS : CONFIG VPN(50) VR(10.0.16.1) RSH-PROXY#

Default Value

None

Command Level

rsh-proxy

Related Commands

None

listen

Set the port that the proxy will listen to. Choose from 1 to 65,535. Use the **no** option to delete a port. Refer to the Internet Assigned Numbers Authority (IANA) at http://www.iana.org for the official port information.

Syntax	*Example*
[NO] LISTEN <port>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) RSH-PROXY# **listen 222** LISTEN 222: Object successfully set IPNOS : CONFIG VPN(50) VR(10.0.16.1) RSH-PROXY# **display listen** No match LISTEN(s): 222 POLICY(s): List is empty

Default Value

None

Command Level

rsh-proxy

Related Commands

ftp-proxy listen
http-proxy listen
plug-proxy listen
pop3-proxy listen
smtp-proxy listen
sql-proxy listen
telnet-proxy listen

policy

Set the name of the proxy policy. The policy is an index to a particular group of proxy settings. The policy name is alphanumeric and can include "_" but no blank spaces. This command also puts you into the remote shell policy command level. Use the **no** option to delete a policy. The following commands are available at this level:

- "display" on page 18-70
- "remote-server" on page 18-71
- "timeout" on page 18-71

Syntax	*Example*
[NO] POLICY <name>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) RSH-PROXY# **policy 1** RSH-PROXY: Object successfully set IPNOS : CONFIG VPN(50) VR(10.0.16.1) RSH-PROXY POLICY(1)#

Default Value

None

Command Level

rsh-proxy

Related Commands

ftp-proxy policy
http-proxy policy
plug-proxy policy
pop3-proxy policy
smtp-proxy policy
sql-proxy policy
telnet-proxy policy

display

View the current remote shell policy configuration.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG VPN(50) VR(10.0.16.1) RSH-PROXY POLICY(1)# **display** VPN : 50 VR : 10.0.16.1 POLICY : 1 TIMEOUT : 0 REMOTE SERVER :

Default Value

None

Command Level

rsh-policy

Related Command

None

remote-server

Set the IP address of the remote server to connect.

Syntax	*Example*
REMOTE-SERVER <host>	IPNOS : CONFIG VPN(1) VR(10.1.1.1) RSH-PROXY POLICY(1)# **remote-server 192.168.18.1** IPNOS : CONFIG VPN(1) VR(10.1.1.1) RSH-PROXY POLICY(1)# **display remote** REMOTE-SERVER : 192.168.18.1
NO REMOTE-SERVER	IPNOS : CONFIG VPN(1) VR(10.1.1.1) RSH-PROXY POLICY(1)# **no remote-server** IPNOS : CONFIG VPN(1) VR(10.1.1.1) RSH-PROXY POLICY(1)# **display remote** REMOTE-SERVER :

Default Value

None

Command Level

rsh-policy

Related Command

None

timeout

Set the idle connection timeout limit.

Syntax	*Example*
TIMEOUT <value>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) RSH-PROXY POLICY(1)# **timeout 4500** IPNOS : CONFIG VPN(50) VR(10.0.16.1) RSH-PROXY POLICY(1)# **display timeout** TIMEOUT : 4500
NO TIMEOUT	IPNOS : CONFIG VPN(50) VR(10.0.16.1) RSH-PROXY POLICY(1)# **no timeout** IPNOS : CONFIG VPN(50) VR(10.0.16.1) RSH-PROXY POLICY(1)# **display timeout** TIMEOUT : 7200

Default Value

7200 seconds (2 hours)

Command Level

rsh-policy

Related Command

timeout

smtp-proxy

Go to the Simple Mail Transfer Protocol (SMTP) proxy command level for outgoing electronic mail. The following commands are available.

- "child-limit" on page 18-73
- "display" on page 18-74
- "listen" on page 18-75
- "policy" on page 18-76

Syntax	*Example*

`FW SMTP-PROXY`	`IPNOS : CONFIG VPN(1) VR(10.1.1.1)# fw smtp-proxy`
	`VR(10.1.1.1): Object successfully set`
	
	`IPNOS : CONFIG VPN(1) VR(10.1.1.1) SMTP-PROXY#`

Default Value

None

Command Level

fw

Related Commands

aol-proxy
cs-proxy
ftp-proxy
http-proxy
ldap-proxy
lnotes-proxy
netm-proxy
nntp-proxy
plug-proxy
pop3-proxy
rsh-proxy
sql-proxy
telnet-proxy

child-limit

Set the maximum number of concurrent SMTP sessions supported.

Syntax	*Example*
CHILD-LIMIT <value>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY# **child-limit 64** IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY# **display child-limit** CHILD-LIMIT : 64 LISTEN(s): List is empty POLICY(s): List is empty

64

Command Level

smtp-proxy

Related Commands

ftp-proxy child-limit
http-proxy child-limit
plug-proxy child-limit
pop3-proxy child-limit
rsh-proxy child-limit
sql-proxy child-limit
telnet-proxy child-limit

display

View the current SMTP proxy configuration.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY# display VPN : 50 VR : 10.0.16.1 PROXY : SMTP-PROXY STATUS : ENABLE CHILD-LIMIT : 64 LISTEN(s): List is empty POLICY(s): List is empty IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY#
DISPLAY <argument>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY# **display proxy** PROXY : SMTP-PROXY STATUS : ENABLE LISTEN(s): List is empty POLICY(s): List is empty IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY#

Default Value

None

Command Level

smtp-proxy

Related Commands

None

listen

Set the port that the proxy will listen to. Use the **no** option to delete a port.

Syntax	*Example*
[NO] LISTEN <port>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY# **listen 25** LISTEN 25: Object successfully set IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY# **display listen** No match LISTEN(s): 25 POLICY(s): List is empty

Default Value

None

Command Level

smtp-proxy

Related Commands

ftp-proxy listen
http-proxy listen
plug-proxy listen
pop3-proxy listen
rsh-proxy listen
sql-proxy listen
telnet-proxy listen

policy

Set the name of the proxy policy. The policy is an index to a particular group of proxy settings. The policy name is alphanumeric and can include "_" but no blank spaces. This command also puts you into the SMTP policy command level. Use the **no** option to delete a policy. The following commands are available at this level:

- "display" on page 18-78
- "local-domain" on page 18-78
- "log-relay" on page 18-79

- "log-spam" on page 18-80
- "logging" on page 18-80
- "mail-hub" on page 18-81
- "max-bytes" on page 18-81
- "max-recip" on page 18-82
- "mime-check" on page 18-82
- "relay-host" on page 18-83
- "relay-reply" on page 18-84
- "spam-host" on page 18-84
- "spam-reply" on page 18-85
- "timeout" on page 18-85

Syntax	*Example*
[NO] POLICY <value>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY# **policy 1** HTTP-PROXY: Object successfully set IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY POLICY(1)#

Default Value

None

Command Level

smtp-proxy

Related Command

ftp-proxy policy
http-proxy policy
plug-proxy policy
pop3-proxy policy
rsh-proxy policy
sql-proxy policy
telnet-proxy policy

display

View the current SMTP policy configuration.

Syntax	*Example*

DISPLAY	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY POLICY(1)#` **`display`**
	`VPN : 50`
	`VR : 10.0.16.1`
	`POLICY : 1`
	
	`MAIL HUB :`
	`LOCAL DOMAIN :`
	
	`LOG RELAY :`
	`LOG SPAM :`
	`LOGGING :`
	
	`MAX BYTES :`
	`MAX RECEIPTS :`
	
	`RELAY :`
	`RELAY REPLAY :`
	
	`SPAM :`
	`SPAM REPLY :`
	
	`TIMEOUT : 0`

Default Value

None

Command Level

smtp-policy

Related Commands

None

local-domain

Specify the DNS name for the local domain that is being protected by the firewall. For example, a typical command might read as local-domain lasales.ibm.com. This value can be up to 64 characters long. This specified

domain is expected to also contains the mail hub (see “mail-hub” on page 18-81).

Syntax	*Example*
`LOCAL-DOMAIN <name>`	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY POLICY(1)# local-domain dns.company.com`

Default Value

None

Command Level

smtp-policy

Related Commands

None

log-relay

Enable or disable logging of relay-reject events to a syslogd server. None disables logging. See “syslogd” on page 16-151.

Syntax	*Example*
`LOG-RELAY {REJECT | NONE}`	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY POLICY(1)# log-relay reject`

Default Value

None

Command Level

smtp-policy

Related Commands

log-spam
logging

log-spam

Enable or disable logging of spam-reject events to a syslogd server. Choose None or Reject. None disables logging. See "syslogd" on page 16-151.

Syntax	*Example*
`LOG-SPAM {REJECT | NONE}`	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY POLICY(1)# log-spam reject`

Default Value

None

Command Level

smtp-policy

Related Commands

log-relay
logging

logging

Enable or disable logging of SMTP events to a syslogd server. See "syslogd" on page 16-151. Choose None or Normal. None disables logging. Because SMTP mail services typically carry huge volumes, indiscriminate logging may slow mail traffic and even network traffic.

Syntax	*Example*
`LOGGING {NONE | NORMAL}`	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY POLICY(1)# logging normal`

Default Value

None

Command Level

smtp-policy

Related Commands

log-relay
log-spam

mail-hub

Specify the name of a host to which the SMTP proxy may relay any permitted messages within the local mail domain. The host can be specified through a host name, such as mail.company.com, or an IP address. This value can be up to 128 characters long.

Syntax	*Example*
MAIL-HUB {<ip_address> | <domain_name>}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) SMTP-PROXY POLICY(1)# **mail-hub mail.company.com** IPNOS : CONFIG VPN(50) VR(2.2.2.2) SMTP-PROXY POLICY(1)# **display mail** MAIL-HUB : mail.company.com
MAIL-HUB {<ip_address> | <domain_name>}	IPNOS : CONFIG VPN(50) VR(2.2.2.2) SMTP-PROXY POLICY(1)# **mail-hub 192.168.16.4** IPNOS : CONFIG VPN(50) VR(2.2.2.2) SMTP-PROXY POLICY(1)# **display mail** MAIL-HUB : 192.168.16.4

Default Value

None

Command Level

smtp-policy

Related Commands

None

max-bytes

Set the maximum size, in bytes, of mail message bodies that the proxy will accept. File attachments are considered part of the body of a message.

Syntax	*Example*
`MAX-BYTES <value>`	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY POLICY(1)# max-bytes 1310720`

Default Value

0 (no byte-limit)

Command Level

smtp-policy

Related Commands

None

max-recip

Define the maximum number of recipients that can be specified in an email.

Syntax	*Example*
`MAX-RECIP <value>`	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY POLICY(1)# max-recip 32`

Default Value

0 (unlimited recipients)

Command Level

smtp-policy

Related Commands

None

mime-check

Check email file attachments.

Syntax	Example
MIME-CHECK { ON | OFF }	IPNOS : CONFIG VPN(50) VR(2.2.2.2) SMTP-PROXY POLICY(1)# **mime-check on** IPNOS : CONFIG VPN(50) VR(2.2.2.2) SMTP-PROXY POLICY(1)# display mime MIME-CHECK : ON
MIME-CHECK { ON | OFF }	IPNOS : CONFIG VPN(50) VR(2.2.2.2) SMTP-PROXY POLICY(1)# **mime-check off** IPNOS : CONFIG VPN(50) VR(2.2.2.2) SMTP-PROXY POLICY(1)# **display mime**

Default Value

off

Command Level

smtp-policy

Related Commands

None

relay-host

Create a list of hosts whose incoming email will be automatically allowed relay by the proxy. You can use domain names or IP addresses.

Syntax	Example
RELAY-HOST <address>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY POLICY(1)# **relay-host niceguy.com**

Default Value

None

Command Level

smtp-policy

Related Commands

None

relay-reply

Set the SMTP code used to reply to an email relay connection attempt. Choose from the following.

- 250 (requested mail action OK)
- 550 (mailbox unavailable)

Syntax	*Example*
`RELAY-REPLY {250 | 550}`	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY POLICY(1)# relay-reply 250`

Default Value

250

Command Level

smtp-policy

Related Commands

None

spam-host

Create a tabular list of domains or IP addresses of known spammers. Incoming email with domains that are on the list will be automatically rejected by the proxy. Domain names, IP addresses, and network addresses may be used in this table.

Syntax	*Example*
`SPAM-HOST <address>`	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY POLICY(1)# spam-host triplexxx.com`

Default Value

None

Command Level

smtp-policy

Related Commands

None

spam-reply

Set the SMTP code that will be used to reply to a known spammer. Choose from the following.

- 250 (requested mail action OK)
- 554 (transaction failed)

The 250 OK reply is sent to the spammer but the message is not delivered, and is silently dropped by the firewall.

Syntax	*Example*
`SPAM-REPLY {250 | 554}`	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY POLICY(1)# spam-reply 554`

Default Value

250

Command Level

smtp-policy

Related Commands

None

timeout

Set the timeout limit for an idle connection.

Syntax	*Example*
TIMEOUT <value>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY POLICY(1)# **timeout 4500** IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY POLICY(1)# **display timeout** TIMEOUT : 4500
NO TIMEOUT	IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY POLICY(1)# **no timeout** IPNOS : CONFIG VPN(50) VR(10.0.16.1) SMTP-PROXY POLICY(1)# **display timeout** TIMEOUT : 7200

Default Value

7200 seconds (2 hours)

Command Level

smtp-policy

Related Command

timeout

sql-proxy

Go tot the Microsoft SQL Server or Sybase SQL Server proxy command level. The following commands are available.

- "child-limit" on page 18-87
- "display" on page 18-88
- "listen" on page 18-89
- "policy" on page 18-90

Syntax	*Example*

`FW SQL-PROXY`	`IPNOS : CONFIG VPN(1) VR(10.1.1.1)# fw sql-proxy`
	`VR(10.1.1.1): Object successfully set`
	
	`IPNOS : CONFIG VPN(1) VR(10.1.1.1) SQL-PROXY#`

Default Value

None

Command Level

fw

Related Commands

aol-proxy
cs-proxy
ftp-proxy
http-proxy
ldap-proxy
lnotes-proxy
netm-proxy
nntp-proxy
plug-proxy
pop3-proxy
rsh-proxy
smtp-proxy
telnet-proxy

child-limit

Set the maximum number of concurrent SQL sessions supported.

Syntax	*Example*
CHILD-LIMIT <value>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) SQL-PROXY# **child-limit 64** IPNOS : CONFIG VPN(50) VR(10.0.16.1) SQL-PROXY# **display child-limit** CHILD-LIMIT : 64 LISTEN(s): List is empty POLICY(s): List is empty

Default Value

64 sessions

Command Level

sql-proxy

Related Commands

ftp-proxy child-limit
http-proxy child-limit
plug-proxy child-limit
pop3-proxy child-limit
rsh-proxy child-limit
smtp-proxy child-limit
telnet-proxy child-limit

display

View the current SQL proxy configuration.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG VPN(50) VR(10.0.16.1) SQL-PROXY# **display** VPN : 50 VR : 10.0.16.1 PROXY : SQL-PROXY STATUS : ENABLE CHILD-LIMIT : 64 LISTEN(s): List is empty POLICY(s): List is empty
DISPLAY <argument>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) SQL-PROXY# **display proxy** PROXY : SQL-PROXY STATUS : ENABLE LISTEN(s): List is empty POLICY(s): List is empty

Default Value

None

Command Level

sql-proxy

Related Commands

None

listen

Set the port that the proxy will listen to. Choose from 1 to 65,535. Use the **no** option to delete a port. Refer to the Internet Assigned Numbers Authority (IANA) at http://www.iana.org for the official port information.

Syntax	*Example*
[NO] LISTEN <port>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) SQL-PROXY# **listen 1498** LISTEN 1498: Object successfully set IPNOS : CONFIG VPN(50) VR(10.0.16.1) SQL-PROXY# **display listen** No match LISTEN(s): 1498 POLICY(s): List is empty

Default Value

None

Command Level

sql-proxy

Related Command

ftp-proxy listen
http-proxy listen
plug-proxy listen
pop3-proxy listen
rsh-proxy listen
smtp-proxy listen
telnet-proxy listen

policy

Set the name of the proxy policy. The policy is an index to a particular group of proxy settings. The policy name is alphanumeric and can include "_" but no blank spaces. This command also puts you into the SQL policy command level. Use the **no** option to delete a policy. The following commands are available.

- "display" on page 18-91
- "remote-port" on page 18-92
- "remote-server" on page 18-92

❖ "timeout" on page 18-93

Syntax	Example
[NO] POLICY <name>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) SQL-PROXY# **policy 1** PLUG-PROXY: Object successfully set IPNOS : CONFIG VPN(50) VR(10.0.16.1) SQL-PROXY POLICY(1)#

Default Value

None

Command Level

sql-proxy

Related Command

ftp-proxy policy
http-proxy policy
plug-proxy policy
pop3-proxy policy
rsh-proxy policy
smtp-proxy policy
telnet-proxy policy

display

View the current SQL policy configuration.

Syntax	Example
DISPLAY	IPNOS : CONFIG VPN(50) VR(10.0.16.1) SQL-PROXY POLICY(1)# **display** VPN : 50 VR : 10.0.16.1 POLICY : 1 TIMEOUT : 0 REMOTE PORT : REMOTE SERVER :

Default Value

None

Command Level

sql-policy

Related Command

None

remote-port

Set the port to connect to on the remote server. Choose from 1 to 65,535. Use the **no** option to reset the command to its default value. Refer to the Internet Assigned Numbers Authority (IANA) at http://www.iana.org for the official port information.

Syntax	*Example*
[NO] REMOTE-PORT <port>	IPNOS : CONFIG VPN(1) VR(10.1.1.1) SQl-PROXY POLICY(1)# **remote-port 1433** IPNOS : CONFIG VPN(1) VR(10.1.1.1) SQL-PROXY POLICY(1)# **display remote** REMOTE-SERVER : 192.168.18.1 REMOTE-PORT : 1433

Default Value

None

Command Level

sql-policy

Related Command

None

remote-server

Set the IP address of the remote server to plug into.

Syntax	Example
[NO] REMOTE-SERVER <host>	IPNOS : CONFIG VPN(1) VR(10.1.1.1) SQL-PROXY POLICY(1)# **remote-server 192.168.18.1** IPNOS : CONFIG VPN(1) VR(10.1.1.1) SQL-PROXY POLICY(1)# **display remote** REMOTE-SERVER : 192.168.18.1 REMOTE-PORT : 1433
[NO] REMOTE-SERVER <host>	IPNOS : CONFIG VPN(1) VR(10.1.1.1) SQL-PROXY POLICY(1)# **no remote-server** IPNOS : CONFIG VPN(1) VR(10.1.1.1) SQL-PROXY POLICY(1)# **display remote** REMOTE-SERVER : REMOTE-PORT : 1433

Default Value

None

Command Level

sql-policy

Related Command

None

timeout

Set the timeout limit for an idle connection.

Syntax	Example
TIMEOUT <value>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) SQL-PROXY POLICY(1)# **timeout 4500** IPNOS : CONFIG VPN(50) VR(10.0.16.1) SQL-PROXY POLICY(1)# **display timeout** TIMEOUT : 4500

Syntax	*Example*
NO TIMEOUT	IPNOS : CONFIG VPN(50) VR(10.0.16.1) SQL-PROXY POLICY(1)# **no timeout** IPNOS : CONFIG VPN(50) VR(10.0.16.1) SQL-PROXY POLICY(1)# **display timeout** TIMEOUT : 7200

Default Value

7200 seconds (2 hours)

Command Level

sql-policy

Related Command

timeout

telnet-proxy

Go to the telnet proxy command level. The following commands are available.

- "child-limit" on page 18-96
- "display" on page 18-96
- "listen" on page 18-97
- "policy" on page 18-98

Syntax	*Example*

`FW TELNET-PROXY`	`IPNOS : CONFIG VPN(1) VR(10.1.1.1)# fw telnet-proxy`
	`VR(10.1.1.1): Object successfully set`
	
	`IPNOS : CONFIG VPN(1) VR(10.1.1.1) TELNET-PROXY#`

Default Value

None

Command Level

fw

Related Commands

aol-proxy
cs-proxy
ftp-proxy
http-proxy
ldap-proxy
lnotes-proxy
netm-proxy
nntp-proxy
plug-proxy
pop3-proxy
rsh-proxy
smtp-proxy
sql-proxy

child-limit

Set the maximum number of concurrent telnet sessions supported by the proxy. Use the **no** option to reset the command to its default value.

Syntax	*Example*

`CHILD-LIMIT <value> -or- NO CHILD-LIMIT`	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) TELNET-PROXY#` **`child-limit 64`**
	
	`IPNOS : CONFIG VPN(50) VR(10.0.16.1) TELNET-PROXY#` **`display child-limit`**
	`CHILD-LIMIT  : 64`
	
	`LISTEN(s):`
	`  List is empty`
	
	`POLICY(s):`
	`  List is empty`

Default Value

64 sessions

Command Level

telnet-proxy

Related Commands

ftp-proxy child-limit
http-proxy child-limit
plug-proxy child-limit
pop3-proxy child-limit
rsh-proxy child-limit
smtp-proxy child-limit
sql-proxy child-limit

display

View the current telnet proxy configuration.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG VPN(50) VR(10.0.16.1) TELNET-PROXY# display VPN : 50 VR : 10.0.16.1 PROXY : TELNET-PROXY STATUS : ENABLE CHILD-LIMIT : 64 LISTEN(s): List is empty POLICY(s): List is empty IPNOS : CONFIG VPN(50) VR(10.0.16.1) TELNET-PROXY#
DISPLAY <argument>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) TELNET-PROXY# **display proxy** PROXY : PLUG-PROXY STATUS : ENABLE LISTEN(s): List is empty POLICY(s): List is empty IPNOS : CONFIG VPN(50) VR(10.0.16.1) TELNET-PROXY#

Default Value

None

Command Level

telnet-proxy

Related Commands

None

listen

Set the port that the proxy will listen to. Use the **no** option to delete a port.

Syntax	*Example*
[NO] LISTEN <port>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) TELNET-PROXY# **listen 23** LISTEN 23: Object successfully set IPNOS : CONFIG VPN(50) VR(10.0.16.1) TELNET-PROXY# **display listen** No match LISTEN(s): 23 POLICY(s): List is empty

Default Value

None

Command Level

telnet-proxy

Related Commands

ftp-proxy listen
http-proxy listen
plug-proxy listen
pop3-proxy listen
rsh-proxy listen
smtp-proxy listen
sql-proxy listen

policy

Set the name of the proxy policy. The policy is an index to a particular group of proxy settings. The policy name is alphanumeric and can include "_" but no blank spaces. This command also puts you into the telnet policy command level. Use the **no** option to delete a policy. The following commands are available at this level:

- "display" on page 18-99
- "timeout" on page 18-100

Syntax	*Example*
[NO] POLICY <name>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) TELNET-PROXY# **policy 1** HTTP-PROXY: Object successfully set IPNOS : CONFIG VPN(50) VR(10.0.16.1) TELNET-PROXY POLICY(1)#

Default Value

None

Command Level

telnet-proxy

Related Commands

ftp-proxy policy
http-proxy policy
plug-proxy policy
pop3-proxy policy
rsh-proxy policy
smtp-proxy policy
sql-proxy policy

display

View the current telnet policy configuration.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG VPN(1) VR(10.1.1.1) TELNET-PROXY# **display** VPN : 1 VR : 10.1.1.1 PROXY : TELNET-PROXY STATUS : ENABLE CHILD-LIMIT : 64 LISTEN(s): 23 POLICY(s): 1

Default Value

None

Command Level

telnet-policy

Related Command

None

timeout

Set the idle connection timeout limit.

Syntax	*Example*
TIMEOUT <value>	IPNOS : CONFIG VPN(50) VR(10.0.16.1) TELNET-PROXY POLICY(1)# **timeout 4500** IPNOS : CONFIG VPN(50) VR(10.0.16.1) TELNET-PROXY POLICY(1)# **display timeout** TIMEOUT : 4500
NO TIMEOUT	IPNOS : CONFIG VPN(50) VR(10.0.16.1) TELNET-PROXY POLICY(1)# **no timeout** IPNOS : CONFIG VPN(50) VR(10.0.16.1) TELNET-PROXY POLICY(1)# **display timeout** TIMEOUT : 7200

Default Value

7200 seconds (2 hours)

Command Level

telnet-policy

Related Command

timeout

CHAPTER

19 Frame Relay over IP Commands

Use these commands to configure a Frame Relay transport over an IP network. This chapter describes the following commands:

Command	*Page*
default-service-slot	19-2
default-window-size	19-3
default-window-timeout	19-4
display	19-5
local-vi	19-6
max-transport	19-7
transport	19-8
admin-status	19-9
display	19-10
dlci	19-11
bc	19-12
be	19-13
cir	19-14
display	19-15
route	19-16
event-window-size	19-17
keep-alive-period	19-17
permit-bad-event	19-18
window-size	19-19
window-timeout	19-19

default-service-slot

Assign the FRoIP process to a specific blade on the IPSX chassis. Choose from slots 1 to 26. This command is an optional feature.

Syntax	*Example*
DEFAULT-SERVICE-SLOT <slot-number>	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP# **default-service-slot 2** IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP# **display default-service-slot** DEFAULT-SERVICE-SLOT : 2

Default Value

Determined by IPNOS software

Command Level

froip

Related Commands

None

default-window-size

Set buffer size for holding packets for reordering. A large value for default-window-size decreases the possibility of out-of-order frames getting dropped, but at the cost of more memory and a possible drop in performance. You can store from 0 to 256 packets at a time. This command sets the default value for all new transports using FRoIP. To configure an individual transport, refer to “window-size” on page 19-19 at the transport level.

Syntax	*Example*
`DEFAULT-WINDOW-SIZE <value>`	`IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP#` **`default-window-size 128`** `IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP#` **`display default-window-size`** `DEFAULT-WINDOW-SIZE : 128`

Default Value

24 packets

Command Level

froip

Related Commands

None

default-window-timeout

Set time limit for holding packets before they are sent out to the Frame Relay network. The longer packets are held, the slower the performance, but it gives the network more time for the stray packets to reach their destination. You can hold packets from 10 to 25,000 milliseconds. This command sets the default value for all new transports using Frame Relay over IP. To configure an individual transport, refer to "window-timeout" on page 19-19 at the transport level.

Syntax	*Example*
`DEFAULT-WINDOW-TIMEOUT <value>`	`IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP#` **`default-window-timeout 1000`** `IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP#` **`display default-window-timeout`** `DEFAULT-WINDOW-TIMEOUT : 1000`

Default Value

1000 milliseconds

Command Level

froip

Related Commands

None

display

View the current Frame Relay over IP configuration.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP# **display** VPN : 1 VR : 10.1.1.1 LOCAL-VI : 1 MAX-TRANSPORT : 24 DEFAULT-WINDOW-SIZE : 10 DEFAULT-WINDOW-TIMEOUT : 1000 DEFAULT-SERVICE-SLOT : 2 ACTIVE TRANSPORTS : 0 TRANSPORT : List is empty IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP#
DISPLAY <argument>	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP# **display** transport TRANSPORT : List is empty IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP#

Default Value

None

Command Level

froip

Related Commands

None

local-vi

Control the path for IP packet transmission.

NOTE: The local-vi command is for advanced users and is only accessible by the ISP. The VI must be assigned an IP address for this command to be successful.

Syntax	*Example*
`LOCAL-VI <viId>`	`IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP#` **`local-vi 1`** `IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP#` **`display local-vi`** `LOCAL-VI : 1`

Default Value

1

Command Level

froip

Related Commands

None

max-transport

Set the maximum number of peer transports that can be used with FRoIP. You can have from 1 to 1024 transports enabled.

Syntax	*Example*
MAX-TRANSPORT <value>	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP# **max-transport 24** IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP# **display max-transport** MAX-TRANSPORT : 24

Default Value

24 peer transports

Command Level

froip

Related Commands

None

transport

Configure individual transport connections to remote peers for FRoIP. Use the remote VR's VI 1 IP address as the transport IP address. This creates one half of the transport connection. In the remote VR, create a FRoIP transport with the local VR's VI IP address to complete the connection.

This command also puts you into the transport level. Several commands are available at this level:

- "admin-status" on page 19-9
- "default-slot" on page 19-9
- "display" on page 19-10
- "dlci" on page 19-11
- "event-window-size" on page 19-17
- "keep-alive-period" on page 19-17
- "permit-bad-event" on page 19-18
- "window-size" on page 19-19
- "window-timeout" on page 19-19

Syntax	*Example*

`TRANSPORT <ip-address>`	`IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP#`
	`transport 10.1.1.3`
	`FRoIP: Object successfully set`
	
	`IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP`
	`TRANSPORT(10.1.1.3)#`

Default Value

None

Command Level

froip

Related Commands

None

admin-status

Enable the transport on an individual switch.

Syntax	*Example*
ADMIN-STATUS {ENABLE | DISABLE}	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3)# **admin-status enable** IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3)# **display admin** ADMIN-STATUS : Enable
ADMIN-STATUS {ENABLE | DISABLE}	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3)# **admin-status disable** IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3)# **display admin** ADMIN-STATUS : Disable

Default Value

disable

Command Level

transport

Related Commands

None

default-slot

Assign the FRoIP transport process to a specific blade on the IPSX chassis. Choose from slots 1 to 26. This command is an optional feature. The default-slot can be set only when creating a new FRoIP transport. Existing transports cannot be changed to a different slot.

Syntax	*Example*
DEFAULT-SLOT <slot-number>	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP# **transport 10.1.1.3** FRoIP: Object successfully set IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3)# **default-slot 2** IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3)# **display default-slot** DEFAULT-SLOT : 2

Default Value

Determined by IPNOS software

Command Level

transport

Related Commands

default-service-slot

display

View the current FRoIP transport configuration.

Syntax	Example
DISPLAY	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3)# **display** VPN : 1 VR : 10.1.1.1 TRANSPORT : 10.1.1.3 ADMIN-STATUS : Enable WINDOW-SIZE : 24 WINDOW-TIMEOUT : 1000 KEEP-ALIVE-PERIOD : 10 EVENT-WINDOW-SIZE : 4 PERMIT-BAD-EVENT : 3 DEFAULT-SLOT : 2 TRANSPORT MODE : ACTIVE OPER-STATUS : UP DLCI : 20
DISPLAY <argument>	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3)# **display transport** TRANSPORT : 10.1.1.3

Default Value

None

Command Level

transport

Related Commands

None

dlci

Configure the Frame Relay data link connection identifier (DLCI). The DLCI is the 10-bit permanent virtual circuit (PVC) termination point identification number. The maximum value is 1024. The number is placed in one endpoint of the Frame Relay connection. This command also puts you into the dlci level. Use the **no** option to delete a DLCI. Several commands are available at this level:

- "bc" on page 19-12

- "be" on page 19-13
- "cir" on page 19-14
- "display" on page 19-15
- "route" on page 19-16

Syntax	*Example*
[NO] DLCI <value>	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3)# **dlci 20** TRANSPORT(10.1.1.3): Object successfully set IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3) DLCI(20)#

Default Value

None

Command Level

transport

Related Commands

None

bc

Set the Frame Relay committed burst size (BC). BC is the amount of data committed by the provider to be transmitted over the measurement interval under normal network conditions. The measurement interval is determined by dividing BC by CIR. Typically, BC is set to the same value as the CIR, in which case the measurement interval is one second. If BC is set to zero, BE must not be zero. Use the **no** option to reset the command to its default value.

Syntax	*Example*
BC <value>	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3) DLCI(20)# **bc 12800** IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3) DLCI(20)# **display bc** BC : 12800

Syntax	Example
NO BC	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3) DLCI(20)# **no bc** no: Unrecognized command IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3) DLCI(20)# **dis bc** BC : 57600

Default Value

57600 bits

Command Level

dlci

Related Commands

be
cir

be

Configure Frame Relay excess burst size (BE). BE is the maximum amount of data beyond the committed burst size that may be burst-transmitted by the provider over the measurement interval. If BC is non-zero, the measurment interval is determined by dividing BC by CIR. If BC is set to zero, the measurement interval is determined by dividing BE by CIR. Use the **no** option to reset the command to its default value.

Syntax	Example
BE <value>	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3) DLCI(20)# **be 11000** IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3) DLCI(20)# **display be** BE : 11000

Syntax	*Example*
NO BE	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3) DLCI(20)# **no be** no: Unrecognized command IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3) DLCI(20)# **display be** BE : 131072

Default Value

131072 bits

Command Level

dlci

Related Commands

bc
cir

cir

Set the Frame Relay committed information rate (CIR) for the sub-interface. CIR is the minimum assigned bandwidth for the sub-interface from the host channel. If CIR is set to zero, BC is ignored and the measurement interval is determined by dividing BE by the bandwidth of the host channel. Use the **no** option to reset the command to its default value.

Syntax	*Example*
CIR <value>	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3) DLCI(20)# **cir 12800** IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3) DLCI(20)# **display cir** CIR : 12800
NO CIR	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3) DLCI(20)# **no cir** no: Unrecognized command IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3) DLCI(20)# **display cir** CIR : 57600

Default Value

57600 bits per second

Command Level

dlci

Related Commands

bc
be

display

Use the **display** command to show the current DLCI configuration.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3) DLCI(20)# **display** VPN : 1 VR : 10.1.1.1 TRANSPORT : 10.1.1.3 DLCI : 20 ROUTE : 2/1:0 DLCI 5 STATUS : ACTIVE BC : 57600 BE : 131072 CIR : 57600 RECEIVED-FECN : 0 RECEIVED-BECN : 0 SENT-FRAMES : 0 SENT-OCTETS : 0 RECEIVED-FRAMES : 0 RECEIVED-OCTETS : 0 MULTICAST : Invalid value(0) TYPE : Invalid value(0) DISCARDS : 0 RECEIVED-DE : 0 SENT-DE : 0
DISPLAY <argument>	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3) DLCI(20)# **display transport** TRANSPORT : 10.1.1.3

Default Value

None

Command Level

dlci

Related Commands

None

route

Use the **route** command to assign a temporary route identifier for a DLCI. Use the **no** option to delete the route Id. Table 19-1 shows the parameters available.

Table 19-1 DLCI Route Parameters

Parameter	*Description*
Slot	Blade slot number
Port	Port number
Channel	Channel number
DLCI {dcli}	Data Link Connection Identifier (DLCI)

Syntax	*Example*
ROUTE {<slot>/<port>:<channel> DLCI <dlci-number>}	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3) DLCI(20)# **route 2/1:0 DLCI 5** IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3) DLCI(20)# **display route** ROUTE : 2/1:0 DLCI 5
NO ROUTE	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3) DLCI(20)#**no route** IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3) DLCI(20)#**display route** ROUTE : No Route

Default Value

None

Command Level

dlci

Related Commands

None

event-window-size

Use the **event-window-size** command as a counter for the number of FRoIP events. The range is from 1 to 10 milliseconds.

Syntax	*Example*
EVENT-WINDOW-SIZE <value>	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3)# **event-window-size 4** IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3)# **display event** EVENT-WINDOW-SIZE : 4

Default Value

4 milliseconds

Command Level

transport

Related Commands

None

keep-alive-period

Use the **keep-alive-period** command to poll the Frame Relay over IP network transport to verify that it is working. A message is sent out over the network to query the status of the environment. You can set the frequency of sending the query message from 3 to 30 seconds.

Syntax	*Example*
KEEP-ALIVE-PERIOD <value>	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3)# **keep-alive-period 10** IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3)# **display keep-alive-period** KEEP-ALIVE-PERIOD : 10

Default Value

10 seconds

Command Level

transport

Related Commands

None

permit-bad-event

Use the **permit-bad-event** command to monitor events on a particular transport. When the number of bad events meets or exceeds the set amount, the transport is considered down. When the number of good events meets or exceeds the set amount, the transport is considered up. You can set from 1 to 10 events to monitor.

Syntax	*Example*
PERMIT-BAD-EVENT <value>	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3)# **permit-bad-event 3** IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3)# **display permit** PERMIT-BAD-EVENT : 3

Default Value

3 events

Command Level

transport

Related Commands

None

window-size

Use the **window-size** command to hold a number of packets into a buffer for reordering. A large value for default-window-size decreases the possibility of out-of-order frames getting dropped, but at the cost of more memory and a possible drop in performance. You can store from 0 to 256 packets at a time.

Syntax	*Example*
WINDOW-SIZE <value>	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3)# **window-size 128** IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3)# **display window-size** WINDOW-SIZE : 128

Default Value

24 packets

Command Level

transport

Related Commands

None

window-timeout

Use the **window-timeout** command to set the time limit for holding packets before they are sent out on the Frame Relay network. The longer packets are held, the slower the performance, but it gives the network more time for the stray packets to reach their destination. You can hold packets from 10 to 25,000 milliseconds.

Syntax	*Example*
WINDOW-TIMEOUT <value>	IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3)# **window-timeout 1000** IPNOS : CONFIG VPN(50) VR(10.1.1.1) FRoIP TRANSPORT(10.1.1.3)# **display window-timeout** WINDOW-TIMEOUT : 1000

Default Value

1000 milliseconds

Command Level

transport

Related Commands

None

CHAPTER

20 Routing Commands

Use these commands to configure routing protocols. See “router” on page 16-129. This chapter describes the following commands:

Command	*Page*
bgp	20-3
bgp	20-3
default-metric	20-8
display	20-9
enable	20-10
event-log	20-11
neighbor	20-12
network	20-19
show	20-19
isis	20-21
area-address	20-21
area-password	20-22
display	20-26
domain-password	20-26
default-info-originate	20-24
enable	20-28
event-log	20-30
export-policy	20-31
level-2-to-1-leaking	20-32
metric-style	20-34

Command	Page
mpls	20-36
spf-interval	20-39
summary-address	20-41
system-type	20-44
ospf	20-46
area	20-46
display	20-50
enable	20-52
event-log	20-52
export-policy	20-54
show	20-54
rip	20-56
display	20-56
enable	20-57
event-log	20-58
export-policy	20-62
import-policy	20-63
show	20-64

bgp

Go to the Border Gateway Protocol (BGP) command level. The following commands are available.

- "bgp" on page 20-3
- "default-metric" on page 20-8
- "display" on page 20-9
- "enable" on page 20-10
- "event-log" on page 20-11
- "neighbor" on page 20-12
- "network" on page 20-19
- "show" on page 20-19

bgp

Set global BGP variables. Table 20-1 describes the parameters.

Table 20-1 BGP Parameters

Parameter	*Description*
Confederation	Set the ID of the BGP confederation and the its local peer ASs. Use confederations to break a large AS into smaller, more manageable sub-ASs. Set the following: • Identifier—Confederation ID number. Choose from 1 to 65,534. A -1 indicates that no confederation ID is configured. • Peers—Add one or more BGP confederation peer sub-ASs. *Note:* Setting up confederations before connecting to a neighboring AS will cause conflicts. See "neighbor" on page 20-12.

Table 20-1 BGP Parameters (Continued)

Parameter	*Description*
Dampening	Enable BGP route flap dampening. Use the **no** option to disable flap dampening. You can also set the following parameters. • Half-life—The amount of time in seconds to reduce the penalty of an unstable route by one-half of the present value. Choose from 300 to 36,000 seconds. The default value is 900 seconds (15 minutes). • Reuse—When the penalty of a suppressed route falls below the reuse threshold, the route is installed in the forwarding table and is available for advertisement to other BGP peers. The default value is 2. This value must be less than the suppress threshold. • Suppress—Suppress a route when the penalty of the route exceeds this threshold. The route is not installed in the forwarding table of the router and it is not advertised to any of the router's peers. The default value is 3. • Max-suppress—The max. suppress metric guarantees that a route will be available for use within a predictable amount of time once it stabilizes. The default value is 16. This value must be greater than the suppress threshold.
Default	Set the default local preference within an AS. If multiple paths to the same destination are available, then the path with the largest local preference value is preferred for sending routes. The default value is 100.

Syntax	*Example*

`[NO] BGP CONFEDERATION IDENTIFIER <id>`	`IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# bgp confed identifier 200`
	
	`IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# commit`
	`Commit configuration`
	`BGP: Object successfully set`
	
	`IPNOS: CONFIG VPN(50) VR(3.3.3.3)# show bgp conf`
	`ROUTER BGP 333`
	`BGP CONFEDERATION IDENTIFIER 200`
	`NEIGHBOR cosine PEER-GROUP`
	`NEIGHBOR cosine REMOTE-AS 6942`
	`NEIGHBOR 172.16.24.2 REMOTE-AS 6942`
	`NEIGHBOR 172.16.24.2 PEER-GROUP cosine`
	`COMMIT # BGP`

Syntax	Example
[NO] BGP CONFEDERATION IDENTIFIER <id>	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **no bgp confederation identifier 200** IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **commit** Commit configuration BGP: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3)# **show bgp config** ROUTER BGP 333 NEIGHBOR cosine PEER-GROUP NEIGHBOR cosine REMOTE-AS 6942 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 PEER-GROUP cosine COMMIT # BGP
[NO] BGP CONFEDERATION PEERS as [as]}	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **bgp confederation peers 511 512** BGP: Object successfully set BGP: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show conf** ROUTER BGP 333 BGP CONFEDERATION IDENTIFIER 200 BGP CONFEDERATION PEERS 511 BGP CONFEDERATION PEERS 512 NEIGHBOR cosine PEER-GROUP NEIGHBOR cosine REMOTE-AS 6942 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 PEER-GROUP cosine COMMIT # BGP
[NO] BGP CONFEDERATION PEERS as [as]}	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **no bgp confederation peer 512** BGP: Object successfully set BGP CONFEDERATION PEERS 512 Object successfully deleted IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 BGP CONFEDERATION IDENTIFIER 200 BGP CONFEDERATION PEERS 511 NEIGHBOR cosine PEER-GROUP NEIGHBOR cosine REMOTE-AS 6942 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 PEER-GROUP cosine COMMIT # BGP

Syntax	*Example*
[NO] BGP DAMPENING [<half-life> <reuse> <suppress> <max-suppress>]}	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **bgp dampening 600 2 4 14** IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **commit** Commit configuration BGP: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3)# **show bgp conf** ROUTER BGP 333 BGP DAMPENING 600 2 4 14 NEIGHBOR cosine PEER-GROUP NEIGHBOR cosine REMOTE-AS 6942 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 PEER-GROUP cosine COMMIT # BGP
[NO] BGP DAMPENING [half-life reuse suppress max-suppress]}	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **no bgp damp** IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **commit** Commit configuration BGP: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3)# **show bgp conf** ROUTER BGP 333 NEIGHBOR cosine PEER-GROUP NEIGHBOR cosine REMOTE-AS 6942 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 PEER-GROUP cosine COMMIT # BGP
[NO] BGP DEFAULT LOCAL-PREFERENCE pref	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **bgp default local-preference 65** IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **commit** Commit configuration BGP: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3)# **show bgp config** ROUTER BGP 333 BGP DEFAULT LOCAL-PREFERENCE 65 BGP DAMPENING 600 2 4 14 NEIGHBOR cosine PEER-GROUP NEIGHBOR cosine REMOTE-AS 6942 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 PEER-GROUP cosine COMMIT # BGP

Syntax	Example
[NO] BGP DEFAULT LOCAL-PREFERENCE pref	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **no bgp default local-preference 65** IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **commit** Commit configuration BGP: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3)# **show bgp conf** ROUTER BGP 333 BGP DAMPENING 600 2 4 14 NEIGHBOR cosine PEER-GROUP NEIGHBOR cosine REMOTE-AS 6942 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 PEER-GROUP cosine COMMIT # BGP

Default Value

See "BGP Parameters" on page 20-3

Command Level

bgp

Related Commands

None

default-metric

Set the default metric for advertisement of all routes to external BGP (E-BGP) peers.

Syntax	*Example*
[NO] DEFAULT-METRIC metric	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **default 10** IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **commit** Commit configuration BGP: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3)# **show bgp config** ROUTER BGP 333 DEFAULT-METRIC 10 NEIGHBOR cosine PEER-GROUP NEIGHBOR cosine REMOTE-AS 6942 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 PEER-GROUP cosine COMMIT # BGP
[NO] DEFAULT-METRIC metric	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **no default 10** IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **commit** Commit configuration BGP: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3)# **show bgp config** ROUTER BGP 333 NEIGHBOR cosine PEER-GROUP NEIGHBOR cosine REMOTE-AS 6942 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 PEER-GROUP cosine COMMIT # BGP

Default Value

None

Command Level

bgp

Related Commands

None

display

View current router configuration.

Syntax	*Example*
DISPLAY	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **display** VPN : 50 VR : 3.3.3.3 BGP : ENABLE LOCAL-AS : 333 DEFAULT-METRIC : -1 LOCAL-PREFERENCE: 100 DAMPENING:DISABLE HALF-LIFE | REUSE | SUPPRESS | MAX-SUPPRESS 900 | 2 | 3 | 16 CONFEDERATION IDENTIFIER:-1 CONFEDERATION PEERS: List is empty PEER-GROUP(s): List is empty NEIGHBOR(s): List is empty PEER-GROUP IMPORT-POLICY(s): List is empty NEIGHBOR IMPORT-POLICY(s): List is empty PEER-GROUP EXPORT-POLICY(s): List is empty NEIGHBOR EXPORT-POLICY(s): List is empty
DISPLAY <argument>	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **display dampening** DAMPENING:DISABLE HALF-LIFE | REUSE | SUPPRESS | MAX-SUPPRESS 900 | 2 | 3 | 16

Default Value

None

Command Level

bgp

Related Commands

show

enable

Enable BGP routing on the VR.

Syntax	*Example*
[NO] BGP ENABLE	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **enable** IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show conf** ROUTER BGP 333 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 COMMIT # BGP
[NO] BGP ENABLE	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **no enable** IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **commit** Commit configuration BGP: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3)# **show bgp conf** ROUTER BGP 333 NO ENABLE NEIGHBOR 172.16.24.2 REMOTE-AS 6942 COMMIT # BGP

Default Value

Enable

Command Level

bgp

Related Commands

None

event-log

Enable event logging. Messages are sent to the screen and to a syslogd server. See "syslogd" on page 16-151. Use the **no** option to disable event-logs. Table 20-5 describes the log parameters.

Syntax	*Example*
[NO] EVENT-LOG AREA PACKET	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **event-log area packet** IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **commit** Commit configuration BGP: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3)# **show bgp config** ROUTER BGP 333 EVENT-LOG AREA PACKET NEIGHBOR 172.16.24.2 REMOTE-AS 6942 COMMIT # BGP
[NO] EVENT-LOG AREA PACKET	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **no event-log area packet** IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **commit** Commit configuration BGP: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3)# **show bgp config** ROUTER BGP 333 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 COMMIT # BGP

Default Value

Disable

Command Level

bgp

Related Commands

None

neighbor

Set BGP neighbor configuration. Set ACL, import-policy, and export-policy before running this command. See "acl" on page 16-7, "export-policy" on page 16-46 and "import-policy" on page 16-61. Table 20-2 describes the parameters.

Table 20-2 BGP Neighbor Parameters

Parameter	*Description*
Export	Add export policy. Set remote-as first.
Import	Add import policy. Set remote-as first.
EBGP-multihop	Add an E-BGP session to run between BGP routers that are not directly connected. The parameter serves as next hop router to the remote E-BGP peer. Set remote-as first.
Next-hop-self	Set current router as the next hop.
Password	Set alphanumeric password beteen BGP peers. Set remote-as first.
Passive	Set BGP to be passive. The peer does not initiate connection requests for establishing a peering session. Set remote-as first.
Peer-group	Create a peer-group name, and also assign a peer group name to a neighbor. Use the peer-group to make changes across all peers in a group, instead of individually. Set remote-as first.
Remote AS	Set the remote AS number of the neighbor. This command activates the connection.
Shutdown	Disable connectivity to a neighbor. Neighbor information is preserved. Set remote-as first.
Timers	Set the hold time limit for waiting to hear an UPDATE or KEEPALIVE message. Keepalive time is 1/3 hold time. The default value is 180 seconds. Minimum value is 20 seconds. Set remote-as first.
Update-source	Change the source IP address of outgoing BGP packets to an IP address other than the outgoing interface address. Typically used when the loopback address is used for peering. Set remote-as first.

Syntax	*Example*
[NO] NEIGHBOR {address|group-name} EXPORT policyId	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **NEIGHBOR 172.16.24.2 EXPORT 100** BGP: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 EXPORT 100 COMMIT # BGP
[NO] NEIGHBOR {address|group-name} EXPORT policyId	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **no NEIGHBOR 172.16.24.2 EXPORT 100** BGP: Object successfully set NEIGHBOR 172.16.24.2 EXPORT 100: Object successfully deleted IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 COMMIT # BGP
NEIGHBOR {address|group-name} EBGP-MULTIHOP ip-address	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **neighbor 172.16.24.2 ebgp-multihop 192.168.52.5** BGP: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 NEXT-HOP-SELF NEIGHBOR 172.16.24.2 EBGP-MULTIHOP 192.168.52.5 NEIGHBOR 172.16.24.3 REMOTE-AS 2468 NEIGHBOR 172.16.24.2 EXPORT 100 COMMIT # BGP
NO NEIGHBOR {address|group-name} EBGP-MULTIHOP	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **no neighbor 172.16.24.2 ebgp-multihop** BGP: Object successfully set NEIGHBOR 172.16.24.2 EBGP-MULTIHOP : Object successfully deleted IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 NEXT-HOP-SELF NEIGHBOR 172.16.24.3 REMOTE-AS 2468 NEIGHBOR 172.16.24.2 EXPORT 100 COMMIT # BGP

Syntax	*Example*
[NO] NEIGHBOR {address|group-name} IMPORT policyId	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **neighbor 172.16.24.2 import 200** BGP: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 NEXT-HOP-SELF NEIGHBOR 172.16.24.3 REMOTE-AS 2468 NEIGHBOR 172.16.24.2 IMPORT 200 NEIGHBOR 172.16.24.2 EXPORT 100 COMMIT # BGP
[NO] NEIGHBOR {address|group-name} IMPORT policyId	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **no neighbor 172.16.24.2 import 200** BGP: Object successfully set NEIGHBOR 172.16.24.2 IMPORT 200: Object successfully deleted IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 NEXT-HOP-SELF NEIGHBOR 172.16.24.2 EXPORT 100 COMMIT # BGP
[NO] NEIGHBOR {address|group-name} NEXT-HOP-SELF	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **neighbor 172.16.24.2 next-hop-self** BGP: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 NEXT-HOP-SELF NEIGHBOR 172.16.24.2 EXPORT 100 COMMIT # BGP
[NO] NEIGHBOR {address|group-name} NEXT-HOP-SELF	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **no neighbor 172.16.24.2 next-hop-self** BGP: Object successfully set NEIGHBOR 172.16.24.2 NEXT-HOP-SELF : Object successfully deleted IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.3 REMOTE-AS 2468 NEIGHBOR 172.16.24.2 EXPORT 100 COMMIT # BGP

Syntax	*Example*
NEIGHBOR {address|group-name} PASSWORD passwd-str	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **NEIGHBOR 172.16.24.2 password 012345** BGP: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 PASSWORD 012345 NEIGHBOR 172.16.24.2 EXPORT 100 COMMIT # BGP
NO NEIGHBOR {address|group-name} PASSWORD	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **no NEIGHBOR 172.16.24.2 password** BGP: Object successfully set NEIGHBOR 172.16.24.2 PASSWORD: Disabled successfully IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 EXPORT 100 COMMIT # BGP
[NO] NEIGHBOR address PASSIVE	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **NEIGHBOR 172.16.24.2 passive** BGP: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 PASSIVE NEIGHBOR 172.16.24.2 EXPORT 100 COMMIT # BGP
[NO] NEIGHBOR address PASSIVE	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **no NEIGHBOR 172.16.24.2 passive** BGP: Object successfully set NEIGHBOR 172.16.24.2 PASSIVE : Object successfully deleted IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 EXPORT 100 COMMIT # BGP

Syntax	*Example*
[NO] NEIGHBOR group-name PEER-GROUP	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **neighbor cosine peer-group** BGP: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show conf** ROUTER BGP 333 NEIGHBOR cosine PEER-GROUP NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 EXPORT 100 COMMIT # BGP
[NO] NEIGHBOR address PEER-GROUP group-name	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **NEIGHBOR 172.16.24.2 peer-group cosine** BGP: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 NEIGHBOR cosine PEER-GROUP NEIGHBOR cosine REMOTE-AS 6942 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 PEER-GROUP cosine NEIGHBOR 172.16.24.2 EXPORT 100 COMMIT # BGP
[NO] NEIGHBOR address PEER-GROUP group-name	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **no NEIGHBOR 172.16.24.2 peer-group cosine** BGP: Object successfully set NEIGHBOR 172.16.24.2 PEER-GROUP cosine: Object successfully deleted IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 NEIGHBOR cosine PEER-GROUP NEIGHBOR cosine REMOTE-AS 6942 COMMIT # BGP
[NO] NEIGHBOR {address|group-name} REMOTE-AS as-number	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **neighbor 172.16.24.2 remote 6942** BGP: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 COMMIT # BGP

Syntax	*Example*
[NO] NEIGHBOR {address|group-name} REMOTE-AS as-number	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **no neighbor 172.16.24.2 remote-as 6942** BGP: Object successfully set NEIGHBOR 172.16.24.2 REMOTE-AS 6942: Object successfully deleted IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 COMMIT # BGP
[NO] NEIGHBOR {address|group-name} SHUTDOWN	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **NEIGHBOR 172.16.24.2 shutdown** BGP: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 NEIGHBOR cosine PEER-GROUP NEIGHBOR cosine REMOTE-AS 6942 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 SHUTDOWN COMMIT # BGP
[NO] NEIGHBOR {address|group-name} SHUTDOWN	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **no NEIGHBOR 172.16.24.2 shutdown** BGP: Object successfully set NEIGHBOR 172.16.24.2 SHUTDOWN : Object successfully deleted IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 NEIGHBOR cosine PEER-GROUP NEIGHBOR cosine REMOTE-AS 6942 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 COMMIT # BGP
NEIGHBOR {address|group-name} TIMERS 0 holdtime	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **NEIGHBOR 172.16.24.2 timers 30 90** BGP: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 NEIGHBOR cosine PEER-GROUP NEIGHBOR cosine REMOTE-AS 6942 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 TIMERS 0 90 COMMIT # BGP

Syntax	*Example*
NO NEIGHBOR {address|group-name} TIMERS	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **no NEIGHBOR 172.16.24.2 timers** BGP: Object successfully set NEIGHBOR 172.16.24.2 TIMERS : Object successfully deleted IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 NEIGHBOR cosine PEER-GROUP NEIGHBOR cosine REMOTE-AS 6942 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 COMMIT # BGP
NEIGHBOR {address|group-name} UPDATE-SOURCE ip-address	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# N**EIGHBOR 172.16.24.2 update-source 10.1.1.1** BGP: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 NEIGHBOR cosine PEER-GROUP NEIGHBOR cosine REMOTE-AS 6942 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 UPDATE-SOURCE 10.1.1.1 COMMIT # BGP
NO NEIGHBOR {address|group-name} UPDATE-SOURCE	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **no NEIGHBOR 172.16.24.2 update-source** BGP: Object successfully set NEIGHBOR 172.16.24.2 UPDATE-SOURCE : Object successfully deleted IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 NEIGHBOR cosine PEER-GROUP NEIGHBOR cosine REMOTE-AS 6942 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 COMMIT # BGP

Default Value

None

Command Level

bgp

Related Commands

None

network

Export certain reachable routes. Routes specified by this command are exported to all BGP peers unless overridden by export policies. More convenient than **neighbor export** command for exporting connected routes to all peers. See "neighbor" on page 20-12.

Syntax	*Example*
[NO] NETWORK ipaddr [MASK ipmask]	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **network 172.16.24.3 mask 255.255.255.0** BGP: Object successfully set

Default Value

None

Command Level

bgp

Related Commands

neighbor

show

View the current BGP protocol configuration. This command is deprecated by the **show** command. See "show" on page 20-19.

Syntax	*Example*
SHOW CONFIGURATION	IPNOS: CONFIG VPN(50) VR(3.3.3.3) BGP(333)# **show config** ROUTER BGP 333 NEIGHBOR cosine PEER-GROUP NEIGHBOR cosine REMOTE-AS 6942 NEIGHBOR 172.16.24.2 REMOTE-AS 6942 NEIGHBOR 172.16.24.2 PEER-GROUP cosine COMMIT # BGP

Default Value

None

Command Level

bgp

Related Commands

BGP show

isis

Go to the Intermediate System-to-Intermediate System (IS-IS) routing protocol command level. The following commands are available.

- "area-address" on page 20-21
- "area-password" on page 20-22
- "default-info-originate" on page 20-24
- "display" on page 20-26
- "domain-password" on page 20-26
- "enable" on page 20-28
- "event-log" on page 20-30
- "export-policy" on page 20-31
- "level-2-to-1-leaking" on page 20-32
- "metric-style" on page 20-34
- "mpls" on page 20-36
- "show" on page 20-39
- "spf-interval" on page 20-39
- "summary-address" on page 20-41
- "system-id" on page 20-43
- "system-type" on page 20-44

area-address

Create an IS-IS area address. It is a 1-13 byte hexadecimal number.

Syntax	*Example*
[NO] AREA-ADDRESS identifier	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **area-address 0101.0101** ISIS: Object successfully set AREA-ADDRESS 0101.0101: Object successfully set IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **show config** ROUTER ISIS SYSTEM-ID 0200.0505.0505 AREA-ADDRESS 0101.0101 COMMIT # ISIS

Syntax	*Example*
[NO] AREA-ADDRESS identifier	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **no area-address 0101.0101** ISIS: Object successfully set AREA-ADDRESS 0101.0101: Object successfully deleted IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **show config** ROUTER ISIS SYSTEM-ID 0200.0505.0505 COMMIT # ISIS

Default Value

None

Command Level

IS-IS

Related Commands

None

area-password

Set the area password.

Syntax	Example

[NO] AREA-PASSWORD <text-string> | [SIMPLE] | [MD5]	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **area-password 0123456**
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis conf**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	  AREA-PASSWORD 0123456 SIMPLE
	AREA-ADDRESS 0101.0101
	COMMIT # ISIS

[NO] AREA-PASSWORD <text-string> | [SIMPLE] | [MD5]	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **no area-password 0123456**
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis conf**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	COMMIT # ISIS

Syntax	*Example*

[NO] AREA-PASSWORD <text-string> | [SIMPLE] | [MD5]	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **area-password 0123456 md5**
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	  AREA-PASSWORD 0123456 MD5
	AREA-ADDRESS 0101.0101
	COMMIT # ISIS

Default Value

None

Command Level

IS-IS

Related Commands

domain-password

default-info-originate

Inject a default route into Level-2 IS-IS routing domain to advertise a preferred default route to the Internet.

Syntax	*Example*

[NO] DEFAULT-INFO-ORIGINATE	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **default-info-originate**
	ISIS: Enabled successfully
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis conf**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  DEFAULT-INFO-ORIGINATE
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	COMMIT # ISIS

[NO] DEFAULT-INFO-ORIGINATE	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **no default-info-originate**
	ISIS: Disabled successfully
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis conf**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	COMMIT # ISIS

Default Value

Disable

Command Level

IS-IS

Related Commands

None

display

View current IS-IS configuration. This command is deprecated by the **show** command.

Syntax	*Example*

DISPLAY	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **display**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	COMMIT # ISIS

Default Value

None

Command Level

IS-IS

Related Commands

None

domain-password

Set the password for the entire IS-IS routing domain.

Syntax	*Example*

[NO] DOMAIN-PASSWORD <text-string> | [SIMPLE] | [MD5]	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **domain-password 0123456**
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	  DOMAIN-PASSWORD 0123456 SIMPLE
	AREA-ADDRESS 0101.0101
	COMMIT # ISIS

[NO] DOMAIN-PASSWORD <text-string> | [SIMPLE] | [MD5]	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **no domain-password 0123456**
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	COMMIT # ISIS

Syntax	*Example*

[NO] DOMAIN-PASSWORD <text-string> | [SIMPLE] | [MD5]	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **domain-password 0123456 md5**
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	  DOMAIN-PASSWORD 0123456 MD5
	AREA-ADDRESS 0101.0101
	COMMIT # ISIS

Default Value

None

Command Level

IS-IS

Related Commands

area-password

enable

Enable IS-IS routing.

Syntax	*Example*

[NO] ENABLE	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **enable**
	ISIS: Enabled successfully
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis conf**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	COMMIT # ISIS

[NO] ENABLE	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# no enable
	ISIS: Disabled successfully
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# commit
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# show isis config
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  NO ENABLE
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	COMMIT # ISIS

Default Value

Enable

Command Level

IS-IS

Related Commands

None

event-log

Enable event logging.

Syntax	*Example*

[NO] EVENT-LOG AREA { NONE | ADJACENCY | LSP | SNP | SPF | ALL | PACKET | PACKET-RECEIVE | PACKET-SEND }	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **event-log area all**
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **comm**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis conf**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	COMMIT # ISIS

[NO] EVENT-LOG AREA { NONE | ADJACENCY | LSP | SNP | SPF | ALL | PACKET | PACKET-RECEIVE | PACKET-SEND }	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **no event area all** IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **comm** Commit configuration ISIS: Object successfully set IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis conf** ROUTER ISIS SYSTEM-ID 0200.0505.0505 AREA-ADDRESS 0101.0101 COMMIT # ISIS

Default Value

Disable

Command Level

IS-IS

Related Commands

None

export-policy

Assign an export policy.

Syntax	*Example*

[NO] EXPORT-POLICY policyId	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **export-policy 100**
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  NO ENABLE
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	EXPORT-POLICY 100
	COMMIT # ISIS

Syntax	*Example*

[NO] EXPORT-POLICY policyId	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **no export-policy 100**
	ISIS: Object successfully set
	EXPORT-POLICY 100: Object successfully deleted
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  NO ENABLE
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	COMMIT # ISIS

Default Value

None

Command Level

IS-IS

Related Commands

None

level-2-to-1-leaking

Leak-down routes from level 2 to level 1.

Syntax	Example

[NO] LEVEL-2-TO-1-LEAKING	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **level-2**
	ISIS: Enabled successfully
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# s**how isis config**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  NO ENABLE
	  LEVEL-2-TO-1-LEAKING
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	COMMIT # ISIS

[NO] LEVEL-2-TO-1-LEAKING	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **no level-2**
	ISIS: Disabled successfully
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  NO ENABLE
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	COMMIT # ISIS

Default Value

Disable

Command Level

IS-IS

Related Commands

None

metric-style

Enable IS-IS networks to use a larger range of metrics for MPLS traffic engineering. Table 20-3 describes the parameters.

Table 20-3 MPLS Metric-Style Parameters

Parameter	*Description*
Wide	Accept only new-style type, length, and value (TLV) objects.
Narrow	Accept old-style TLV objects.
Transition	Accept both old-style and new-style TLVs.

Syntax	*Example*

[NO] METRIC-STYLE {WIDE | NARROW | TRANSITION}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **metric-style wide**
	ISIS: Object successfully set
	METRIC-STYLE: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  NO ENABLE
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	METRIC-STYLE WIDE
	COMMIT # ISIS

Syntax	Example

[NO] METRIC-STYLE {WIDE | NARROW | TRANSITION}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **metric-style narrow**
	ISIS: Object successfully set
	METRIC-STYLE: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  NO ENABLE
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	COMMIT # ISIS

[NO] METRIC-STYLE {WIDE | NARROW | TRANSITION}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **metric-style transition**
	ISIS: Object successfully set
	METRIC-STYLE: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  NO ENABLE
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	METRIC-STYLE TRANSITION
	COMMIT # ISIS

Syntax	*Example*

[NO] METRIC-STYLE {WIDE | NARROW | TRANSITION}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **no metric-style**
	ISIS: Object successfully set
	METRIC-STYLE: Disabled successfully
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  NO ENABLE
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	COMMIT # ISIS

Default Value

Wide

Command Level

IS-IS

Related Commands

mpls

mpls

Apply Multiprotocol Label Switching (MPLS) traffic engineering to IS-IS level 1 or level 2.

Syntax	Example

[NO] MPLS TRAFFIC-ENG {LEVEL-1 | LEVEL-2}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **mpls traffic level-1**
	ISIS: Object successfully set
	MPLS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  NO ENABLE
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	MPLS TRAFFIC-ENG LEVEL-1
	COMMIT # ISIS

[NO] MPLS TRAFFIC-ENG {LEVEL-1 | LEVEL-2}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **mpls traffic level-2**
	ISIS: Object successfully set
	MPLS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  NO ENABLE
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	MPLS TRAFFIC-ENG LEVEL-2
	COMMIT # ISIS

Syntax	*Example*

[NO] MPLS TRAFFIC-ENG {LEVEL-1 | LEVEL-2}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **no mpls traffic level-2**
	ISIS: Object successfully set
	MPLS: Disabled successfully
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  NO ENABLE
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	COMMIT # ISIS

Default Value

Disable

Command Level

IS-IS

Related Commands

None

show

View the current IS-IS configuration.

Syntax	*Example*

`SHOW CONFIG`	`IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS#` **`show config`**
	`ROUTER ISIS`
	`  SYSTEM-ID 0200.0505.0505`
	`  SPF-INTERVAL 10 LEVEL-1`
	`  EVENT-LOG AREA ADJACENCY`
	`  EVENT-LOG AREA LSP`
	`  EVENT-LOG AREA SNP`
	`  EVENT-LOG AREA SPF`
	`  EVENT-LOG AREA ALL`
	`AREA-ADDRESS 0101.0101`
	`COMMIT  # ISIS`

Default Value

None

Command Level

IS-IS

Related Commands

None

spf-interval

Set minimum time between shortest path first (SPF) computations when a network changes. Set for both level 1 and level 2 IS-IS networks.

Syntax	*Example*

[NO] SPF-INTERVAL value {level-1 | level-2 }	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **spf-interval 10 level-1**
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  SPF-INTERVAL 10 LEVEL-1
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	COMMIT # ISIS

[NO] SPF-INTERVAL value {level-1 | level-2 }	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **spf-interval 15 level-2**
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  SPF-INTERVAL 10 LEVEL-1
	  SPF-INTERVAL 15 LEVEL-2
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	COMMIT # ISIS

Syntax	Example

NO SPF-INTERVAL <LEVEL-1|LEVEL-2|LEVEL-1-2>	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **no spf-interval level-1-2**
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	COMMIT # ISIS

Default Value

5 seconds

Command Level

IS-IS

Related Commands

None

summary-address

Summarize routes from IS-IS level 1 to level 2.

Syntax	*Example*

[NO] SUMMARY-ADDRESS prefix mask	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **summary-address 176.24.0.0 255.255.0.0**
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	SUMMARY-ADDRESS 172.16.0.0 255.255.0.0
	SUMMARY-ADDRESS 176.24.0.0 255.255.0.0
	COMMIT # ISIS

[NO] SUMMARY-ADDRESS prefix mask	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **no summary-address 172.16.0.0 255.255.0.0**
	ISIS: Object successfully set
	SUMMARY-ADDRESS 172.16.0.0: Object successfully deleted
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	SUMMARY-ADDRESS 176.24.0.0 255.255.0.0
	COMMIT # ISIS

Default Value

None

Command Level

IS-IS

Related Commands

None

system-id

Change the system identifier. A 6 byte address specified in a hexadecimal format (xxxx.xxxx.xxxx).

Syntax	*Example*

SYSTEM-ID <address>	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **system-id 1722.4032.0001**
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config**
	ROUTER ISIS
	  SYSTEM-ID 1722.4032.0001
	  NO ENABLE
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	COMMIT # ISIS

Default Value

None

Command Level

IS-IS

Related Commands

None

system-type

Set IS-IS router type. Choose from level 1, level 2, or both levels 1 and 2.

Syntax	*Example*

[NO] SYSTEM-TYPE level-1 | level-2 | level-1-2	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **system-type level-1**
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  SYSTEM-TYPE LEVEL-1
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	SUMMARY-ADDRESS 176.24.0.0 255.255.0.0
	COMMIT # ISIS

[NO] SYSTEM-TYPE level-1 | level-2 | level-1-2	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **system-type level-2**
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  SYSTEM-TYPE LEVEL-2
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	SUMMARY-ADDRESS 176.24.0.0 255.255.0.0
	COMMIT # ISIS

Syntax	*Example*

[NO] SYSTEM-TYPE level-1 | level-2 | level-1-2	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **system-type level-1-2**
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5) ISIS# **commit**
	Commit configuration
	ISIS: Object successfully set
	
	IPNOS: CONFIG VPN(50) VR(5.5.5.5)# **show isis config**
	ROUTER ISIS
	  SYSTEM-ID 0200.0505.0505
	  EVENT-LOG AREA ADJACENCY
	  EVENT-LOG AREA LSP
	  EVENT-LOG AREA SNP
	  EVENT-LOG AREA SPF
	  EVENT-LOG AREA ALL
	AREA-ADDRESS 0101.0101
	SUMMARY-ADDRESS 176.24.0.0 255.255.0.0
	COMMIT # ISIS

Default Value

Level-1-2 (level 1 and level 2 router)

Command Level

IS-IS

Related Commands

None

ospf

Go to the Open Shortest Path First (OSPF) routing protocol command level. The following commands are available in this level:

- "area" on page 20-46
- "display" on page 20-50
- "enable" on page 20-52
- "event-log" on page 20-52
- "export-policy" on page 20-54
- "show" on page 20-54

Syntax	*Example*
`ROUTER OSPF`	`IPNOS : CONFIG VPN(50) vr(1.1.1.10)# router ospf` `IPNOS : CONFIG VPN(50) vr(1.1.1.10) OSPF#`

Default Value

None

Command Level

router command

Related Commands

bgp rip
bgp bgp

area

Set the OSPF area identification number. Once the area is set, you can configure other features based on RFC 2328. Use the **no** option to delete a command or to reset it to its default value. Table 20-4 describes the parameters available.

Table 20-4 OSPF Area Parameters

Parameter	*Description*
AreaID	Set the area identifier. Area 0.0.0.0 is used for the OSPF backbone.

Table 20-4 OSPF Area Parameters (Continued)

Parameter	*Description*
AreaID default-cost	Inject a default summary route into a stub or NSSA area. Choose from 1 to 65,535. Set the area as a stub or NSSA first.
AreaID NSSA	Assign the OSPF area as a ”Not so stubby area” (NSSA).
AreaID Range	Summarize and filter inter-area routes. Set the following options: • Address—IP address • Mask—Subnet mask • Not-advertise—Prevent the generation of external routes (LSA type 5 or 7) into the OSPF backbone. • NSSA—Import NSSA external routes (LSA type 7 packets)
AreaID Stub [no-summary]	Assign the OSPF area as a stub area. Stub areas have only one way in or out of the area. By default the area sends a summary of its routes to the adjacent area. You can stop the area from sending summaries by using the no-summary parameter.

Syntax	*Example*
[NO] AREA <areaid>	IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **area 1.1.1.1** OSPF: Object successfully set AREA 1.1.1.1: Object successfully set IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **show conf** ROUTER OSPF AREA 0.0.0.1 AREA 1.1.1.1 COMMIT # OSPF
[NO] AREA <areaId>	IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **no area 1** OSPF: Object successfully set AREA 1: Object successfully deleted IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **show conf** ROUTER OSPF AREA 1.1.1.1 COMMIT # OSPF
[NO] AREA <areaId> STUB [NO-SUMMARY]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **area 1.1.1.1 stub** OSPF: Object successfully set AREA 1.1.1.1 STUB: Object successfully set IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **show conf** ROUTER OSPF AREA 1.1.1.1 STUB COMMIT # OSPF

Syntax	*Example*
[NO] AREA <areaId> STUB [NO-SUMMARY]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **area 1.1.1.1 stub no-summary** OSPF: Object successfully set AREA 1.1.1.1 STUB: Object successfully set IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **show conf** ROUTER OSPF AREA 1.1.1.1 STUB NO-SUMMARY COMMIT # OSPF
[NO] AREA <areaId> STUB [NO-SUMMARY]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **no area 1.1.1.1 stub** OSPF: Object successfully set AREA 1.1.1.1: OSPF stub disable successful IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **show config** ROUTER OSPF AREA 1.1.1.1 COMMIT # OSPF
[NO] AREA <areaId> NSSA	IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **area 1.1.1.1 nssa** OSPF: Object successfully set AREA 1.1.1.1 NSSA: Object successfully set IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **show conf** ROUTER OSPF AREA 1.1.1.1 NSSA COMMIT # OSPF
AREA <areaId> DEFAULT-COST <metric>	IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **area 1.1.1.1 default 5** OSPF: Object successfully set AREA 1.1.1.1 DEFAULT-COST: Object successfully set IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **show conf** ROUTER OSPF AREA 1.1.1.1 NSSA AREA 1.1.1.1 DEFAULT-COST 5 COMMIT # OSPF
[NO] AREA <areaId> DEFAULT-COST	IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **no area 1.1.1.1 default** OSPF: Object successfully set AREA 1.1.1.1 DEFAULT-COST: Object successfully deleted IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **show conf** ROUTER OSPF AREA 1.1.1.1 NSSA COMMIT # OSPF

Syntax	Example
[NO] AREA areaId RANGE address mask [NOT-ADVERTISE] [NSSA]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **area 1.1.1.1 range 104.168.32.0 255.255.255.0** OSPF: Object successfully set AREA 1.1.1.1 RANGE: Object successfully set IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **show conf** ROUTER OSPF AREA 1.1.1.1 NSSA AREA 1.1.1.1 RANGE 104.168.32.0 255.255.255.0 COMMIT # OSPF
[NO] AREA areaId RANGE address mask [NOT-ADVERTISE] [NSSA]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **no area 1.1.1.1 range 104.168.32.0 255.255.255.0** OSPF: Object successfully set AREA 1.1.1.1 RANGE 104.168.32.0 255.255.255.0: Object successfully deleted IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **show conf** ROUTER OSPF AREA 1.1.1.1 NSSA COMMIT # OSPF
[NO] AREA areaId RANGE address mask [NOT-ADVERTISE] [NSSA]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **area 1.1.1.1 range 104.168.32.0 255.255.255.0 not-advertise** OSPF: Object successfully set AREA 1.1.1.1 RANGE: Object successfully set IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **show conf** ROUTER OSPF AREA 1.1.1.1 NSSA AREA 1.1.1.1 RANGE 104.168.32.0 255.255.255.0 NOT-ADVERTISE COMMIT # OSPF
[NO] AREA areaId RANGE address mask [NOT-ADVERTISE] [NSSA]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **area 1.1.1.1 range 104.168.32.0 255.255.255.0 nssa** OSPF: Object successfully set AREA 1.1.1.1 RANGE: Object successfully set IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **show conf** ROUTER OSPF AREA 1.1.1.1 NSSA AREA 1.1.1.1 RANGE 104.168.32.0 255.255.255.0 NSSA-RANGE COMMIT # OSPF

Syntax	*Example*
[NO] AREA areaId RANGE address mask [NOT-ADVERTISE] [NSSA]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **no area 1.1.1.1 range 104.168.32.0 255.255.255.0 nssa** OSPF: Object successfully set AREA 1.1.1.1 RANGE 104.168.32.0 255.255.255.0: Object successfully deleted IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **show conf** ROUTER OSPF AREA 1.1.1.1 NSSA COMMIT # OSPF

Default Value

None

Command Level

ospf

Related Commands

None

display

View the current OSPF configuration. This command is deprecated by the **show** command. See "show" on page 20-54.

Syntax	*Example*

```
DISPLAY  IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# display
         VPN              : 50
         VR               : 7.7.7.7
         OSPF             : ENABLE
         ROUTERID         : 0.0.0.0

         AREA(s):
         AREA             | TYPE             | STUB
         1.1.1.1          | NSSA             |

         DEFAULT COST(s):
         List is empty

         AREA AGGREGATE(s):
         AREA     | IP             | NETMASK        | EFFECT    | TYPE
         1.1.1.1  | 104.168.32.0   | 255.255.255.0  | ADVERTISE | NSSA-RANGE

         NEIGHBOR(s):
         List is empty

         EXPORT-POLICY(s):
         List is empty

         LSDB(s):
         AREA    | TYPE | LSID    | ROUTERID | SEQUENCE| AGE     | CHECKSUM
         1.1.1.1 | 1    | 7.7.7.7 | 7.7.7.7  | 80000001| 671     | f16

         EXT. LSDB(s):
         List is empty

         EVENT-LOG LEVEL: NONE

         EVENT-LOG PACKET: none

         EVENT-LOG EVENTS: none
```

Default Value

None

Command Level

ospf

Related Commands

show

enable

Enable OSPF routing on the VR.

Syntax	*Example*

[NO} ENABLE	`IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF#` **`enable`**
	`OSPF: Enabled successfully`
	
	`IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF#` **`show conf`**
	`ROUTER OSPF`
	`AREA 1.1.1.1 NSSA`
	`AREA 1.1.1.1 RANGE 104.168.32.0 255.255.255.0 NSSA-RANGE`
	`COMMIT # OSPF`

[NO} ENABLE	`IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF#` **`no enable`**
	`OSPF: Object successfully set`
	`OSPF: Disabled successfully`
	
	`IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF#` **`show conf`**

Default Value

None

Command Level

ospf

Related Commands

None

event-log

Enable event-log messages.

Syntax	*Example*
[NO] EVENT-LOG AREA [PACKET | EVENTS]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **event-log area packet** OSPF: Object successfully set EVENT-LOG PACKET: Ospf Log successfully set IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# show conf ROUTER OSPF AREA 1.1.1.1 NSSA AREA 1.1.1.1 RANGE 104.168.32.0 255.255.255.0 NSSA-RANGE COMMIT # OSPF
[NO] EVENT-LOG AREA [PACKET | EVENTS]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **no event-log area packet** IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **commit** Commit configuration OSPF: Object successfully set IPNOS: CONFIG VPN(50) VR(7.7.7.7)#
[NO] EVENT-LOG LEVEL [CRITICAL | MAJOR | MINOR | INFORM | DEBUG]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **event-log level major** OSPF: Object successfully set EVENT-LOG: Ospf Log successfully set IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **comm** Commit configuration OSPF: Object successfully set IPNOS: CONFIG VPN(50) VR(7.7.7.7)# **show ospf conf** ROUTER OSPF AREA 1.1.1.1 NSSA AREA 1.1.1.1 RANGE 104.168.32.0 255.255.255.0 NSSA-RANGE COMMIT # OSPF

Default Value

None

Command Level

ospf

Related Commands

None

export-policy

Apply an export policy to OSPF.

Syntax	*Example*
[NO] EXPORT-POLICY <policyId>	IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **export 100** OSPF: Object successfully set IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **show conf** ROUTER OSPF AREA 1.1.1.1 NSSA AREA 1.1.1.1 RANGE 104.168.32.0 255.255.255.0 NSSA-RANGE EXPORT-POLICY 100 COMMIT # OSPF
[NO] EXPORT-POLICY <policyId>	IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **no export 100** OSPF: Object successfully set EXPORT-POLICY 100: Object successfully deleted IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **show conf** ROUTER OSPF AREA 1.1.1.1 NSSA AREA 1.1.1.1 RANGE 104.168.32.0 255.255.255.0 NSSA-RANGE COMMIT # OSPF

Default Value

None

Command Level

ospf

Related Command

VR export-policy

show

View current OSPF configuration.

Syntax	Example
SHOW {CONFIGURATION | OSPF}	IPNOS: CONFIG VPN(50) VR(7.7.7.7) OSPF# **show config** ROUTER OSPF AREA 1.1.1.1 NSSA AREA 1.1.1.1 RANGE 104.168.32.0 255.255.255.0 NSSA-RANGE COMMIT # OSPF

Default Value

None

Command Level

ospf

Related Commands

None

rip

Go to the Routing Information Protocol (RIP) command level. The following commands are available.

- "display" on page 20-56
- "enable" on page 20-57
- "event-log" on page 20-58
- "export-policy" on page 20-62
- "import-policy" on page 20-63
- "show" on page 20-64

display

View the current RIP configuration.

Syntax	*Example*
DISPLAY	IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **display** VPN : 50 VR : 1.1.1.1 RIP : ENABLE IMPORT-POLICY(s): List is empty EXPORT-POLICY(s): List is empty RIPv2RteChanges : 0
DISPLAY <argument>	IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **display import** IMPORT-POLICY(s): List is empty

Default Value

None

Command Level

rip

Related Commands

show

enable

Use the **enable** command to enable RIP. Use the **no** option to disable the protocol.

Syntax	*Example*
[NO] ENABLE	IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **enable** RIP: Enabled successfully IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **display rip** RIP : ENABLE IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP#
[NO] ENABLE	IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **no enable** RIP: Disabled successfully IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **display rip** RIP : DISABLE

Default Value

Enable

Command Level

rip

Related Commands

None

event-log

Enable event logging. Messages are sent to the screen and to a syslogd server. See "syslogd" on page 16-151. Use the **no** option to disable event-logs. Table 20-5 describes the log parameters.

Table 20-5 RIP Event-Log Parameters

Parameter	*Description*
Area	Sets the event log areas to display RIP area messages. The areas are the following: • Packet—RIP packet messages. • Policy—RIP policy -related messages. • All—Packet and policy messages.
Level	Sets the severity level. Only event log messages of that level or higher are displayed. The levels are the following: • Critical—Only critical routing messages. • Error—All of the above messages plus routing errors. • Warning—All of the above messages plus system warnings. • Info—All of the above messages plus general system information. • Debug—All of the above messages plus debug reports.

Syntax	*Example*

`[NO] LOG AREA {PACKET | POLICY | ALL}`	`IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP#` **`event area packet`**
	
	`IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP#` **`display`**
	`VPN : 50`
	`VR : 1.1.1.1`
	`RIP : ENABLE`
	`EVENT-LOG : AREA PACKET`
	
	`IMPORT-POLICY(s):`
	`List is empty`
	
	`EXPORT-POLICY(s):`
	`List is empty`
	
	`RIPv2RteChanges : 0`

Syntax	Example
[NO] LOG AREA {PACKET | POLICY | ALL}	IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **event area policy** IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **display** VPN : 50 VR : 1.1.1.1 RIP : ENABLE EVENT-LOG : AREA POLICY IMPORT-POLICY(s): List is empty EXPORT-POLICY(s): List is empty RIPv2RteChanges : 0
[NO] LOG AREA {PACKET | POLICY | ALL}	IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **event area all** IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **display** VPN : 50 VR : 1.1.1.1 RIP : ENABLE EVENT-LOG : AREA ALL IMPORT-POLICY(s): List is empty EXPORT-POLICY(s): List is empty RIPv2RteChanges : 0
[NO] LOG AREA {PACKET | POLICY | ALL}	IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **no event area all** IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **display** VPN : 50 VR : 1.1.1.1 RIP : ENABLE IMPORT-POLICY(s): List is empty EXPORT-POLICY(s): List is empty RIPv2RteChanges : 0

Syntax	*Example*
[NO] LOG AREA {PACKET | POLICY | ALL}	IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **event area policy** IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **display** VPN : 50 VR : 1.1.1.1 RIP : ENABLE EVENT-LOG : AREA POLICY IMPORT-POLICY(s): List is empty EXPORT-POLICY(s): List is empty RIPv2RteChanges : 0
[NO] LOG AREA {PACKET | POLICY | ALL}	IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **event area all** IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **display** VPN : 50 VR : 1.1.1.1 RIP : ENABLE EVENT-LOG : AREA ALL IMPORT-POLICY(s): List is empty EXPORT-POLICY(s): List is empty RIPv2RteChanges : 0
[NO] LOG AREA {PACKET | POLICY | ALL}	IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **no event area all** IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **display** VPN : 50 VR : 1.1.1.1 RIP : ENABLE IMPORT-POLICY(s): List is empty EXPORT-POLICY(s): List is empty RIPv2RteChanges : 0

Syntax	*Example*
[NO] LOG LEVEL {CRITICAL | ERROR | WARNING | INFO | DEBUG}	IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **event level info** IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **display** VPN : 50 VR : 1.1.1.1 RIP : ENABLE EVENT-LOG : AREA ALL EVENT-LOG : LEVEL CRITICAL|ERROR|WARNING|INFORM IMPORT-POLICY(s): List is empty EXPORT-POLICY(s): List is empty RIPv2RteChanges : 0
[NO] LOG LEVEL {CRITICAL | ERROR | WARNING | INFO | DEBUG}	IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **no event level info** IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **display** VPN : 50 VR : 1.1.1.1 RIP : ENABLE IMPORT-POLICY(s): List is empty EXPORT-POLICY(s): List is empty RIPv2RteChanges : 0

Default Value

Disabled

Command Level

rip

Related Commands

None

export-policy

Add an export policy.

Syntax	*Example*
[NO] EXPORT-POLICY <policyID>	IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **export 100** RIP: Object successfully set IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **show config** CREATE VPN 50 CREATE VR 1.1.1.1 ROUTER RIP EXPORT-POLICY 100 COMMIT # RIP VI 1 COMMIT # VI 1 VI 2 ROUTER RIP COMMIT # VI 2 COMMIT # VR 1.1.1.1 ROUTING
[NO] EXPORT-POLICY <policyID>	IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **no export 100** RIP: Object successfully set EXPORT-POLICY 100: Object successfully deleted IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# show config CREATE VPN 50 CREATE VR 1.1.1.1 ROUTER RIP COMMIT # RIP VI 1 COMMIT # VI 1 VI 2 ROUTER RIP COMMIT # VI 2 COMMIT # VR 1.1.1.1 ROUTING

Default Value

None

Command Level

rip

Related Commands

bgp rip
export-policy
import-policy

import-policy

Add an import policy.

Syntax	*Example*
[NO] IMPORT-POLICY <policyID>	IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **import 200** RIP: Object successfully set IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **show config** CREATE VPN 50 CREATE VR 1.1.1.1 ROUTER RIP IMPORT-POLICY 200 COMMIT # RIP VI 1 COMMIT # VI 1 VI 2 ROUTER RIP COMMIT # VI 2 COMMIT # VR 1.1.1.1 ROUTING
[NO] IMPORT-POLICY <policyID>	IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **no import 200** RIP: Object successfully set IMPORT-POLICY 200: Object successfully deleted IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **show config** CREATE VPN 50 CREATE VR 1.1.1.1 ROUTER RIP COMMIT # RIP VI 1 COMMIT # VI 1 VI 2 ROUTER RIP COMMIT # VI 2 COMMIT # VR 1.1.1.1 ROUTING

Default Value

None

Command Level

rip

Related Commands

bgp rip
export-policy
import-policy

show

Use the **show** command to display RIP configuration, network neighbors, or current routing statistics.

Syntax	*Example*
SHOW {CONFIGURATION | NEIGHBORS | STATISTICS}	IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **show configuration** CREATE VPN 50 CREATE VR 1.1.1.1 2 ROUTER RIP COMMIT # RIP VI 1 COMMIT # VI 1 VI 2 COMMIT # VI 2 COMMIT # VR 1.1.1.1 ROUTING
SHOW {CONFIGURATION | NEIGHBORS | STATISTICS}	IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# **show neighbor** Targets: 192.168.24.1 -> 224.0.0.9 Interface: 192.168.24.1 VI2
SHOW {CONFIGURATION | NEIGHBORS | STATISTICS}	IPNOS: CONFIG VPN(50) VR(1.1.1.1) RIP# show statistics Route Changes: 0 VI: 2 Bad Packets: 0 Bad Routes: 0 Triggered Updates: 0

Default Value

None

Command Level

rip

Related Commands

None

CHAPTER

21 VI Commands

Use these commands to configure a virtual interface (VI). This chapter describes the following commands:

Command	*Page*
clear	21-3
connected-to	21-7
crypto	21-10
display	21-11
event-log	21-16
gre	21-19
checksum	21-20
display	21-21
key-control	21-22
key-value	21-23
sequence	21-24
ip	21-26
isis	21-37
packet-filter	21-53
qos	21-57
display	21-57
ds-byte-filter	21-58
rate-limit	21-61
remote	21-63
router	21-66

Command	*Page*
show	21-68
tunnel-keepalive	21-69
count	21-69
display	21-70
fails	21-71
frequency	21-72
target	21-73
timeout	21-73

clear

Reset statistics. Table 21-1 describes the parameters available.

Table 21-1 VI Clear Command Parameters

Parameter	*Description*
Counter	Clear VI statistics.
IPSec-stats	Clear IPSec statistics.
IKE-stats	Clear IKE statistics.

Syntax	*Example*
CLEAR COUNTER	IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)# **clear counter** COUNTER: Vi 500 - Counter successfully cleared IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)# **display** VPN : 210 VR : 192.168.120.1 VI : 500 ADDRESS : 172.16.40.2|255.255.255.252 REMOTE-ADDRESS : 0.0.0.0 BROADCAST : 1 MTU : 1500 EVENT-LOG AREA : IPSEC EVENT-LOG LEVEL : MAJOR RATE-LIMIT : DISABLE CONNECTED-TO : TUNNEL 192.168.120.1/192.168.110.1 GRE : DISABLE CRYPTO MAP : map400 STATE : UP IS-IS Config: IS-IS not enabled PACKET-FILTER(s): List is empty DS-BYTE-FILTER(s): List is empty VI Statistics: Received Octets 0 Received Unicast Packets 0 Received Non-Unicast Packets 0 Received Discarded Packets 0 Received Errors 0 Received Unknown Protocol 0 Transmitted Octets 0 Transmitted Unicast Packets 0 Transmitted Non-Unicast Packets 0 Transmitted Discarded Packets 0 Transmitted Errors 0 ...

Syntax	Example
CLEAR IPSEC-STATS	IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)# **clear ipsec-stats** IPSEC-STATS: Vi 500 - Counter successfully cleared IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)# **dis ipsec-stats** IPSEC Statistics: rcv ah pkt dropped 0 rcv ah pkt bad ah 0 rcv ah pkt bad spi 0 rcv esp pkt dropped 0 rcv esp pkt bad esp 0 rcv good pkt 0 snd pkt denied 0 esp total inbound SAs 0 esp total inbound SAs failed 0 esp current inbound SAs 0 esp total outbound SAs 0 esp total outbound SAs failed 0 esp current outbound SAs 0 ah total inbound SAs 0 ah total inbound SAs failed 0 ah current inbound SAs 0 ah total outbound SAs 0 ah total outbound SAs failed 0 ah current outbound SAs 0 esp crypto cep establish ok 0 esp crypto cep establish failed 0 esp good encrypt pkt 0 esp good decrypt pkt 0 ah good encrypt pkt 0 ah good decrypt pkt 0 esp bad encrypt pkt 0 esp bad decrypt pkt 0 ah bad encrypt pkt 0

Syntax	*Example*
CLEAR IKE-STATS	IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)# **clear ike-stats** IKE-STATS: Vi 500 - Counter successfully cleared IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)# **display ike-stats** IKE Statistics: Phase 1 Negotiations (SUCCESSFUL) : 0 Negotiations (FAILED) : 0 Packets In: 0 Packets Rejected: 0 Packets Out: 0 Phase 2 Negotiations (SUCCESSFUL) : 0 Negotiations (FAILED) : 0 Packets In: 0 Packets Rejected: 0 Packets Out: 0

Default Value

None

Command Level

vi

Related Commands

None

connected-to

Connect the VI to another interface. This command is identical to the **remote** command. Table 21-2 describes the four types of connections.

Table 21-2 VI Connection Type

Connection Type	*Description*
Bundle	Link to a MLPPP bundle. Requires a blade slot number and the bundle ID.
IP-Tunnel	IP in IP tunnel or GRE tunnel. If GRE is enabled, then a GRE tunnel is established. Refer to "gre" on page 21-19 for more information. If GRE is disabled, then an RFC 1853 IP in IP tunnel is established. In an IP in IP tunnel, each IP packet has a series of IP headers stacked on top of the original IP header.The original header is called the Inner IP Header. The "endpoints" of the tunnel are stored in the Outer IP Header. The tunnel headers are any other headers for the path, such as security headers specific to the tunnel configuration.
Logical Link	Link to a subinterface such as a Frame Relay DLCI or an ATM VCI. Requires the slot, port, channel, and sub-interface.
Loopback	Loopback signals to itself. Use to test the VI.
Tunnel	IPSec tunnel. Requires two IP addresses. *Note:* Assign an IPSec crypto map to the VI before committing the tunnel.
Virtual Interface	Link two VIs together. Requires the VPN, VR, and VI names.

Syntax	*Example*

Disconnect VIs:
`CONNECTED-TO DISCONNECTED`
	`IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(2)#` **`connect disconnect`**
	
	`IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(2)#` **`commit`**
	`Commit configuration`
	`VI(2): Object successfully set`
	
	`IPNOS: CONFIG VPN(50) VR(2.2.2.2)#` **`vi 2`**
	`VR(2.2.2.2): Object successfully set`
	
	`IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(2)#` **`display connect`**
	`CONNECTED-TO  : DISCONNECTED`

Syntax	*Example*
Link to another VI: CONNECTED-TO VIRTUAL <vpnId>/<vrId>/<viId>	IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(2)# **connect virtual 50/1.1.1.1/2** IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(2)# **commit** Commit configuration VI(2): Object successfully set IPNOS: CONFIG VPN(50) VR(2.2.2.2)# **vi 2** VR(2.2.2.2): Object successfully set IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(2)# **display** connect CONNECTED-TO : VIRTUAL 50/1.1.1.1/2
Link to subinterface: CONNECTED-TO LOGICAL <slot>/<port>:<chan>:<subif>	IPNOS : CONFIG VPN(50) VR(192.128.32.1) VI(1)# **connected-to logical 2/1:0:1** IPNOS : CONFIG VPN(50) VR(192.128.32.1) VI(1)# **display connected-to** CONNECTED-TO : LOGICAL 2/1:0:1
Establish IPSec tunnel: CONNECTED-TO TUNNEL <addressA>/<addressB>	IPNOS : CONFIG VPN(1) VR(192.128.32.1) VI(1)# **connected-to tunnel 2.2.2.2/3.3.3.3** IPNOS : CONFIG VPN(50) VR(192.128.32.1) VI(1)# **display connected-to** CONNECTED-TO : TUNNEL 2.2.2.2/3.3.3.3
Link to MLPPP bundle: CONNECTED-TO BUNDLE <slot>/<bundle-id>	IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(2)# **connect bundle 2/1** IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(2)# **display** connect CONNECTED-TO : BUNDLE 2/1
Establish loopback interface: CONNECTED-TO LOOPBACK	IPNOS : CONFIG VPN(50) VR(2.2.2.2) VI(2)# **connected-to loopback** IPNOS : CONFIG VPN(50) VR(2.2.2.2) VI(2)# **display connected-to** CONNECTED-TO : LOOPBACK

Syntax	*Example*

Establish GRE tunnel:
`CONNECTED-TO IP-TUNNEL <src-addr>/<dest-addr>`
	`IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(2)#` **`connect ip-tunnel 10.10.10.2/10.10.10.1`**
	
	`IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(2)#` **`display`**
	`VPN : 50`
	`VR : 2.2.2.2`
	`VI : 2`
	`ADDRESS : 10.10.10.2|255.255.255.0`
	`REMOTE-ADDRESS : 0.0.0.0`
	`BROADCAST : 1`
	`MTU : 1500`
	`EVENT-LOG AREA : No crypto map applied.`
	`RATE-LIMIT : DISABLE`
	`CONNECTED-TO : IP-TUNNEL 10.10.10.2/10.10.10.1`
	`GRE : ENABLE`
	
	`...`
	
	`GRE Tunnel Statistics:`
	`Key errors 0`
	`Sequence errors 0`
	`Unknown protocols 0`
	`Header errors 0`

Default Value

Disconnected

Command Level

vi

Related Command

remote

crypto

Assign an Internet Protocol Security (IPSec) crypto map to the VI. See "Crypto Commands" in Chapter 17 for more information about IPSec and crypto maps. Use the **no** option to unassign the crypto map from the VI. You must connect the tunnel immediately after assigning the crypto map.

NOTE: Set the IPNOS system time before applying a crypto map to a VI. The correct time and date are needed for certificate time stamps and for IKE and IPSec rekey timers. Refer to "time" on page 3-60 for more information.

Syntax	*Example*
[NO] CRYPTO MAP <name>	IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)# **crypto map map400** IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)# **display crypto** CRYPTO MAP : map400
[NO] CRYPTO MAP <name>	IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)# **no crypto map** IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)# **display crypto** CRYPTO MAP :

Default Value

None

Command Level

vi

Related Command

crypto map

display

Use the **display** command to view the current configuration of the VI. You can also limit the display to only the items you want to see.

Syntax	*Example*
DISPLAY	IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(2)# **display** VPN : 50 VR : 2.2.2.2 VI : 2 ADDRESS : 192.168.24.2|255.255.255.0 REMOTE-ADDRESS : 192.168.24.1 BROADCAST : 1 MTU : 1500 EVENT-LOG AREA : No crypto map applied. RATE-LIMIT : DISABLE CONNECTED-TO : VIRTUAL 50/1.1.1.1/2 GRE : DISABLE CRYPTO MAP : STATE : UP IS-IS Config: IS-IS not enabled PACKET-FILTER(s): List is empty DS-BYTE-FILTER(s): List is empty VI Statistics: Received Octets 104 Received Unicast Packets 2 Received Non-Unicast Packets 0 Received Discarded Packets 0 Received Errors 0 Received Unknown Protocol 0 Transmitted Octets 112 Transmitted Unicast Packets 2 Transmitted Non-Unicast Packets 0 Transmitted Discarded Packets 0 Transmitted Errors 0

Syntax	Example
With rate-limit: DISPLAY	IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(2)# **display** VPN : 50 VR : 2.2.2.2 VI : 2 ADDRESS : 192.168.24.2|255.255.255.0 REMOTE-ADDRESS : 192.168.24.1 BROADCAST : 1 MTU : 1500 EVENT-LOG AREA : No crypto map applied. RATE-LIMIT : 10000 bytes per seconds with 1500 bytes burst size CONNECTED-TO : VIRTUAL 50/1.1.1.1/2 GRE : DISABLE CRYPTO MAP : STATE : UP IS-IS Config: IS-IS not enabled PACKET-FILTER(s): List is empty DS-BYTE-FILTER(s): List is empty VI Statistics: Received Octets 104 Received Unicast Packets 2 Received Non-Unicast Packets 0 Received Discarded Packets 0 Received Errors 0 Received Unknown Protocol 0 Transmitted Octets 112 Transmitted Unicast Packets 2 Transmitted Non-Unicast Packets 0 Transmitted Discarded Packets 0 Transmitted Errors 0 Rate-Limited VI Statistics: Transmit Bytes Submitted 0 Transmit Packets Submitted 0 Transmit Bytes Dropped 0 Transmit Packets Dropped 0 Receive Bytes Submitted 0 Receive Packets Submitted 0 Receive Bytes Dropped 0 Receive Packets Dropped 0 Time Elapsed Since Activation 10

Syntax	Example
With crypto map: DISPLAY	IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)# **display** VPN : 210 VR : 192.168.120.1 VI : 500 ADDRESS : 172.16.40.2|255.255.255.252 REMOTE-ADDRESS : 0.0.0.0 BROADCAST : 1 MTU : 1500 EVENT-LOG AREA : IPSEC EVENT-LOG LEVEL : MAJOR RATE-LIMIT : DISABLE CONNECTED-TO : TUNNEL 192.168.120.1/192.168.110.1 GRE : DISABLE CRYPTO MAP : map400 STATE : UP IS-IS Config: IS-IS not enabled PACKET-FILTER(s): List is empty DS-BYTE-FILTER(s): List is empty VI Statistics: Received Octets 0 Received Unicast Packets 0 Received Non-Unicast Packets 0 Received Discarded Packets 0 Received Errors 0 Received Unknown Protocol 0 Transmitted Octets 0 Transmitted Unicast Packets 0 Transmitted Non-Unicast Packets 0 Transmitted Discarded Packets 0 IPSEC Statistics: rcv ah pkt dropped 0 rcv ah pkt bad ah 0 rcv ah pkt bad spi 0 rcv esp pkt dropped 0 rcv esp pkt bad esp 0

Syntax	Example
DISPLAY (continued)	

```
snd pkt denied                    0
esp total inbound SAs             0
esp total inbound SAs failed      0
esp current inbound SAs           0
esp total outbound SAs            0
esp total outbound SAs failed     0
esp current outbound SAs          0
ah total inbound SAs              0
ah total inbound SAs failed       0
ah current inbound SAs            0
ah total outbound SAs             0
ah total outbound SAs failed      0
ah current outbound SAs           0
esp crypto cep establish ok       0
esp crypto cep establish failed   0
esp good encrypt pkt              0
esp good decrypt pkt              0
ah good encrypt pkt               0
ah good decrypt pkt               0
esp bad encrypt pkt               0
esp bad decrypt pkt               0
ah bad encrypt pkt                0
ah bad decrypt pkt                0

IKE Statistics:
Phase 1
Negotiations (SUCCESSFUL) :       0
Negotiations (FAILED) :           0
Packets In:                       0
Packets Rejected:                 0
Packets Out:                      0

Phase 2
Negotiations (SUCCESSFUL) :       0
Negotiations (FAILED) :           0
Packets In:                       0
Packets Rejected:                 0
Packets Out:                      0
```

Default Value

None

Command Level

vi

Related Command

show

event-log

Enable viewing of IPSec and IKE events. This command only works when an active IPSec tunnel is running on the VI. These messages are also sent to the syslogd server. See "syslogd" on page 16-151. There are two major parameters: area and level. Table 21-3 describes event-log areas.

Table 21-3 IPSec Event Message Areas

Area	*Description*
All	Shows all possible log messages.
Apps	Shows application messages, such as firewall, RADIUS, or PPP.
Auth	Shows authentication-related messages.
Configuration	Shows configuration messages, such as creating a VR.
Emergency	Shows emergency messages, such as the loss of a control blade.
IPSec	Shows IPSec and IKE specific messages. This is the default value.
Link	Shows link layer messages, such as PPP, Frame Relay, or ATM.
Protocol	Shows protocol-related messages, except for IPSec, IKE, or link layers.
System	Shows general system messages.
Trap	Shows SNMP trap messages.

Table 21-4 describes event-log levels.

Table 21-4 IPSec Event Message Severity Levels

Level	*Description*
Critical	Critical system problems that affect normal IPSX operations, such as overheating, or PPP fails to negotiate any connection. This is the highest level. Only critical level messages will be displayed.
Major	Major problems, such as tunnels down. This is the default level.
Minor	Minor errors, such as transform set not found.
Inform	General information, such as IPSec tunnel is up or down.
Debug	Debug codes, such as programming messages. This is the lowest level. All levels of messages will be displayed. Note: IPSX switch system performance will drop dramatically because of all the log messages it has to display at this level.

Syntax	*Example*
[NO] EVENT-LOG AREA {APPS | AUTH | CONFIGURATION | EMERGENCY | IPSEC | LINK | PROTOCOL | SYSTEM | TRAP }	IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)# **event-log area all** IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)# **display event** EVENT-LOG AREA : APPS AUTH CONFIGURATION EMERGENCY IPSEC LINK PROTOCOL SYSTEM TRAP EVENT-LOG LEVEL : MAJOR
[NO] EVENT-LOG AREA {APPS | AUTH | CONFIGURATION | EMERGENCY | IPSEC | LINK | PROTOCOL | SYSTEM | TRAP }	IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)# **no event-log area all** IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)# **display event** EVENT-LOG AREA : NONE EVENT-LOG LEVEL : MAJOR
[NO] EVENT-LOG AREA {APPS | AUTH | CONFIGURATION | EMERGENCY | IPSEC | LINK | PROTOCOL | SYSTEM | TRAP }	IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)# **event-log area ipsec** IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)# **display event** EVENT-LOG AREA : IPSEC EVENT-LOG LEVEL : MAJOR
[NO] EVENT-LOG AREA {APPS | AUTH | CONFIGURATION | EMERGENCY | IPSEC | LINK | PROTOCOL | SYSTEM | TRAP }	IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)# **event-log area link** IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)# **display event** EVENT-LOG AREA : IPSEC LINK EVENT-LOG LEVEL : MAJOR
[NO] EVENT-LOG LEVEL [CRITICAL | DEBUG | INFORM | MAJOR | MINOR]	IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)# **event-log level inform** IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)# **display event** EVENT-LOG AREA : IPSEC EVENT-LOG LEVEL : INFORM
[NO] EVENT-LOG LEVEL [CRITICAL | DEBUG | INFORM | MAJOR | MINOR]	IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)# **no event-log level** IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)# **display event** EVENT-LOG AREA : IPSEC EVENT-LOG LEVEL : MAJOR

Default Value

None

Command Level

vi

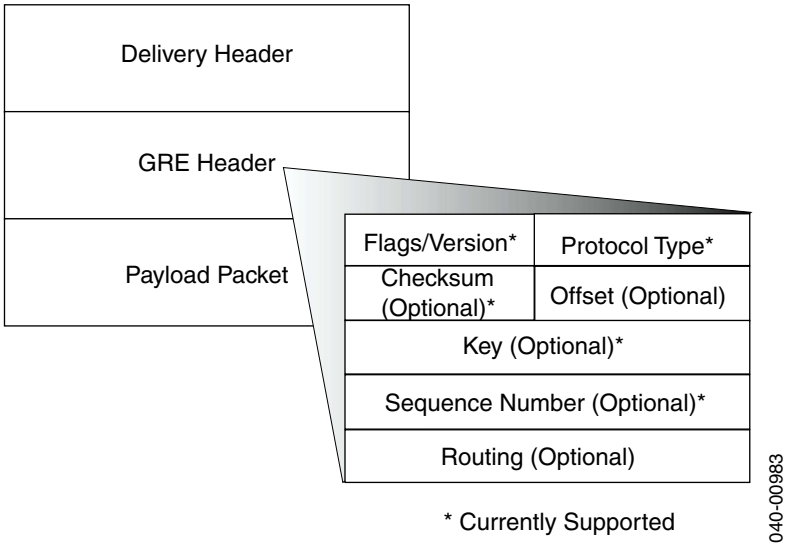
Related Command

syslogd

gre

Go to the Generic Routing Encapsulation (GRE) command level. GRE is a simple method of encapsulating arbitrary packets within an arbitrary transport protocol. A GRE "tunnel" is actually a logical term for transporting GRE encapsulated packets between a GRE source and destination. RFCs 1701 and 1702 are supported. Figure 21-1 describes the GRE packet header in detail.

Figure 21-1 GRE Packet Header



Once GRE tunneling is enabled, you need to connect the VI to another GRE tunnel endpoint. See "connected-to" on page 21-7 and "remote" on page 21-63. The following commands are available.

- "checksum" on page 21-20
- "display" on page 21-21
- "key-control" on page 21-22
- "key-value" on page 21-23
- "sequence" on page 21-24

Syntax	*Example*
[NO] GRE	IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3)# **gre** VI(3): Object successfully set GRE VI(3): Object successfully set IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3) GRE#
[NO] GRE	IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3)# **no gre** VI(3): Object successfully set GRE VI(3): Disabled successfully IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3)#

Default Value

Disabled

Command Level

vi

Related Commands

None

checksum

Enable the checksum option in the GRE header. The checksum field contains the IP (one's complement) checksum of the GRE header and the payload packet. It is used to verify that the GRE packet has no errors.

Syntax	Example
[NO] CHECKSUM	IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3) GRE# **checksum** IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3) GRE# **display checksum** CHECKSUM : ENABLE IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3) GRE# **commit** Commit configuration GRE: Object successfully set
[NO] CHECKSUM	IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3) GRE# **no checksum** IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3) GRE# **display checksum** CHECKSUM : DISABLE IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3) GRE# **commit** Commit configuration GRE: Object successfully set

Default Value

Disabled

Command Level

gre

Related Commands

None

display

View the current configuration of the GRE encapsulation. It also shows the GRE tunnel statistics.

Syntax	*Example*
DISPLAY	IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3) GRE# display GRE : ENABLE KEY-CONTROL : DISABLE SEQUENCE : DISABLE CHECKSUM : DISABLE GRE Tunnel Statistics: Key errors 0 Sequence errors 0 Unknown protocols 0 Header errors 0

Default Value

None

Command Level

gre

Related Commands

None

key-control

Enable the key option in the GRE header. See “key-value” on page 21-23 for more information.

Syntax	*Example*
[NO] KEY-CONTROL	IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3) GRE# **key-control** IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3) GRE# **display key** KEY-CONTROL : ENABLE KEY-VALUE : 0 IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3) GRE# **commit** Commit configuration GRE: Object successfully set
[NO] KEY-CONTROL	IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3) GRE# **no key-control** IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3) GRE# **display key** KEY-CONTROL : DISABLE IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3) GRE# **commit** Commit configuration GRE: Object successfully set

Default Value

Disabled

Command Level

gre

Related Command

key-value

key-value

Set the value of the key field in the GRE header. The key field contains a four octet number inserted by the encapsulator. It is used by the receiver to authenticate the source of the packet. Both the sender and the receiver must have the same keys. Choose from 0 to 4,294,967,295.

Use the **key-control** command to enable **key-value** authentication. Refer to "key-control" on page 21-22 for more information.

Syntax	*Example*
KEY-VALUE <value>	IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3) GRE# **key-value 123456** IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3) GRE# **display key** KEY-CONTROL : ENABLE KEY-VALUE : 123456 IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3) GRE# **commit** Commit configuration GRE: Object successfully set

Default Value

0

Command Level

gre

Related Command

key-control

sequence

Enable the sequence option in the GRE header. The sequence number field contains an unsigned 32-bit integer which is inserted by the encapsulator. Use sequence numbers to establish the packet order.

Syntax	Example
[NO] SEQUENCE	IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3) GRE# **sequence** IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3) GRE# **display sequence** SEQUENCE : ENABLE IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3) GRE# **commit** Commit configuration GRE: Object successfully set
[NO] SEQUENCE	IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3) GRE# **no sequence** IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3) GRE# **display sequence** SEQUENCE : DISABLE IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(3) GRE# **commit** Commit configuration GRE: Object successfully set

Default Value

Disable

Command Level

gre

Related Commands

None

ip

Configure several Internet Protocol (IP) parameters. Use the **no** option to reset the command to its default value. Table 21-5 describes the parameters available.

Table 21-5 Virtual Interface IP Parameters

Parameter	*Description*
Address	Sets the IP address and network mask. The default values are 0.0.0.0 | 0.0.0.0
Broadcast	Sets the broadcast to all ones or all zeros. The default value is 1.
MTU	Sets the maximum transmittal unit (MTU). This is the largest possible unit of data that can be transmitted. The default value is 1500. Use values from 20 to 1,500.
OSPF	Sets the Open Shortest Path First (OSPF) parameters. The following is a list of options: • Area—Set up an OSPF area on this VI. *Note:* Commit this command before using the other **ip ospf** command settings. • Authentication-key—Simple character-based password authentication key. • Cost—Set the metric to determine which route or link is the best path to a destination. Choose from 1-65,535. • Dead-Interval—Length of time before a router is declared dead. Based on how long a router's Hello packets have not been seen. Default is 40 seconds. • Hello-Interval—Length of time between sending Hello packets. Default is 10 seconds. Choose from 1 to 65,535 seconds. • Priority—Priority of the VI for designated router status. Default value is 1. Choose from 0 to 255. • Retransmit-Interval—Length of time between retransmitting LSA packets. Default is 5 seconds. Choose from 1 to 65,535 seconds. • Transmit-delay—Estimated time to transmit a link-state update packet over the VI. Default is 1 second. Choose from 1 to 65,535 seconds. • Message-digest-key—Set MD5 key identifier and password.
Unnumbered	Uses an unnumbered IP address.
Remote-address	Set a remote IP address. It can only be configured with a local address with a prefix length of 32.

Table 21-5 Virtual Interface IP Parameters (Continued)

Parameter	*Description*
No	Reset the command to its default value. The following is a list of default values: • no ip address—Reset IP address to 0.0.0.0 | 0.0.0.0. • no ip broadcast—Reset IP broadcast to 1. • no ip mtu—Reset MTU to 1500. • no ip ospf—Deactivate OSPF. • no ip ospf authentication-key—Clear the key. • no ip ospf cost—Reset the metric to 10. • no ip ospf dead-interval—Set to 40 seconds. • no ip ospf hello-interval—Set to 10 seconds. • no ip ospf priority—Set to 1. • no ip ospf retransmit-interval—Sets to 5 seconds. • no ip ospf transmit-delay—Set to 1 second. • no ip unnumbered—Reset the IP address to 0.0.0.0 | 0.0.0.0. • no ip remote-address—Remove remote IP address.

Syntax	*Example*

`[NO] IP ADDRESS <address> <netmask>`	`IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# ip address 104.168.32.1 255.255.255.0`
	
	`IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# display address`
	`ADDRESS : 104.168.32.1|255.255.255.0`

`[NO] IP ADDRESS <address> <netmask>`	`IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# no ip address 104.168.32.1 255.255.255.0`
	
	`IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# display address`
	`ADDRESS : 0.0.0.0|0.0.0.0`

`IP BROADCAST {1 | 0}`	`IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# ip broadcast 0`
	
	`IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# display broad`
	`BROADCAST : 0`

`NO IP BROADCAST`	`IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# no ip broadcast`
	
	`IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# display broad`
	`BROADCAST : 1`

Syntax	*Example*
IP MTU <size> -or- NO IP MTU	IPNOS : CONFIG VPN(1) VR(1.1.1.1) VI(1)# **ip mtu 1500**
[NO] IP OSPF AREA [areaId]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **ip ospf area 1** VI(2): Object successfully set AREA 1: Object successfully set IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **show conf** CREATE VI 2 IP ADDRESS 104.168.32.1 255.255.255.0 COMMIT # VI 2 VI 2 IP OSPF AREA 0.0.0.1 IP OSPF PRIORITY 128 COMMIT # VI 2 VI 2 CONNECTED-TO DISCONNECTED GRE COMMIT # GRE COMMIT # VI 2 VI 2 COMMIT # VI 2
[NO] IP OSPF AUTHENTICATION-KEY [key]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **ip ospf authentication-key 0123456** VI(2): Object successfully set AUTHENTICATION-KEY 0123456: Object successfully set IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **show config** CREATE VI 2 IP ADDRESS 104.168.32.1 255.255.255.0 COMMIT # VI 2 VI 2 IP OSPF AREA 0.0.0.1 IP OSPF PRIORITY 128 COMMIT # VI 2 VI 2 CONNECTED-TO DISCONNECTED GRE COMMIT # GRE COMMIT # VI 2 VI 2 COMMIT # VI 2

Syntax	Example
[NO] IP OSPF AUTHENTICATION-KEY [key]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **no ip ospf authentication-key** VI(2): Object successfully set AUTHENTICATION-KEY: Reset successfully IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **show conf** CREATE VI 2 IP ADDRESS 104.168.32.1 255.255.255.0 COMMIT # VI 2 VI 2 IP OSPF AREA 0.0.0.1 IP OSPF PRIORITY 128 COMMIT # VI 2 VI 2 CONNECTED-TO DISCONNECTED GRE COMMIT # GRE COMMIT # VI 2 VI 2 COMMIT # VI 2
[NO] IP OSPF COST {cost}	IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **ip ospf cost 4** VI(2): Object successfully set COST 4: Object successfully set IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **show conf** CREATE VI 2 IP ADDRESS 104.168.32.1 255.255.255.0 COMMIT # VI 2 VI 2 IP OSPF AREA 0.0.0.1 IP OSPF PRIORITY 128 IP OSPF COST 4 COMMIT # VI 2 VI 2 CONNECTED-TO DISCONNECTED GRE COMMIT # GRE COMMIT # VI 2 VI 2 COMMIT # VI 2

Syntax	*Example*
[NO] IP OSPF COST {cost}	IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **no ip ospf cost** VI(2): Object successfully set COST: Reset successfully IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **show conf** CREATE VI 2 IP ADDRESS 104.168.32.1 255.255.255.0 COMMIT # VI 2 VI 2 IP OSPF AREA 0.0.0.1 IP OSPF PRIORITY 128 COMMIT # VI 2 VI 2 CONNECTED-TO DISCONNECTED GRE COMMIT # GRE COMMIT # VI 2 VI 2 COMMIT # VI 2
[NO] IP OSPF DEAD-INTERVAL [value]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# ip ospf dead-interval 45 VI(2): Object successfully set DEAD-INTERVAL 45: Object successfully set IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# show conf CREATE VI 2 IP ADDRESS 104.168.32.1 255.255.255.0 COMMIT # VI 2 VI 2 IP OSPF AREA 0.0.0.1 IP OSPF PRIORITY 128 IP OSPF DEAD-INTERVAL 45 COMMIT # VI 2 VI 2 CONNECTED-TO DISCONNECTED GRE COMMIT # GRE COMMIT # VI 2 VI 2 COMMIT # VI 2

Syntax	Example
[NO] IP OSPF DEAD-INTERVAL [value]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **no ip ospf dead-interval** VI(2): Object successfully set DEAD-INTERVAL: Reset successfully IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **show conf** CREATE VI 2 IP ADDRESS 104.168.32.1 255.255.255.0 COMMIT # VI 2 VI 2 IP OSPF AREA 0.0.0.1 IP OSPF PRIORITY 128 COMMIT # VI 2 VI 2 CONNECTED-TO DISCONNECTED GRE COMMIT # GRE COMMIT # VI 2 VI 2 COMMIT # VI 2 IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)#
[NO] IP OSPF HELLO-INTERVAL [value]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **ip ospf hello-interval 5** VI(2): Object successfully set HELLO-INTERVAL 5: Object successfully set IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **show conf** CREATE VI 2 IP ADDRESS 104.168.32.1 255.255.255.0 COMMIT # VI 2 VI 2 IP OSPF AREA 0.0.0.1 IP OSPF PRIORITY 128 IP OSPF HELLO-INTERVAL 5 COMMIT # VI 2 VI 2 CONNECTED-TO DISCONNECTED GRE COMMIT # GRE COMMIT # VI 2 VI 2 COMMIT # VI 2

Syntax	*Example*
[NO] IP OSPF HELLO-INTERVAL [value]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **no ip ospf hello-interval** VI(2): Object successfully set HELLO-INTERVAL: Reset successfully IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **show conf** CREATE VI 2 IP ADDRESS 104.168.32.1 255.255.255.0 COMMIT # VI 2 VI 2 IP OSPF AREA 0.0.0.1 IP OSPF PRIORITY 128 COMMIT # VI 2 VI 2 CONNECTED-TO DISCONNECTED GRE COMMIT # GRE COMMIT # VI 2 VI 2 COMMIT # VI 2
[NO] IP OSPF PRIORITY [value]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **ip ospf priority 110** VI(2): Object successfully set PRIORITY 110: Object successfully set IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **show conf** CREATE VI 2 IP ADDRESS 104.168.32.1 255.255.255.0 COMMIT # VI 2 VI 2 IP OSPF AREA 0.0.0.1 IP OSPF PRIORITY 110 COMMIT # VI 2 VI 2 CONNECTED-TO DISCONNECTED GRE COMMIT # GRE COMMIT # VI 2 VI 2 COMMIT # VI 2

Syntax	Example
[NO] IP OSPF PRIORITY [value]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **no ip ospf priority** VI(2): Object successfully set PRIORITY: Reset successfully IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **show conf** CREATE VI 2 IP ADDRESS 104.168.32.1 255.255.255.0 COMMIT # VI 2 VI 2 IP OSPF AREA 0.0.0.1 COMMIT # VI 2 VI 2 CONNECTED-TO DISCONNECTED GRE COMMIT # GRE COMMIT # VI 2 VI 2 COMMIT # VI 2
[NO] IP OSPF RETRANSMIT-INTERVAL [value]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **ip ospf retransmit-interval 6** VI(2): Object successfully set RETRANSMIT-INTERVAL 6: Object successfully set IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **show conf** CREATE VI 2 IP ADDRESS 104.168.32.1 255.255.255.0 COMMIT # VI 2 VI 2 IP OSPF AREA 0.0.0.1 IP OSPF RETRANSMIT-INTERVAL 6 COMMIT # VI 2 VI 2 CONNECTED-TO DISCONNECTED GRE COMMIT # GRE COMMIT # VI 2 VI 2 COMMIT # VI 2

Syntax	*Example*
[NO] IP OSPF RETRANSMIT-INTERVAL [value]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **no ip ospf retransmit-interval** VI(2): Object successfully set RETRANSMIT-INTERVAL: Reset successfully IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **show conf** CREATE VI 2 IP ADDRESS 104.168.32.1 255.255.255.0 COMMIT # VI 2 VI 2 IP OSPF ARE.0.0.1 COMMIT # VI 2 VI 2 CONNECTED-TO DISCONNECTED GRE COMMIT # GRE COMMIT # VI 2 VI 2 COMMIT # VI 2
[NO] IP OSPF TRANSMIT-DELAY [value]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **ip ospf transmit-delay 3** VI(2): Object successfully set TRANSMIT-DELAY 3: Object successfully set IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# CREATE VI 2 IP ADDRESS 104.168.32.1 255.255.255.0 COMMIT # VI 2 VI 2 IP OSPF AREA 0.0.0.1 IP OSPF TRANSMIT-DELAY 3 COMMIT # VI 2 VI 2 CONNECTED-TO DISCONNECTED GRE COMMIT # GRE COMMIT # VI 2 VI 2 COMMIT # VI 2

Syntax	Example
[NO] IP OSPF TRANSMIT-DELAY [value]	IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **no ip ospf transmit-delay** VI(2): Object successfully set TRANSMIT-DELAY: Reset successfully IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **show conf** CREATE VI 2 IP ADDRESS 104.168.32.1 255.255.255.0 COMMIT # VI 2 VI 2 IP OSPF AREA 0.0.0.1 COMMIT # VI 2 VI 2 CONNECTED-TO DISCONNECTED GRE COMMIT # GRE COMMIT # VI 2 VI 2 COMMIT # VI 2
[NO] IP UNNUMBERED	IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **ip unnumbered** IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **display address** ADDRESS : UNNUMBERED
[NO] IP UNNUMBERED	IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **no ip unnumbered** IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **display address** ADDRESS : 0.0.0.0|0.0.0.0
[NO] IP REMOTE-ADDRESS address	IPNOS: CONFIG VPN(50) VR(7.7.7.7) VI(2)# **ip remote-address 172.16.24.1** Remote address can be configured only for local address with prefixlen 32!

Default Value

See Table 21-5.

Command Level

vi

Related Commands

router ospf
export-policy metric

isis

Configure an IS-IS connection. Table 21-6 describes the parameters.

Table 21-6 IS-IS Connection Parameters

Parameter	*Description*
Circuit-type	Adjacency types for the circuit. Choose from the following levels: • Level 1 • Level 1 and 2 This is the default value. • Level 2 only
CSNP-interval;	Interval in seconds for periodic transmission of CSNP's, based on level. Choose from 1-600 seconds, and one of the following levels: • Level 1 • Level 2 Default value is 10 seconds.
DR-hello-interval	Hello interval in seconds between level 1 or level 2 PDUs. Used if the system is the designated router. Choose from 1-65,535 milliseconds and one of the following levels: • Level 1 • Level 2 Default value is 10 seconds.
Hello-interval	Interval between ISIS hello messages. Choose from 1-65,535 seconds and one of the following levels: • Level 1 • Level 2 Default value is 10 seconds.
Hello-multiplier	Multiplier to obtain the hold-time from hello-interval. Choose from 2-100 and one of the following levels: • Level 1 • Level 2 Default value is 3.
Passive	Set ISIS protocol as passive. Default value is disabled.
Password	Password used for authenticating ISIS Hello PDU's. Use up to 256 alphanumeric characters, and one of the following levels: • Level 1 • Level 2 Default value is none.

Table 21-6 IS-IS Connection Parameters (Continued)

Parameter	*Description*
Priority	Priority for the selection of the designated router. Choose from 1-127 and one of the following levels: • Level 1 • Level 2 Default values is 64.
PSNP-interval	Minimum interval between sending partial sequence number PDUs (PSNPs). Choose from 1-120 and one of the following levels: • Level 1 • Level 2 Default value is 2.
LSP-interval	Minimum time interval between sending consecutive LSP's. Choose from 1-1000 milliseconds. Default value is 33.
LSP-throttle-interval	Minimal interval, in milliseconds, between transmissions of LSPs on a point to point interface. Choose from 1-65535 milliseconds. Default value is 33.
Metric	Metric for the traffic on the interface. Choose from 0-63 and one of the following levels: • Level 1 • Level 2 Default value is 10.
Retransmit-interval	Minimum interval, in seconds, between re-transmission of each LSP over a point to point (P2P) link. Choose from 1-300 seconds and one of the following levels: • Level 1 • Level 2 Default value is 5 seconds.

Syntax	*Example*
ISIS CIRCUIT-TYPE {level-1 | level-1-2 | level-2-only} or NO ISIS CIRCUIT-TYPE	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **isis circuit-type level-1** VI(2): Object successfully set CIRCUIT-TYPE level-1: Object successfully set IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS ISIS CIRCUIT-TYPE LEVEL-1 COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2
ISIS CIRCUIT-TYPE {level-1 | level-1-2 | level-2-only} or NO ISIS CIRCUIT-TYPE	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **isis circuit-type level-1-2** VI(2): Object successfully set CIRCUIT-TYPE level-1-2: Object successfully set IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2

Syntax	*Example*
ISIS CIRCUIT-TYPE {level-1 | level-1-2 | level-2-only} or NO ISIS CIRCUIT-TYPE	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **isis circuit-type level-2-only** VI(2): Object successfully set CIRCUIT-TYPE level-2-only: Object successfully set IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# show config CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS ISIS CIRCUIT-TYPE LEVEL-2-only COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2
ISIS CIRCUIT-TYPE {level-1 | level-1-2 | level-2-only} or NO ISIS CIRCUIT-TYPE	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# no isis circuit-type VI(2): Object successfully set CIRCUIT-TYPE: Reset successfully IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# show config CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2

Syntax	*Example*
ISIS CSNP-INTERVAL <seconds> {level-1 | level-2}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **isis csnp-interval 250 level-1** VI(2): Object successfully set CSNP-INTERVAL 250: Object successfully set IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS ISIS CSNP-INTERVAL 250 LEVEL-1 COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2
NO ISIS CSNP-INTERVAL {level-1 | level-2}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **no isis csnp-interval level-1** VI(2): Object successfully set CSNP-INTERVAL: Reset successfully IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2

Syntax	*Example*
ISIS DR-HELLO-INTERVAL <seconds> {level-1 | level-2}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **isis dr-hello-interval 2000 level-1** VI(2): Object successfully set DR-HELLO-INTERVAL 2000: Object successfully set IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS ISIS DR-HELLO-INTERVAL 2000 LEVEL-1 COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2
NO ISIS DR-HELLO-INTERVAL {level-1 | level-2}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **no isis dr-hello-interval level-1** VI(2): Object successfully set DR-HELLO-INTERVAL: Reset successfully IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2

Syntax	*Example*
ISIS HELLO-INTERVAL <seconds> {level-1 | level-2}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **isis hello-interval 100 level-1** VI(2): Object successfully set HELLO-INTERVAL 100: Object successfully set IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS ISIS HELLO-INTERVAL 100 LEVEL-1 COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2
NO ISIS HELLO-INTERVAL {level-1 | level-2}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **no isis hello-interval level-1** VI(2): Object successfully set HELLO-INTERVAL: Reset successfully IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2

Syntax	*Example*
ISIS HELLO-MULTIPLIER <multiplier> {level-1 | level-2}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **isis hello-multi 15 level-1** VI(2): Object successfully set HELLO-MULTIPLIER 15: Object successfully set IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS ISIS HELLO-MULTIPLIER 15 LEVEL-1 COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2
NO ISIS HELLO-MULTIPLIER {level-1 | level-2}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **no isis hello-multi level-1** VI(2): Object successfully set HELLO-MULTIPLIER: Reset successfully IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2

Syntax	Example
[NO] ISIS PASSIVE	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **isis passive** VI(2): Object successfully set ISIS PASSIVE: Object successfully set IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS ISIS PASSIVE COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2
[NO] ISIS PASSIVE	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **no isis passive** VI(2): Object successfully set PASSIVE: Disabled successfully IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2

Syntax	*Example*
ISIS PASSWORD <text-string> {LEVEL-1 | LEVEL-2}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **isis password 0123456 level-1** VI(2): Object successfully set PASSWORD 0123456: Object successfully set IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS ISIS PASSWORD 0123456 LEVEL-1 COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2
NO ISIS PASSWORD { level-1 | level-2 }	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **no isis password level-1** VI(2): Object successfully set PASSWORD: Reset successfully IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2

Syntax	*Example*
ISIS PRIORITY <value> {level-1 | level-2}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **isis priority 5 level-1** VI(2): Object successfully set PRIORITY 5: Object successfully set IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS ISIS PRIORITY 5 LEVEL-1 COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2
NO ISIS PRIORITY {level-1 | level-2}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **no isis priority level-1** VI(2): Object successfully set PRIORITY: Reset successfully IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2

Syntax	*Example*
ISIS PSNP-INTERVAL <seconds> {level-1 | level-2}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **isis psnp-interval 8 level-1** VI(2): Object successfully set PSNP-INTERVAL 8: Object successfully set IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS ISIS PSNP-INTERVAL 8 LEVEL-1 COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2
NO ISIS PSNP-INTERVAL {level-1 | level-2}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **no isis psnp-interval level-1** VI(2): Object successfully set PSNP-INTERVAL: Reset successfully IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2

Syntax	Example
ISIS LSP-INTERVAL <milliseconds>	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **isis lsp-interval 55** VI(2): Object successfully set LSP-INTERVAL 55: Object successfully set IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS ISIS LSP-INTERVAL 55 COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2
NO ISIS LSP-INTERVAL	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **no isis lsp-interval** VI(2): Object successfully set LSP-INTERVAL: Reset successfully IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2

Syntax	*Example*
ISIS LSP-THROTTLE-INTERVAL <milliseconds>	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# isis lsp-throttle-interval 555 VI(2): Object successfully set LSP-THROTTLE-INTERVAL 555: Object successfully set IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# show config CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS ISIS LSP-THROTTLE-INTERVAL 555 COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2
NO ISIS LSP-THROTTLE-INTERVAL	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **no isis lsp-throttle-interval** VI(2): Object successfully set LSP-THROTTLE-INTERVAL: Reset successfully IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2

Syntax	*Example*
ISIS METRIC <value> {level-1 | level-2}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **isis metric 14 level-1** VI(2): Object successfully set METRIC 14: Object successfully set IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS ISIS METRIC 14 LEVEL-1 COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2
NO ISIS METRIC {level-1 | level-2}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **no isis metric level-1** VI(2): Object successfully set METRIC: Reset successfully IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2

Syntax	*Example*
ISIS RETRANSMIT-INTERVAL <seconds> {level-1 | level-2}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **isis retransmit-interval 35 level-1** VI(2): Object successfully set RETRANSMIT-INTERVAL 35: Object successfully set IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS ISIS RETRANSMIT-INTERVAL 35 LEVEL-1 COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2
NO ISIS RETRANSMIT-INTERVAL {level-1 | level-2}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **no isis retransmit-interval level-1** VI(2): Object successfully set RETRANSMIT-INTERVAL: Reset successfully IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2

Default Value

None

Command Level

vi

Related Commands

VR router isis

packet-filter

Create a packet filter for a specific VI. A packet filter sets the rules for deciding whether to drop or forward a packet. This filter applies only to this VI. You must create a general access control list (ACL) at the VR level before referring to it. See "acl" on page 16-46. Table 21-7 describes the packet-filter parameters.

Table 21-7 VI Packet Filter Parameters

Parameter	*Description*
Permit	Permit the packet into the VI.
Deny	Denie the packet access to the VI. Drop the packet.
Firewall	Apply a firewall proxy to the packet. *Note:* You need to configure the appropriate firewall proxy.
Inbound	Affect inbound packets.
Outbound	Affect outbound packets.
GenAclId	Affect packets coming from the selected range of IP addresses stated in the ACL.
Counter	Enable a packet filter counter. Every time a packet hits a particular filter, the filter's counter increments by one. When the counter reaches 2,147,483,647 (2 billion), the counter resets to zero. To see the current count, type **display packet-filter**. If event logs are enabled, the packet filter count is also sent to the syslog daemon (syslogd) server. The default value is disabled. *WARNING:* When enabled, the counting operation slows down IPNOS. Use this parameter for debugging purposes only.
Event-log	Enable packet filter event logs. Every time a packet hits a particular filter, an event log is generated. To reduce chances of a denial of service attack (DoS), a limit of 100 packet filter event logs are generated per filter in a ten second interval. These event logs are then sent to the syslog server. Refer to "syslogd" on page 16-151 for information on enabling the syslog daemon (syslogd) server. Also, the counter is enabled (or reset to zero if it is already enabled) when packet filter event logs are enabled. The default value is disabled. *Note:* When enabled, the event log generation slows down IPNOS dramatically. Use this parameter for debugging purposes only.

Syntax	*Example*
[NO] PACKET-FILTER <filterId> {PERMIT | DENY | FIREWALL} {INBOUND | OUTBOUND} <genAclId> [COUNTER | EVENT-LOG]	IPNOS : CONFIG VPN(50) VR(3.3.3.3) VI(2)# **packet-filter 1 permit inbound 23** VR(3.3.3.3): Object successfully set PACKET-FILTER 1: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3) VI(2)# **display packet** PACKET-FILTER(s): PACKET-FILTER | ACTION | DIRECTION | ACL | EVENT-LOG | COUNTER 1 | PERMIT | INBOUND | 23 | DISABLE | 0
[NO] PACKET-FILTER <filterId> {PERMIT | DENY | FIREWALL} {INBOUND | OUTBOUND} <genAclId> [COUNTER | EVENT-LOG]	IPNOS : CONFIG VPN(50) VR(3.3.3.3) VI(2)# **packet-filter 2 deny inbound 24** VR(3.3.3.3): Object successfully set PACKET-FILTER 2: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3) VI(2)# **display packet** PACKET-FILTER(s): PACKET-FILTER | ACTION | DIRECTION | ACL | EVENT-LOG | COUNTER 1 | PERMIT | INBOUND | 23 | DISABLE | 0 2 | DENY | INBOUND | 24 | DISABLE | 0
[NO] PACKET-FILTER <filterId> {PERMIT | DENY | FIREWALL} {INBOUND | OUTBOUND} <genAclId> [COUNTER | EVENT-LOG]	IPNOS : CONFIG VPN(50) VR(3.3.3.3) VI(2)# **packet-filter 3 firewall inbound 25 counter** VR(3.3.3.3): Object successfully set PACKET-FILTER 3: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3) VI(2)# **display packet** PACKET-FILTER(s): PACKET-FILTER | ACTION | DIRECTION | ACL | EVENT-LOG | COUNTER 1 | PERMIT | INBOUND | 23 | DISABLE | 0 2 | DENY | INBOUND | 24 | DISABLE | 0 3 | FIREWALL | INBOUND | 25 | COUNTER | 0

Syntax	Example
[NO] PACKET-FILTER <filterId> {PERMIT | DENY | FIREWALL} {INBOUND | OUTBOUND} <genAclId> [COUNTER | EVENT-LOG]	IPNOS : CONFIG VPN(50) VR(3.3.3.3) VI(2)# **packet-filter 4 deny outbound 26 event-log** VR(3.3.3.3): Object successfully set PACKET-FILTER 4: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3) VI(2)# **display packet** PACKET-FILTER(s): PACKET-FILTER | ACTION | DIRECTION | ACL | EVENT-LOG | COUNTER 1 | PERMIT | INBOUND | 23 | DISABLE | 0 2 | DENY | INBOUND | 24 | DISABLE | 0 3 | FIREWALL | INBOUND | 25 | COUNTER | 0 4 | DENY | OUTBOUND | 26 | EVENT-LOG | 0
[NO] PACKET-FILTER <filterId> {PERMIT | DENY | FIREWALL} {INBOUND | OUTBOUND} <genAclId> [COUNTER | EVENT-LOG]	IPNOS : CONFIG VPN(50) VR(3.3.3.3) VI(2)# **packet-filter 5 firewall inbound 27 event-log** VR(3.3.3.3): Object successfully set PACKET-FILTER 5: Object successfully set IPNOS: CONFIG VPN(50) VR(3.3.3.3) VI(2)# **display packet** PACKET-FILTER(s): PACKET-FILTER | ACTION | DIRECTION | ACL | EVENT-LOG | COUNTER 1 | PERMIT | INBOUND | 23 | DISABLE | 0 2 | DENY | INBOUND | 24 | DISABLE | 0 3 | FIREWALL | INBOUND | 25 | COUNTER | 0 4 | DENY | OUTBOUND | 26 | EVENT-LOG | 0 5 | FIREWALL | INBOUND | 27 | EVENT-LOG | 0
[NO] PACKET-FILTER <filterId> {PERMIT | DENY | FIREWALL} {INBOUND | OUTBOUND} <genAclId> [COUNTER | EVENT-LOG]	IPNOS : CONFIG VPN(50) VR(3.3.3.3) VI(2)# **no packet-filter 5** VR(3.3.3.3): Object successfully set PACKET-FILTER 5: Object successfully deleted IPNOS: CONFIG VPN(50) VR(3.3.3.3) VI(2)# **display packet** PACKET-FILTER(s): PACKET-FILTER | ACTION | DIRECTION | ACL | EVENT-LOG | COUNTER 1 | PERMIT | INBOUND | 23 | DISABLE | 0 2 | DENY | INBOUND | 24 | DISABLE | 0 3 | FIREWALL | INBOUND | 25 | COUNTER | 0 4 | DENY | OUTBOUND | 26 | EVENT-LOG | 0

Default Value

None

Command Level

vi

Related Commands

vr packet-filter

qos

Go to the quality of service command level. The following commands are available at this level.

- "display" on page 21-57
- "ds-byte-filter" on page 21-58

Syntax	*Example*
QOS	IPNOS : CONFIG VPN(50) VR(3.3.3.3) VI(2)# **qos** QOS VI(2): Object successfully set IPNOS : CONFIG VPN(50) VR(3.3.3.3) VI(2) QOS#

Default Value

None

Command Level

VI

Related Commands

None

display

View the current status of QoS features.

Syntax	*Example*
DISPLAY	IPNOS : CONFIG VPN(50) VR(3.3.3.3) VI(2) QOS# **display** DS-BYTE-FILTER(s): DS-BYTE-FILTER | ACTION | VALUE | DIRECTION | ACL 1 | SETVALUE | 32 | OUTBOUND | 1

Default Value

None

Command Level

QoS

Related Commands

None

ds-byte-filter

Set the differentiated service (DS) bits based upon an ACL filter of packets going through a specific VI. Similar in function to the **ds-byte** and **packet-filter** commands, the **ds-byte-filter** command extends the granularity of setting a packet's DS bits. Used with ACLs, a specific class or flow of packets can be singled out to have their DS bits set to a specific value. For each entry, one of two actions must be selected: SET the DS bits field to the specified value or IGNORE the DS bits field and leave it at its original value.

If a packet satisfies the ACL associated with a **ds-byte-filter** entry, the action specified in that entry is applied to the packet. As with the **packet-filter** feature, there are two independent **ds-byte-filter** commands: one that is virtual interface specific and one that applies to the entire virtual router. The entries in the virtual interface table is applied first and if there are no matches, the entries in the virtual router table are applied. With each command, each entry is applied to a packet until its ACL is satisfied. If no entries within either the virtual interface **ds-byte-filter** table or the virtual router **ds-byte-filter** table are satisfied, the packet's ds-byte field will be set to the value specified by the Virtual Router's **ds-byte** parameter. Use the **no** option to delete a **ds-byte-filter**. Table 21-8 describes the **ds-byte-filter** parameters.

Table 21-8 ds-byte-filter Parameters

Parameter	*Description*
FilterID	ID of the filter.
Setvalue	Defines the values to be placed in the Type Of Service octet of each IP packet sent out or forwarded by the VR. The valid range covers both 7-bit configurations recommended in RFC 1349 (Precedence and TOS bits) and 6-bit configurations recommended in RFC 2474 (Differentiated Services). If the value selected is 256, the DS-Byte is not changed and packets are forwarded with their existing Type Of Service byte value. Table 21-9 describes the pre-set values used for compatibility with non-CoSine routers.
Inbound | Outbound	Direction of packets on which this filter is applied. Currently only outbound packets are filtered.
genACLId	ACL ID. All packets transmitted or received are acted upon by this filter.

Table 21-9 Possible Values for IP Precedence Compatibility

Value	*Description*
0	Routine
32	Priority
64	Immediate
96	Flash
128	Flash Override
160	Critical/ECP
192	Internet Control
224	Network Control

Syntax	*Example*
DS-BYTE-FILTER <filterId> {SETVALUE <value> | IGNORE} {INBOUND | OUTBOUND} <GenaclId>	IPNOS : CVPN(50) VR(3.3.3.3) VI(2) QOS# **ds-byte-filter 1 setvalue 32 outbound 1** DS-BYTE-FILTER 1: Object successfully set IPNOS : CONFIG VPN(50) VR(3.3.3.3) VI(2) QOS# **display** DS-BYTE-FILTER(s): DS-BYTE-FILTER | ACTION | VALUE | DIRECTION | ACL 1 | SETVALUE | 32 | OUTBOUND | 1
DS-BYTE-FILTER filterId {SETVALUE value | IGNORE} {INBOUND | OUTBOUND} <GenaclId>	IPNOS : CONFIG VPN(50) VR(3.3.3.3) VI(2) QOS# **ds-byte-filter 2 ignore outbound 1** DS-BYTE-FILTER 2: Object successfully set IPNOS : CONFIG VPN(50) VR(3.3.3.3) VI(2) QOS# **display** DS-BYTE-FILTER(s): DS-BYTE-FILTER | ACTION | VALUE | DIRECTION | ACL 1 | SETVALUE | 32 | OUTBOUND | 1 2 | IGNORE | - | OUTBOUND | 1

Syntax	*Example*
NO DS-BYTE-FILTER <filterId>	IPNOS : CONFIG VPN(50) VR(3.3.3.3) VI(2) QOS# **no ds-byte-filter 2** DS-BYTE-FILTER 2: Object successfully deleted IPNOS : CONFIG VPN(50) VR(3.3.3.3) VI(2) QOS# **display** DS-BYTE-FILTER(s): DS-BYTE-FILTER | ACTION | VALUE | DIRECTION | ACL 1 | SETVALUE | 32 | OUTBOUND | 1

Default Value

Disabled

Command Level

QOS

Related Commands

ds-byte (VR command)
ds-byte-filter (VR QOS command)
packet-filter (VR command)
packet-filter (VI command)

rate-limit

Set the speed of a connection to a certain bandwidth. These connections include VI to VI, VI to LI, and tunnels. For example, you can limit a connection to 500 kbps bandwidth when it is configured for a 1.5 Mbps T1 interface. You can set the rate or the burst size to zero, but you cannot set them both to zero at the same time.

You must connect the VI to another interface before commiting the **rate-limit.** Refer to "connected-to" on page 21-7 for more information. Use the no option to reset the command to its default value. Table 21-10 describes the parameters.

Table 21-10 VI Rate-limit Parameters

Parameter	*Description*
Rate	Long term data throughput, in bytes per second, that the VI commits to support under normal network conditions.
Burst Size	Largest burst of data, in bytes, above the committed rate that will be allowed temporarily by the VI.

Syntax	*Example*
RATE-LIMIT <rate> <burstsize>	IPNOS : CONFIG VPN(50) VR(3.3.3.3) VI(2)# **rate-limit 100000 1500** IPNOS : CONFIG VPN(50) VR(3.3.3.3) VI(2)# **display rate** RATE-LIMIT : 100000 bytes per second with 1500 bytes burst size
[NO] RATE-LIMIT	IPNOS : CONFIG VPN(50) VR(3.3.3.3) VI(2)# **no rate-limit** IPNOS : CONFIG VPN(50) VR(3.3.3.3) VI(2)# **display rate** RATE-LIMIT : not enabled

Default Value

Disabled

Command Level

vi

Related Commands

None

remote

Connect the VI to another interface. It is identical to the **connected-to** command. Table 21-11 describes the four types of connections.

Table 21-11 VI Connection Type

Connection Type	*Description*
Bundle	Link to a MLPPP bundle. Requires a blade slot number and the bundle ID.
IP-Tunnel	IP in IP tunnel or GRE tunnel. If GRE is enabled, then a GRE tunnel is established. Refer to "gre" on page 21-19 for more information. If GRE is disabled, then an RFC 1853 IP in IP tunnel is established. In an IP in IP tunnel, each IP packet has a series of IP headers stacked on top of the original IP header.The original header is called the Inner IP Header. The "endpoints" of the tunnel are stored in the Outer IP Header. The tunnel headers are any other headers for the path, such as security headers specific to the tunnel configuration.
Logical Link	Link to a subinterface such as a Frame Relay DLCI or an ATM VCI. Requires the slot, port, channel, and sub-interface.
Loopback	Loopback signals to itself. Use to test the VI.
Tunnel	IPSec tunnel. Requires two IP addresses. *Note:* Assign an IPSec crypto map to the VI before committing the tunnel.
Virtual Interface	Link two VIs together. Requires the VPN, VR, and VI names.

Syntax	*Example*

Disconnect VIs:
`REMOTE DISCONNECTED`
	`IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(2)#` **`remote disremote`**
	
	`IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(2)#` **`commit`**
	`Commit configuration`
	`VI(2): Object successfully set`
	
	`IPNOS: CONFIG VPN(50) VR(2.2.2.2)#` **`vi 2`**
	`VR(2.2.2.2): Object successfully set`
	
	`IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(2)#` **`display remote`**
	`CONNECTED-TO  : DISremote`

Syntax	*Example*
Link to another VI: REMOTE VIRTUAL <vpnId>/<vrId>/<viId>	IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(2)# **remote virtual 50/1.1.1.1/2** IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(2)# **commit** Commit configuration VI(2): Object successfully set IPNOS: CONFIG VPN(50) VR(2.2.2.2)# **vi 2** VR(2.2.2.2): Object successfully set IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(2)# **display** remote CONNECTED-TO : VIRTUAL 50/1.1.1.1/2
Link to subinterface: REMOTE LOGICAL <slot>/<port>:<chan>:<subif>	IPNOS : CONFIG VPN(50) VR(192.128.32.1) VI(1)# **CONNECTED-TO logical 2/1:0:1** IPNOS : CONFIG VPN(50) VR(192.128.32.1) VI(1)# **display CONNECTED-TO** CONNECTED-TO : LOGICAL 2/1:0:1
Establish IPSec tunnel: REMOTE TUNNEL <addressA>/<addressB>	IPNOS : CONFIG VPN(1) VR(192.128.32.1) VI(1)# **CONNECTED-TO tunnel 2.2.2.2/3.3.3.3** IPNOS : CONFIG VPN(50) VR(192.128.32.1) VI(1)# **display CONNECTED-TO** CONNECTED-TO : TUNNEL 2.2.2.2/3.3.3.3
Link to MLPPP bundle: REMOTE BUNDLE <slot>/<bundle-id>	IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(2)# **remote bundle 2/1** IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(2)# **display** remote CONNECTED-TO : BUNDLE 2/1
Establish loopback interface: REMOTE LOOPBACK	IPNOS : CONFIG VPN(50) VR(2.2.2.2) VI(2)# **CONNECTED-TO loopback** IPNOS : CONFIG VPN(50) VR(2.2.2.2) VI(2)# **display CONNECTED-TO** CONNECTED-TO : LOOPBACK

Syntax	*Example*

Establish GRE tunnel:
`REMOTE IP-TUNNEL <src-addr>/<dest-addr>`
	`IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(2)# remote ip-tunnel 10.10.10.2/10.10.10.1`
	
	`IPNOS: CONFIG VPN(50) VR(2.2.2.2) VI(2)# display`
	`VPN : 50`
	`VR : 2.2.2.2`
	`VI : 2`
	`ADDRESS : 10.10.10.2|255.255.255.0`
	`REMOTE-ADDRESS : 0.0.0.0`
	`BROADCAST : 1`
	`MTU : 1500`
	`EVENT-LOG AREA : No crypto map applied.`
	`RATE-LIMIT : DISABLE`
	`CONNECTED-TO : IP-TUNNEL 10.10.10.2/10.10.10.1`
	`GRE : ENABLE`
	
	`...`
	
	`GRE Tunnel Statistics:`
	`Key errors 0`
	`Sequence errors 0`
	`Unknown protocols 0`
	`Header errors 0`

Default Value

Disconnected

Command Level

vi

Related Commands

connected-to

router

Enable routing protocol on the VI. Remember to assign IP addresses and enable routing on both VIs. See “ip” on page 21-26. See Chapter 20, “Routing Commands” for routing protocol configuration commands.

Syntax	*Example*
[NO] ROUTER { RIP | ISIS | MPLS | RSVP}	IPNOS: CONFIG VPN(50) VR(1.1.1.1) VI(2)# **router rip** VI(2): Object successfully set RIP: Object successfully set IPNOS: CONFIG VPN(50) VR(1.1.1.1) VI(2)# show config CREATE VI 2 IP ADDRESS 192.168.24.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER RIP COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/2.2.2.2/2 COMMIT # VI 2 VI 2 COMMIT # VI 2
[NO] ROUTER { RIP | ISIS | MPLS | RSVP}	IPNOS: CONFIG VPN(50) VR(1.1.1.1) VI(2)# **no router rip** VI(2): Object successfully set RIP: Disabled successfully IPNOS: CONFIG VPN(50) VR(1.1.1.1) VI(2)# **show config** CREATE VI 2 IP ADDRESS 192.168.24.1 255.255.255.0 COMMIT # VI 2 VI 2 COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/2.2.2.2/2 COMMIT # VI 2 VI 2 COMMIT # VI 2

Syntax	Example
[NO] ROUTER { RIP | ISIS | MPLS | RSVP}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **router isis** VI(2): Object successfully set ISIS Routing: Object successfully set IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2
[NO] ROUTER { RIP | ISIS | MPLS | RSVP}	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **no router isis** VI(2): Object successfully set ISIS Routing: Disabled successfully IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# **show config** CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2

Default Value

Disabled

Command Level

vi

Related Commands

None

show

View the current VI and routing protocol configuration.

Syntax	*Example*
SHOW CONFIG	IPNOS: CONFIG VPN(50) VR(5.5.5.5) VI(2)# show config CREATE VI 2 IP ADDRESS 172.24.32.1 255.255.255.0 COMMIT # VI 2 VI 2 ROUTER ISIS COMMIT # VI 2 VI 2 CONNECTED-TO VIRTUAL 50/6.6.6.6/2 COMMIT # VI 2 VI 2 COMMIT # VI 2

Default Value

None.

Command Level

vi

Related Commands

None

tunnel-keepalive

Go to the tunnel-keepalive command level. This command polls the VI tunnel to verify that it is still active. It is only available for VIs with tunnel connections. Use the **no** option to disable the command. The following commands are available.

- "count" on page 21-69
- "display" on page 21-70
- "fails" on page 21-71
- "frequency" on page 21-72
- "target" on page 21-73
- "timeout" on page 21-73

Syntax	*Example*

`[NO] TUNNEL-KEEPALIVE`	`IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500)#` **`tunnel-keepalive`**
	`VI(500): Object successfully set`
	
	`IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500) TUNNEL-KEEPALIVE#`

Default Value

Disabled

Command Level

vi

Related Commands

None

count

Set the number of requests sent in each keep-alive set.

Syntax	*Example*
COUNT <number>	IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500) TUNNEL-KEEPALIVE# **count 5** IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500) TUNNEL-KEEPALIVE# **display count** COUNT : 5

Default Value

4 requests

Command Level

tunnel-keepalive

Related Commands

None

display

View the current keep-alive configuration.

Syntax	*Example*
DISPLAY	IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500) TUNNEL-KEEPALIVE# **display** TARGET : 0.0.0.0 COUNT : 4 TIMEOUT : 3 FAILS : 3 FREQUENCY : 300 ADMIN-STATUS : DISABLE Tunnel Keepalive Statistics: Requests Sent 0 Responses Received 0 Minimum RTT (msec) 0 Maximum RTT (msec) 0 Average RTT (msec) 0 Errors (while sending) 0 Time Outs 0 Lates 0 Unreachables 0 Current State TESTING

Default Value

None

Command Level

tunnel-keepalive

Related Commands

None

fails

Number of requests that can be unanswered before tunnel is declared down.

Syntax	Example
FAILS <number>	IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500) TUNNEL-KEEPALIVE# **fails 5** IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500) TUNNEL-KEEPALIVE# **display fails** FAILS : 5

Default Value

3 failed requests

Command Level

tunnel-keepalive

Related Commands

None

frequency

Number of seconds between each set of keep-alive requests.

Syntax	Example
FREQUENCY <seconds>	IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500) TUNNEL-KEEPALIVE# **frequency 500** IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500) TUNNEL-KEEPALIVE# **display frequency** FREQUENCY : 500

Default Value

300 seconds

Command Level

tunnel-keepalive

Related Commands

None

target

Target of the tunnel keepalive request.

Syntax	*Example*
TARGET <address>	IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500) TUNNEL-KEEPALIVE# **target 192.168.24.3** IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500) TUNNEL-KEEPALIVE# **display target** TARGET : 192.168.24.3

Default Value

None

Command Level

tunnel-keepalive

Related Commands

None

timeout

Amount of time in seconds waiting for each tunnel keepalive request.

Syntax	*Example*
TIMEOUT <seconds>	IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500) TUNNEL-KEEPALIVE# **timeout 5** IPNOS: CONFIG VPN(210) VR(192.168.120.1) VI(500) TUNNEL-KEEPALIVE# **display timeout** TIMEOUT : 5

Default Value

3 seconds

Command Level

tunnel-keepalive

Related Commands

None

APPENDIX

A SNMP Traps

The following table decribes the trap messages used for IPNOS Simple Network Management Protocol (SNMP), version 2. The trap messages allow a remote network management server such as InVision SMS to get or set IPNOS functions and configurations.

Table A-1 SNMP Traps

Type	*Event Name*	*OID*	*Description*	*Release*
Generic Trap	Cold Start	1.3.6.1.6.3.1.1.5.1	The SNMP agent has just come up after an IPSX reboot.	1.1
	Authentic ation Failure	1.3.6.1.6.3.1.1.5.5	An incoming SNMP packet does not have the proper authentication, i.e., bad community name.	1.1
Enterprise -Specific Traps	Blade Removed	1.3.6.1.4.1.3085.3.1.1.1.1	The SNMP agent has detected that the one of the existing Blade was removed from its slot.	1.1
	Blade Inserted	1.3.6.1.4.1.3085.3.1.1.1.2	The SNMP agent has detected that the one of the Blade was inserted in the empty slot.	1.1
	Blade State Changed	1.3.6.1.4.1.3085.3.1.1.1.3	The SNMP agent has detected that a Blade has changed state. A description of the Blade as well as the entered state is identified in this trap.	1.1
	Fan Status Change	1.3.6.1.4.1.3085.3.1.1.1.4	The SNMP agent has detected a change in the fan status.	2.0

Table A-1 SNMP Traps (Continued)

Type	Event Name	OID	Description	Release
	Configuration Changed	1.3.6.1.4.1.3085.3.1.1.1.5	The SNMP agent has detected that Configuration was changed. The Trap PDU for this csOrionConfigChange, also carries each of the modified object's OID followed by the changed value. In effect, there are one or more VarBinds follows the csOrionNumConfigChanges object of the Trap PDU. Since these OIDs are the member of one of the table defined in this MIB, type information of the value shall be got from the OID. If the csOrionConfigChangeType is *delete*, the VarBind that follows the csOrionNumConfigChanges contains the Index(es) of the entry which was deleted. If the csOrionConfigChangeType is *addModify*, then the VarBind contains all the changed OID-Value pairs followed by the index(es).	1.1
	Temperature Status Change	1.3.6.1.4.1.3085.3.1.1.1.6	This trap indicates that the temperature status of the Orion system has changed states. If the value of csOrionSystemTempStatus is tempRisingPastUpperCriticalThreshold(1), the given sensor has detected a transition from the normal temperature range to the hot range. If the value of csOrionSystemTempStatus is tempFallingBelow-UpperCriticalThreshold (2), the given sensor has detected a transition from the 'hot' range back down to the normal range.	2.0
	Power Supply Status Change	1.3.6.1.4.1.3085.3.1.1.1.7	This trap indicates that one of the power supplies has changed state.	2.0

Table A-1 SNMP Traps (Continued)

Type	Event Name	OID	Description	Release
	Logical Link Down	1.3.6.1.4.1.3085.3.1.1.1.8	This trap indicates that a Logical Link on the Orion is about to enter the down state.	1.1
	Logical Link Up	1.3.6.1.4.1.3085.3.1.1.1.9	This trap indicates that a Logical Link on the Orion is about to enter the up state.	1.1
	Virtual Interface Down	1.3.6.1.4.1.3085.3.1.1.1.10	The SNMP agent has detected that the csOrionVIfOperStatus object for one of its communication links is about to enter the down state from some other state (but not from the notPresent state). This other state is indicated by the included value of csOrionVIfOperStatus.	1.1
	Virtual Interface Up	1.3.6.1.4.1.3085.3.1.1.1.11	The SNMP agent has detected that the csOrionVIfOperStatus object for one of its communication links left the down state and transitioned into some other state (but not into the notPresent state). This other state is indicated by the included value of ifOperStatus.	1.1
	Customer VR Up	1.3.6.1.4.1.3085.3.1.1.1.28	This signifies that a Virtual Router which services a customer network has come up.	Future
	Customer VR Down	1.3.6.1.4.1.3085.3.1.1.1.29	This signifies that a Virtual Router which services a customer network has gone down.	Future
	Service Provider VR Up	1.3.6.1.4.1.3085.3.1.1.1.30	This signifies that a Virtual Router which is designated as a service provider VR has come up.	Future
	Service Provider VR Down	1.3.6.1.4.1.3085.3.1.1.1.31	This signifies that a Virtual Router which is designated as a service provider VR has gone down.	Future

Table A-1 SNMP Traps (Continued)

Type	Event Name	OID	Description	Release
	Control VR UP	1.3.6.1.4.1.3085.3.1.1.1.32	This signifies that a Virtual Router which is designated as a control VR has come up. If a csOrionBladeRemoved Trap precedes this trap and if that blade is the master control blade then this trap might be from the control VR of the new master blade.	Future
	Control VR Down	1.3.6.1.4.1.3085.3.1.1.1.33	This signifies that a Virtual Router which is designated as a control VR has gone down.	Future
	Service Provider VI Up	1.3.6.1.4.1.3085.3.1.1.1.34	This signifies that a Virtual Interface on a Service Provider VR has come up.	1.1
	Service Provider VI Down	1.3.6.1.4.1.3085.3.1.1.1.35	This signifies that a Virtual Interface on a Service Provider VR has gone down.	1.1
	Loss of Signal	1.3.6.1.4.1.3085.3.1.1.1.36	This signifies that a physical port on a blade is experiencing a loss of signal (LOS) on the line. Applicable to both DS3 and OC3.	1.1
	Loss of Frame	1.3.6.1.4.1.3085.3.1.1.1.37	This signifies that a physical port on a blade is experiencing a loss of Frame (LOF) on the line. Applicable to both DS3 and OC3.	1.1
	DS3 Alarm Indication	1.3.6.1.4.1.3085.3.1.1.1.38	This signifies that a physical port on a blade generates an alarm (AIS) that indicates a failure in upstream far end.	1.1
	Loss of Clock	1.3.6.1.4.1.3085.3.1.1.1.39	This signifies that a physical port on a blade is experiencing a loss of Clock (LOC) on the line. Applicable to both DS3 and OC3.	1.1
	Blade Restore	1.3.6.1.4.1.3085.3.1.1.1.40	This indicates whether a blade has been restored successfully. The VarBinds that follow contain the list of VPN/VRs affected.	1.1

Table A-1 SNMP Traps (Continued)

Type	Event Name	OID	Description	Release
	DS3 Port Idle	1.3.6.1.4.1.3085.3.1.1.1.41	This signifies that a physical port on a blade received an IDLE signal from the far end indicating it is unequipped to handle any data transfer.	1.1
	DS3 Alarm Clear	1.3.6.1.4.1.3085.3.1.1.1.42	This signifies that a physical port on a blade, which had sent an alarm earlier, indicates that the previous alarm is cleared. Applicable to DS3.	1.1
	OC3 Alarm	1.3.6.1.4.1.3085.3.1.1.1.43	This signifies that an OC3 physical port on a blade had sent an alarm.	1.1
	OC3 Alarm Clear	1.3.6.1.4.1.3085.3.1.1.1.44	This signifies that an OC3 physical port on a blade, which had sent an alarm earlier, indicates that the previous alarm has been cleared.	1.1
	Packet Ring Status Change	1.3.6.1.4.1.3085.3.1.1.1.45	This signifies that the packet ring status of the chassis has changed. If both endpoints change from zero to r[1,26], that means the status has changed to protection ring. If both endpoints change from r[1,26] to zero, that means the status has changed to no protection ring. If one/both endpoints change from r[1,26] to different r[1,26], that means the topology has changed but not ring status.	2.1

APPENDIX

B Glossary

A–C

access blade IPSX card that provides line termination, hardware-assisted IP forwarding, and encryption services. Access blades include the DS3 blade (channelized and unchannelized) and the OC3 asynchronous transfer mode (ATM) and packet over SONET (POS) blade.

access control A defined set of actions applied against network traffic (for example, “permit” or “deny”) to control access to network resources. Access control is a core functionality of enterprise security deployment, and is commonly implemented at network gateways.

access control list (ACL) A sequential list of permit and deny conditions. The list defines the connections permitted to pass through a device, usually a router. Access control lists act as a crude means of limiting access to a network.

access list Identifies groups of packets. Once packets are defined on an access list, specific actions and services, such as packet filtering or IPSec, can be configured and implemented. An access list defines addresses that are and are not encrypted.

Advanced Encryption Standard (AES) An initiative by the federal government to select a standard encryption algorithm capable of protecting sensitive government material. The algorithm will replace DES as the standard for symmetric, block cipher encryption and will include a 128-bit key length. AES is designed to encrypt data more quickly and efficiently than DES or Triple-DES.

asymmetric algorithm An encryption algorithm that uses one key to encrypt a message and another to decrypt the message.

audit trail In computer security systems, a chronological record of system resource use, including user login, file access, and whether any actual or attempted security violations occurred.

authentication The process of checking and confirming an entity's identity. Also, the act of ensuring that a communication between two parties has not been tampered with. IPSec virtual private networks use the Authentication Header to perform this function.

Authentication Header (AH) A component of the IPSec protocol used to verify that the contents of a packet have not been changed, and to validate the identity of the sender. See also *encapsulated security payload*.

bastion host A specific host used to intercept packets entering or leaving a network. The host includes proxy versions of services to remove the security burden from hosts on the network. In general, a bastion host is a particularly strong point of defense.

blade The hot-swappable card that provides services in the IPSX switch. CoSine IPSX blades are the access blades, processor blades, and control blades.

block cipher An encryption method that places data in fixed-size blocks before encryption. For example, a 256-bit message that was subjected to a 64-bit block cipher, such as DES, would be encrypted in four discrete blocks.

brute force attack An attempt to illicitly recover a cryptographic key by trying to guess all possible key values.

certificate authority (CA) A trusted third-party organization or company that issues digital certificates used to create digital signatures and public-private key pairs. The role of the CA in this process is to guarantee that individuals granted the unique certificates are who they claim to be. CAs also act as repositories for digital certificates. See also *digital certificate.*

certificate revocation list (CRL) A list of certificates that are no longer valid. CRLs are maintained by a certificate authority, and are checked to ensure the validity of existing certificates.

Challenge Handshake Authentication Protocol (CHAP) A protocol that authenticates remote users. CHAP is a server-driven, three-step authentication mechanism that depends on a shared secret password that resides on both the server and the client.

channels The primary conduits for customer traffic. Channels range in capacity from a single DS0 (56/64 Kbps) to 24 aggregated DS0s—a full T1 bandwidth.

ciphertext A message that has been encrypted to maintain its privacy when travelling over untrusted networks.

command line interface (CLI) An environment in which you interact with the operating system by entering commands and arguments at the command line. The IPSX switches run CoSine's CLI-based operating system called IPNOS.

content security The ability to specify the content of a communication as an element of a security policy, as opposed to defining a security policy on the basis of header information only. Effective content security requires that a firewall understand the internal details of the protocols and services it monitors. Examples of content security include checking for viruses in download files, not allowing emails from or to specific addresses, and allowing access to web pages that contain certain words only during specified time periods.

control blade IPSX service blade that is the central point for management and configuration of all switch elements.

crypto map A software configuration that performs two primary functions: it selects data flows that need security processing, and it defines the policy for these flows and the crypto peer to which traffic needs to go.

D–F

Data Encryption Standard (DES) A widely used encryption algorithm that incorporates private (secret) keys. DES is a 64-bit block cipher used because of its speed. See also *Triple-DES*.

demilitarized zone (DMZ) A network located outside a trusted or secure network that is protected from an untrusted network. The DMZ sits between the Internet and an internal network's line of defense—usually a combination of firewalls and bastion hosts. Typically, it contains devices accessible to Internet traffic, such as web (HTTP), FTP, SMTP (email), and DNS servers.

denial-of-service (DoS) attack An attack on a network that overwhelms the target with spurious data to prevent legitimate connection attempts. Different from attacks whose purpose is to penetrate the target system, DoS attacks do not reveal sensitive data to the attacker. Many DoS attacks exploit limitations in the TCP/IP protocols.

Device Manager An application manager in the InVision™ Service Management System (SMS) used for switch and blade configuration and management.

Dialup Manager An application manager in InVision SMS used to configure Point-to-Point Tunneling Protocol, (PPTP), Layer 2 Tunneling Protocol, (L2TP), and IPSec dialup services, as well as RADIUS-based authorization and authentication.

Diffie-Hellman A public key scheme, invented by Whitfield Diffie and Martin Hellman, used for sharing a secret key without communicating secret information, thus avoiding the need for a secure channel. Once correspondents have computed the secret shared key, they can use it to encrypt communications.

digital certificate The digital equivalent of an ID card; an electronic document attached to a public key by a certificate authority. Certificates are used during the key exchange process to ensure that keys are exchanged by two known entities. They are a safe means of distributing public keys because they can be validated and signed by a certificate authority.

digital signature A digital code that can be attached to an electronically transmitted message, uniquely identifying the sender. Because they cannot be forged, digital signatures guarantee that people sending messages really are who they claim to be, and that messages have not been tampered with in transit. They are a key component of most authentication schemes.

DS3 blade, channelized Access blade on which the IPSX administrator can use the CLI or InVision SMS software to delegate and configure up to 128 logical channel groups per port.

DS3 blade, unchannelized Access blade on which the IPSX administrator can configure the entire high-speed line as a single connection.

DSL Manager An application manager in InVision SMS used to configure digital subscriber line (DSL) connection profiles and service selection templates.

E1 Concentrator An E1 Concentrator multiplexes 16 E1 channels to a single E3 channel.

E3 blade, channelized Access blade that has three interfaces, or ports, each of which carries a 34.368 Mbps signal. Among the three E3 ports, you can configure a maximum of 256 channels to carry customer traffic.

E3 blade, unchannelized Access blade that provides two physical interfaces, each carrying a 34.368 Mbps signal. The entire bandwidth of each unchannelized interface is treated as one large channel; thus an unchannelized E3 blade actually carries two channels.

encapsulating security payload (ESP) A fundamental component of IPSec-compliant VPNs, ESP specifies an IP packet's encryption, data integrity checks, and sender authentication, which are added as a header to the IP packet. See also *Authentication Header*.

export policy A routing policy that incorporates configured protocols, actions, and general access IDs for outbound packet transmissions. See also *import policy*.

firewall A security gateway positioned between two different networks, usually between a trusted network and the Internet. A firewall ensures that all communications that cross it conform to the organization's security policy. Firewalls track and control communications, deciding whether to pass, reject, encrypt, or log them. They also can be used to secure sensitive portions of a local network.

Firewall Manager An InVision SMS tool used to configure firewalls and other security services.

Frame Over IP Manager An InVision SMS tool used to link connectionless IP networks and the dedicated, connection-oriented Frame Relay infrastructure used by many telecommunications companies.

fully meshed network A network in which devices are organized in a mesh topology; each node is connected to every other node by means of a physical link or a virtual link.

G–I

gateway A device positioned between two networks through which all communications must pass. Gateways are commonly used to enforce security policies and to provide encryption, decryption, and authentication services.

hash A one-way function that takes a message of any length and produces a fixed-length digest. In security, a message digest is used to validate that the contents of a message have not been altered in transit. The Secure Hash Algorithm (SHA-1) and Message Digest 5 (MD5) are commonly used hashes.

hashed message authentication code (HMAC) A mechanism for message authentication that uses cryptographic hash functions. HMAC can be used with any iterative cryptographic hash function—for example, MD5 or SHA-1—in combination with a secret shared key. The cryptographic strength of HMAC depends on the properties of the underlying hash function.

import policy A routing policy that incorporates configured actions, general access IDs, AS origin information, and preference assignments for inbound packet transmissions. See also *export policy*.

InGage Customer Network Management A browser-based customer care system for the subscriber site that enables enterprise network managers to monitor and maintain their portion of the SP's network. See also *InVision Service Management System*.

Internet Key Exchange (IKE) The key management protocol used in IPSec, IKE combines ISAKMP and Oakley to create encryption keys and Security Associations. See also *ISAKMP*.

Internet Protocol Security (IPSec) The industry standard for establishing VPNs, IPSec comprises a group of protocols and algorithms to provide authentication and encryption of data across IP-based networks.

Internet Security Association and Key Management Protocol (ISAKMP) A protocol that allows the receiver of a communication to obtain a public key and use digital certificates to authenticate the sender's identity. ISAKMP is designed to be key-exchange-independent; that is, to support many different key exchanges. See also *Internet Key Exchange* and *Oakley*.

InVision Service Management System (SMS) A UNIX-based, multi-tiered system that allows SPs to easily deliver, provision, account for, and maintain IP services for thousands of enterprise networks.

IP Service Delivery Platform A platform that enables SPs to provide IP services to their wholesale and retail subscribers. The platform consists of three elements: the IPSX switch, InVision SMS software, and InGage Customer Network Management software.

IP Services Manager An InVision SMS tool used to create, configure, and delete VPNs and their elements (virtual routers, virtual interfaces, logical interfaces, and tunnel interfaces). The IP Services Manager is also used to modify and configure protocols and to map IP services to network elements.

IPSX 3500™ A CoSine IP service processing switch that provides managed services such as virtual private networks and other processor-intensive IP services. The chassis is a backplane design with 8 slots. You can install three chassis in a single rack. A component of the IP Service Delivery Platform.

IPSX 9000™ A CoSine IP service processing switch that provides managed services such as virtual private networks and other processor-intensive IP services. The chassis is a midplane design with 26 slots, 13 on each side of the chassis. A component of the IP Service Delivery Platform.

key management A mechanism for distributing encryption keys in a public key scheme. Key management includes key generation, certification (which can also be performed by an external certificate authority), and distribution. Key management can be manual or automated. As a VPN grows, a scalable key management system, such as IKE, becomes more important. See also *IKE*.

Lightweight Directory Access Protocol (LDAP) A mechanism used by Internet clients to access and manage a database of directory services over a TCP/IP connection. LDAP is a simplification of the X.500 directory access protocol.

M–P

Message Digest 5 (MD5) A one-way hashing algorithm that produces a 128-bit hash. It is used in AH and ESP. See also *SHA-1*.

network address translation (NAT) An Internet standard that enables a local-area network (LAN) to use one set of IP addresses for internal traffic and a second set for external traffic. This reduces the need for globally unique IP addresses, and provides security by hiding internal IP addresses.

network element An aspect of a network, either physical or virtual, that can be managed through the InVision SMS System Manager.

Oakley A key determination protocol based on the Diffie-Hellman algorithm that provides added security, including authentication. Oakley was the key-exchange algorithm mandated for use with the initial version of ISAKMP, although various algorithms can be used. Oakley describes a series of key exchanges called "modes" and details the services provided by each; for example, perfect forward secrecy for keys, identity protection, and authentication. See also *ISAKMP*.

OC-3 Asynchronous Transfer Mode (ATM) blade Access blade that offers optical networking support. OC-3 ATM blades provide ATM connectivity through a single 155 Mbps interface, which is divided into ATM subinterfaces and virtual channels that carry smaller allocations of bandwidth.

OC-3 Packet over SONET (POS) blade Like OC-3 ATM blades, OC-3 POS blades have one active interface that runs at 155 Mbps. The bandwidth occupied by user payloads is 130.152 Mbps; the remainder is used for overhead. OC-3 POS blades support both SONET and Synchronous Digital Hierarchy (SDH).

packet A bundle of data that includes a header containing control information and (usually) user data. In IP networks, packets are often called datagrams.

packet filter A simple type of firewall technology that bases security decisions on information contained in an IP packet header, such as service type, source, or destination. Packet filtering is commonly used in routers. A router decides whether to permit or deny a packet's transit based on whether the packet matches a predetermined set of parameters.

payload The portion of an IP packet that holds actual message data. In VPNs, the payload is the part of a packet that is authenticated or encrypted.

Perfect Forward Secrecy (PFS) A condition derived from an encryption system that changes encryption keys often and ensures that no two sets of keys have any relation to each other. The advantage of PFS is that if one set of keys is compromised, only communications using those keys are at risk. An example of a system that uses PFS is Diffie-Hellman.

Point-to-Point Protocol (PPP) A method for transmitting packets over serial point-to-point links, such as dial-up lines. PPP provides no data authentication or encryption capabilities.

private key The private part of an asymmetric encryption schema, known only to the owner and never transmitted over the network.

processor blade Processor blades provide additional processing power on the IPSX switch. They are designed to support application firewalls, proxies, NAT, and encryption.

public key The publicly available component of an integrated asymmetric key pair. Public keys are included in digital certificates, and are often referred to as encryption keys.

public key cryptography A cryptographic system that uses both a private and a public key. Each communicator has a private key that is kept secret and a public key that is published for everyone. The sender looks up the recipient's public key and uses it to encrypt the message. The recipient uses the private key to decrypt the message. Because key owners do not need to transmit their private keys to anyone for their messages to be decrypted, the private keys are not in transit and are not vulnerable.

public key infrastructure (PKI) A system of digital certificates, certificate authorities, and other registration authorities that verify and authenticate the identity of each party involved in a secure transaction. A PKI enables organizations to set up and define the scope and participants of a VPN deployment by establishing a trust relationship between their respective certificate authorities, each of which is then responsible for validating the users of its organization for participation within the VPN. See also *certificate authority*, *digital certificate*, and *digital signature*.

Routing Manager The InVision SMS sub-application used to create routing policies in the IPSX switch.

RSA A public key cryptographic system algorithm (named after its inventors, Rivest, Shamir and Adleman) with a variable key length.

S–V

Secure Hash Algorithm (SHA-1) A widely used hash function for use with Digital Signal Standard (DSS). SHA-1 is more secure than MD5.

Security Association (SA) An IPSec term that describes an agreement between two parties about what rules to use for authentication and encryption algorithms, key exchange mechanisms, and secure communications.

Security Parameter Index (SPI) A portion of the IPSec Authentication Header that communicates what security protocols, such as authentication and encryption, are being used for each packet in a VPN connection.

security policy A set of rules that defines an organization's network security parameters, including access control, authentication, encryption, content security, network address translation, logging, and other security components.

service level agreement (SLA) A contract between a service provider and customer that specifies a level of network service, such as bandwidth availability, network uptime, and other measures of performance. It also specifies remunerations if these service levels are not met.

session key A cryptographic key for encrypting data for a fixed period, usually a single communication session. The key is discarded when the session is over, limiting the likelihood that the key will be broken and the data compromised.

shared secret A string of text or numbers communicated between two parties in an out-of-band connection, such as over the phone, through the mail, or on a disk. Both parties use the same key for authentication, encryption, and decryption.

Site-to-Site Tunnel Manager An application manager in InVision SMS used to establish tunnel connections and to set up IPSec and IKE security.

symmetric algorithm An encryption algorithm in which the same secret key is used for both encryption and decryption. Also known as conventional, secret-key, and single-key algorithms. Block and stream ciphers are classes of symmetric algorithms.

System Manager An application manager in InVision SMS used to view, configure, add, or delete a particular IPSX switch or VPN. The System Manager also functions as a central navigation point to the Device Manager and the IP Services Manager.

transport mode An IPSec mode of operation in which the data payload is encrypted, but the original IP header is left untouched. The IP addresses of the source or destination can be modified if the packet is intercepted. Because of its construction, transport mode can be used only when the communication endpoint and cryptographic endpoint are the same location. VPN gateways providing encryption and decryption services for protected hosts cannot use transport mode for protected VPN communications. See also *tunnel mode*.

Triple-DES A 168-bit encryption algorithm that encrypts data blocks with three different keys in succession, achieving a higher level of encryption. Triple-DES is one of the strongest encryption algorithms available for use in VPNs.

tunnel A designated path through which encapsulated data travels across the Internet.

tunnel interface (TI) An interface on the IPSX switch through which data is encrypted and decrypted in a manner that allows the encrypted tunnel to appear as a routable link.

tunnel mode An IPSec mode of operation in which the entire IP packet, including the header, is encrypted and authenticated and a new VPN header is added, protecting the entire original packet. This mode can be used by both VPN clients and VPN gateways, and protects communications that come from or go to non-IPSec systems. See also *transport mode*.

tunneling The act of transmitting data formed with one protocol within the structures of another protocol, allowing the data to be sent in the prevailing format of a network—usually an IP network.

virtual interface (VI) A VI is to a virtual router what an IP interface is to a physical router.

virtual private network (VPN) A set of networking services that enable private, secure communications over public IP networks. It enables entities to use the Internet securely as part of their networks' infrastructures by encrypting the traffic that goes over the public network.

virtual router (VR) Associated with VPNs, VRs must reside on a particular blade. A VR examines an incoming packet, determines its destination, performs any necessary encryption, decryption, or filtering, and sends it to its next destination. VRs also provide other IP services, such as firewalls and network address translation. Like physical routers, VRs have interfaces. See also *virtual private network* and *virtual interface*.

Index

Symbols

? command 2-2

Numerics

01xCC parameter
    oc3 J0 command 9-2
0x01 parameter
    oc3 J0 command 9-2
10 Mbps 4-7
100 Mbps 4-7
3des parameter
    encryption comnmand 17-24

A

AAL 10-56
AAL1 10-56
    using ABR 10-56
    using CBR 10-56
AAL3/4 10-56
    using NRT-VBR 10-56
    using RT-VBR 10-57
AAL5 10-56
    using UBR 10-57
a-b parameter
    bgp-community command 16-49, 16-63
ABOR parameter
    FTP function 18-22
    FTP transaction 18-26
abort command 2-3
ABR 10-56
    using ICR 10-36
    using MCR 10-40
access command 18-5
    compared to acl command 18-5
access control list
    see acl command
access parameter
    fw command 18-5
access-list command 16-5
    compared to acl command 16-5
ACCT parameter
    FTP function 18-22
    FTP transaction 18-26
ACHR parameter
    HTTP packet header 18-41
acl command 16-7
    assigning to export policy 16-46
    assigning to import policy 16-62
    assigning to IPSec-dialup crypto map 17-32
    assigning to IPSec-ISAKMP crypto map 17-40
    assigning to IPSec-manual crypto map 17-47
    assigning to VI packet filter 21-53
    assigning to VR packet filter 16-108
    compared to access command 18-5
    compared to access-list command 16-5
action command
    export policy 16-47
    firewall proxy 18-5
    import policy 16-62
ACTX parameter
    HTTP feature 18-37
address parameter
    ip command 21-26
    match command 17-32, 17-40, 17-48
    traceroute command 16-162
Address Resolution Protocol
    see ARP 16-16
administrative status 4-8
admin-status command 19-9
AENC parameter
    HTTP packet header 18-41
aggregate parameter
    protocol command 16-54
agressive-mode parameter
    set flag command 17-33, 17-41
AH 17-3
ah-md5-hmac parameter
    transform-set command 17-3
ah-rfc1828 parameter
    transform-set command 17-3
ah-sha-hmac parameter
    transform-set command 17-3
AIS
    sending loopback on channelized DS3 blade 5-10
    sending loopback on channelized E3 blade 7-11
    sending loopback on unchannelized DS3 blade 6-7
    sending loopback on unchannelized E3 blade 8-7
alarm indicator signal
    See AIS
alarm parameter
    oc3 command 9-2

all parameter
area command 14-3, 21-16
protocol command 16-54
ALLO parameter
FTP function 18-22
FTP transaction 18-26
ALNG parameter
HTTP packet header 18-41
ALOW parameter
HTTP packet header 18-41
America OnLine
See AOL
ANSI
setting Frame Relay LMI type 12-3
ansi parameter
frame-relay lmi-type command 12-3
anti-clogging tokens 17-14, 17-20
AOL
setting up a firewall proxy 18-10
aol-proxy command 18-10
aol-proxy parameter
fw command 18-10
APPE parameter
FTP function 18-22
FTP transaction 18-26
apps parameter
area command 14-3, 21-16
area 20-46
area command 20-46
area identification number 20-46
area parameter
ip ospf command 21-26
ARP 16-16
clearing cache 16-22
arp command 16-16
AS 16-48, 16-53, 16-65
AS parameter
peer-as command 16-66
Assigning a crypto map to a VI 21-10
Asynchronous Transfer Mode
See ATM
asynchronous transfer mode
See ATM
ATM 10-1, 10-40, 10-56
ATM adaptation layer
See AAL
ATM cell 10-56, 10-69
atm command 10-3, 10-4, 10-23
ATM interface
activate ILMI 10-4
assigning interface name 10-4
assigning neighbor ATM interface 10-5
assigning neighbor ATM interface name 10-5
enable administrative status 10-4
setting ILMI VCI 10-4
setting ILMI VPI 10-4
setting maximum VCCs 10-4
setting maximum VCI bits 10-4
setting maximum VPCs 10-4
setting maximum VPI bits 10-5
ATM PVC 10-56
ATM sub-interface 10-32
auth parameter
area command 14-3, 21-16
authentication 16-25, 16-115, 17-3
setting OSPF password on a VI 21-26
authentication command 17-21
authentication header
See AH
authentication parameter
ppp command 14-7, 16-93
pptp command 16-115
autoneg 4-3
autoneg command 4-3
autonomous system
See AS
Available Bit Rate
See ABR
available bit rate
See ABR

B

bandwidth 21-7
BC
setting for FRoIP 19-12
bc command 19-12
bc parameter
frame-relay command 12-10
BE
setting for FRoIP 19-13
be command 19-13
be parameter
frame-relay command 12-10
BGP 16-48, 20-3
deleting IP address 16-22
origin type for import policy 16-65
bgp command 20-3
bgp parameter
protocol command 16-54
bgp-as command 16-48
bgp-community 16-63
bgp-community command 16-49
BIN parameter
HTTP function 18-38
HTTP transaction 18-43
Binary 18-38, 18-43
bits per second
See bps
blade 3-1

blade command 3-3
block 16-47
boot configuration 3-14
bootcfg command 3-12
Border Gateway Protocol
    See BGP
broadcast parameter
    ip command 21-26
bundle parameter 21-7, 21-63
burstsize parameter
    rate-limit command 21-61

C

cablelength 6-2
cablelength command 5-2, 6-2
cache 16-16
CBR 10-56, 10-57
CDUP parameter
    FTP function 18-22
    FTP transaction 18-26
cells per second 10-36, 10-38, 10-40, 10-41
CENC parameter
    HTTP packet header 18-41
certificate 16-17, 16-18
    request 17-11
certificate command 16-17, 16-18
CGI 18-37
Challenge Handshake Authentication Protocol, Message Digest 5
    See CHAPMD5
Challenge-Handshake Authentication Protocol
    See CHAPMD5
channel 5-3, 7-2
channel command 5-3, 7-2
channel parameter
    route command 19-16
CHAPMD5 16-115
    used in PPP 14-7
chapmd5 parameter
    ppp authentication command 14-7
child-limit 18-16
child-limit command 18-16, 18-31, 18-52, 18-59, 18-67, 18-73, 18-87, 18-96
CIR
    setting for FRoIP 19-14
cir command 19-14
cir parameter
    frame-relay command 12-10
CISCO
    setting Frame Relay LMI type 12-3
Cisco
    router compatibility with ISAKMP Phase I rekeys 17-27
cisco parameter
    frame-relay lmi-type command 12-3
clear 16-22
clear command 16-22
    clearing ARP cache 16-22
    clearing DHCP relay statistics 16-22
    clearing IS-IS routes 16-22
    clearing VI statistics counters 16-22
    clearing VR statistics counters 16-22
    deleting BGP IP address 16-22
clear counters command
    clearing DHCP relay statistics 16-29
CLI 1-1
CLNG parameter
    HTTP packet header 18-41
clock 6-3, 8-2
    used with OC-3 blades 9-2
clock command 5-5, 6-3, 8-2
clock parameter
    oc3 command 9-2
cls command 2-4
command
    Structure 1-2
Command Line Interface 1-1
command line interface
    See CLI
command shortcuts 1-4
commit command 2-5
Committed Burst Size
    See BC
Committed burst Size
    See BC
Committed information rate
    See CIR
committed information rate
    See CIR
community parameter
    snmp-server command 16-145
compatability flags for ISAKMP 17-10
CompuServe
    setting up a firewall proxy 18-11
config command 2-6
config mode 1-3
config-req-retransmit parameter
    ppp command 14-7, 16-93
configuration
    mode 2-6
configuration parameter
    area command 14-3, 21-16
connected-to command 21-7
connection parameter
    pptp command 16-115
console terminal 1-2
constant bit rate
    See CBR
context 1-3

Control blade restrictions with VR creation 15-2
control parameter
type command 16-164
controller 3-35
conventions, typographical xxv
CoSine 1-2
Cosine IPSX 9000 Command Line Reference xxiii
CoSine IPSX 9000™ 1-1
cost 16-52
cost parameter
ip ospf command 21-26
count parameter
ping command 16-112
CRC
used with OC-3 blades 9-2
crc parameter
oc3 command 9-2
create 3-36, 15-2, 16-24
create command 3-36, 15-2, 16-24
Creating an ISAKMP Phase I policy 17-20
critical parameter
level command 14-3, 18-13, 21-16
crypto 16-25, 21-10
crypto command 16-25, 21-10
Crypto Commands 17-1
crypto map
allowing one ISAKMP Phase II key exchange per host 17-41
allowing one ISAKMP Phase II key exchange per network 17-33, 17-41
assigning a ISAKMP policy 17-33, 17-41
assigning a local identity 17-34, 17-42
assigning a transform-set 17-34, 17-42
assigning a transform-set to a manual crypto map 17-49
assigning to a VI 21-10
defining a cipher key for inbound ESP 17-49
defining a cipher key for outbound ESP 17-49
defining a key for inbound AH 17-48
defining a key for outbound AH 17-48
defining an authentication key for inbound ESP 17-49
defining an authentication key for outbound ESP 17-49
defining SPI for inbound AH 17-48
defining SPI for inbound ESP 17-48
defining SPI for outbound AH 17-48
defining SPI for outbound ESP 17-49
displaying current IPSec-dialup configuration 17-30
forcing ISAKMP Phase I negotiations in fewer packets 17-33, 17-41
idle timeout of a dialup connection 17-33
IPSEC-dialup 17-29
IPSEC-ISAKMP 17-29
IPSEC-manual 17-29
manually typing in a pre-shared key 17-33, 17-41
resetting entire SA 17-33, 17-41
setting a map name 17-29
setting a priority number for multiple maps 17-29
setting aggressive mode 17-33, 17-41
setting Diffie Hellman key exchange group for ISAKMP Phase II 17-34, 17-42
setting flags 17-33, 17-41
setting ID for ISAKMP Phase II key exchanges 17-32, 17-40
setting ISAKMP Phase I negotiations to take place in normal six-packet exchange 17-33, 17-41
setting lifetime of ISAKMP phase II keys 17-34, 17-42
setting main mode 17-33, 17-41
setting PFS for ISAKMP Phase II 17-33, 17-41
cryptographic synchronization 17-5
cs-proxy command 18-11
cs-proxy parameter
fw command 18-11
customer parameter
type command 16-164
CWD parameter
FTP function 18-22
FTP transaction 18-26
Cyclic Redundancy Check
See CRC

D

data encryption standard
See DES
Data EncryptionStandard
See DES
data link connection identifier
See DLCI
DATE parameter
HTTP packet header 18-41
dce parameter
frame-relay intf-type command 12-3
dead-interval parameter
ip ospf command 21-26
Dead-intervals on a VI 21-26
debug parameter
level command 14-4, 18-13, 21-16
decr-TTL 16-27
decr-ttl command 16-27
default-cost parameter
area command 20-47
default-service-slot command 19-2
default-slot command 19-9
default-window-size command 19-3
default-window-timeout command 19-4

DELE parameter
FTP function 18-22
FTP transaction 18-26
delete 15-5, 16-28
delete command 15-5, 16-28
delete file command 3-52
delete VPN, VR, VI command 3-37
denial-of-service attack 17-14
denial-of-service attacks 17-19
Deny a packet into a VI 21-53
Deny parameter
VR packet-filter command 16-108
deny parameter
action command 18-6
VI packet-filter command 21-53
DES 17-3
DES encryption 17-24
des parameter
encryption command 17-24
destination
IP address 16-9
destination command 18-6
destination port
See dstport
dhcp-relay command 16-29
differentiated services
see DS
Diffie-Hellman 17-25
setting for ISAKMP Phase II 17-34, 17-42
digital signature 17-21
Digital Signature Standard
See DSS
dir command 3-54
DIR parameter
HTTP function 18-38
HTTP transaction 18-43
disable parameter
admin-status command 19-9
lns command 16-90, 16-92
display
ACL configuration 16-8
ATM interface configuration 10-17, 11-2
ATM subinterface 10-32
BGP configuration 20-9
blade configuration 3-5
boot configuration 3-15
certificate configuration 16-19
Channelized DS3 Blade configuration 5-6
Channelized DS3 Blade encapsulation 5-18
Channelized E3 Blade configuration 7-4
Channelized E3 Blade encapsulation 7-16
DLCI configuration 19-15
Ethernet Blade configuration 4-4
event-log configuration 18-12
export policy configuration 16-51
firewall access configuration 18-7
Frame Relay interface 12-2
Frame Relay PVC 12-9
FRoIP configuration 19-5
FRoIP transport configuration 19-10
FTP policy configuration 18-19
FTP proxy configuration 18-16
HDLC configuration 13-1
HTTP policy configuration 18-36
HTTP proxy configuration 18-33
import policy configuration 16-64
IPNOS configuration 3-38
IPSec dialup configuration 17-30
IPSec ISAKMP configuration 17-39
IPSec manual configuration 17-47
ISAKMP policy configuration 17-23
ISAKMP server configuration 17-9
L2TP configuration 16-80
L2TP tunnel configuration 16-97, 16-100
OC-3 blade configuration 10-17, 11-2
OSPF configuration 20-50
plug policy configuration 18-55
plug proxy configuration 18-53
POP3 policy configuration 18-63
POP3 proxy configuration 18-60
POS interface configuration 10-17, 11-2
PPP configuration 14-2
PPTP configuration 16-114
QoS configuration 16-118, 21-57
RIP configuration 20-56
rsh policy configuration 18-70
rsh proxy configuration 18-67
SMTP policy configuration 18-78
SMTP proxy configuration 18-74
SQL policy configuration 18-91
SQL proxy configuration 18-88
status panel modem port connection 3-29
telnet policy configuration 18-99
telnet proxy configuration 18-96
transform set configuration 17-4
Unchannelized DS3 Blade configuration 6-4
Unchannelized E3 Blade configuration 8-3
VI configuration 21-11
VPN configuration 15-6
VR configuration 16-37
distinguished name
used with crypto maps 17-34, 17-42
dist-name parameter
set local-isakmp-identity command 17-34, 17-42
DLCI 12-10
creating one in FRoIP 19-11
setting identification number 12-10
dlci command 19-11
dlci parameter
route command 19-16

DNS 16-43
dns command 16-43
domain name 16-43
domain name servers
    See DNS
drain parameter
    lns command 16-90, 16-92
DS
    using VR ds-byte command 16-44, 16-119
DS3
    blade 5-6
    provisioning a Frame Relay PVC 12-8
ds-byte 16-44
ds-byte command 16-44
    compared to ACL tos command 16-44
    compated to tos 16-14
ds-byte-filter command 16-119, 21-58
ds-byte-filter on a VI
    compared to ds-byte on a VR 21-58
    compared to VI packet-filter 21-58
ds-byte-filter on VI 21-58
DSS 17-22
dss-sig parameter
    authentication command 17-22
dstaddr 16-9
dstaddr command 16-9
dstport 16-11
dstport command 16-11
dte parameter
    frame-relay intf-type command 12-3
Dual static NAT 16-68
duplex 4-5
duplex command 4-5
Dynamic NAT 16-68
dynamic-pkt-filter command 18-21

E

e1 command 7-6
E1 interface 7-6
E3
    provisioning a Frame Relay PVC 12-8
E3 framer 8-7
EBGP-multihop parameter
    neighbor command 20-12
egp-as 16-52
emergency parameter
    area command 14-3, 21-16
enable command 10-33
enable parameter
    admin-status command 19-9
    atm command 10-4
    lns command 16-90, 16-92
    oc3 command 9-2
    ppp command 14-7
    snmp-server command 16-145
    syslogd command 16-152
enable traps parameter
    snmp-server command 16-145
Encapsulated Security Protocol
    See ESP
encapsulation 4-6, 5-20, 6-5, 7-18, 8-4
encapsulation command 4-6, 5-20, 6-5, 7-18, 8-4, 10-34, 11-7
Encrypt user data packets using transfom-sets 17-3
encryption 16-25, 17-3
encryption and authentication type
    ah-md5-hmac 17-3
    ah-rfc1828 17-3
    ah-sha-hmac 17-3
    esp-3des 17-3
    esp3-md5-hmac 17-3
    esp3-sha-hmac 17-3
    esp-md5-hmac 17-3
    esp-rfc1829 17-3
    esp-sha-hmac 17-3
encryption and authentication types
    esp-des 17-3
encryption command 17-24
encryption parameter
    pptp command 16-115
end-address parameter
    access-list command 16-5
ESP 17-3
esp3-md5-hmac parameter
    transform-set command 17-3
esp3-sha-hmac parameter
    transform-set command 17-3
esp-des parameter
    transform-set command 17-3
esp-des-rfc1829 parameter
    transform-set command 17-3
esp-md5-hmac parameter
    transform-set command 17-3
esp-sha-hmac parameter
    transform-set command 17-3
Ethernet
    link 4-5
ethernet command 4-2
event-log command 21-16
event-window-size command 19-17
Excess burst size
    See BE
excess burst size
    See BE
EXEC parameter
    HTTP function 18-39
    HTTP transaction 18-43
executable languages 18-37
EXP parameter

HTTP packet header 18-41
expire-timer command 17-10
export parameter
neighbor command 20-12
export policy
protocol
export all routes 16-54
export BGP routes 16-54
export is-is routes 16-54
export local non-OSPF routes 16-54
export OSPF routes 16-54
export RIP routes 16-54
export static routes 16-54
export-policy 16-46, 20-54, 20-62
export-policy command 16-46, 20-54, 20-62

F

facility parameter
syslogd command 16-152
feature command 18-37
fiewall
access priority 18-5
FIFO 11-3
File Transfer Protocol 18-43
See FTP
filename 3-14
filesystem command 3-50
FilterID
See ds-byte-filter
filterid parameter
VI ds-byte-filter command 21-58
VR ds-byte-filter command 16-119
firewall 16-58
Firewall Commands 18-1
firewall parameter
VI packet-filter command 21-53
VR packet-filter command 16-108
Firewall proxies on a VI 21-53
flag 16-20
flag command 16-20, 17-10
flag parameter
set command 17-33, 17-41
FQDN
setting for crypto map identity 17-34, 17-42
fqdn parameter
set local-isakmp-identity command 17-34, 17-42
fragments 16-128
FRAM parameter
HTTP feature 18-37
Frame Relay 12-1
LMI
setting interface type 12-3
setting the CDE polling verification time 12-3
setting the DCE error threshold 12-3
setting the DCE monitored events counter 12-3
setting the DTE error threshold 12-3
setting the DTE monitored events counter 12-3
setting the DTE polling interval time 12-3
setting the LMI type 12-3
setting the polling interval 12-3
over an IP network 19-1
PVC
enforcing rates 12-10
setting committed burst size 12-10
setting committed information rate 12-10
setting DLCI number 12-10
setting excess burst size 12-10
setting protocol over Frame Relay 12-10
switching frame forwarding to another DLCI 12-10
frame-relay command 12-3, 12-10
framing 6-6
framing command 5-9, 6-6
framing parameter
oc3 command 9-2
FRoIP
DLCI
assigning a route to a DLCI 19-16
displaying current configuration 19-15
transport
setting keepalive message frequency 19-17
setting limit to bad or good events 19-18
setting packet buffer window size 19-19
setting the number of events 19-17
setting timeout for holding packets 19-19
froip command 16-57
FROM parameter
HTTP packet header 18-41
FTP 18-16
setting up a firewall proxy 18-15
FTP parameter
HTTP function 18-39
HTTP transaction 18-43
FTP proxy 18-16
ftp-proxy 18-15
ftp-proxy command 18-15
ftp-proxy parameter
fw command 18-15
full duple 4-5
fully qualified domain name
See FQDN
function command 18-22, 18-38
fw 16-58
fw command 16-58

G

genACLId

See ds-byte-filter
genaclid parameter
VI ds-byte-filter command 21-58
VI packet-filter command 21-53
VR ds-byte-filter command 16-119
VR packet-filter command 16-108
general access control list
see acl command
Global and Local Configurations 1-2
global commands 2-1
global commands compared to local commands 1-3
GRE tunnel 21-7, 21-63

H

half duplex 4-5
handoff command 18-25, 18-40
hard-kilobytes parameter
lifetime command 17-27
set security-association command 17-34, 17-42
hard-seconds parameter
lifetime command 17-27
set security-association command 17-34, 17-42
hash command 17-26
Hashed Message Authentication Code
See HMAC
hashed message authentication Codes
See HMAC
hashing engine 17-26
hashing scheme 17-11
HDLC 5-3, 7-2, 13-1
header command 18-41
hello-interval parameter
ip ospf command 21-26
Hello-intervals on a VI 21-26
help command 2-8
HELP parameter
FTP function 18-22
FTP transaction 18-26
high level data link control
See HDLC
HMAC 17-3
HMAC-MD5 17-3
HMAC-SHA-1 17-3
home-as 16-65
home-as command 16-65
hopcount parameter
traceroute command 16-162
hops 16-163
host parameter
snmp-server command 16-145
syslogd command 16-152
hostname 16-60
hostname command 16-60
HREQ parameter
HTTP function 18-39
HTTP transaction 18-43
HTM2 parameter
HTTP feature 18-37
HTML 18-37
HTML 2.0 18-37
HTML Frames 18-37
HTTP
setting up a firewall proxy 18-31
HTTP parameter
HTTP function 18-39
HTTP transaction 18-44
HTTP proxy 18-33
http-proxy 18-31
http-proxy command 18-31
http-proxy parameter
fw command 18-31
hw-address parameter
using arp command 16-16
HyperText Markup Language
See HTML
Hypertext Transfer Protocol
See HTTP

I

IANA 16-11
ICR 10-36, 10-56, 10-57
setting QoS to ABR 10-36
icr command 10-36
identity-pfs parameter
set flag command 17-33
idle-timeout parameter
pptp command 16-115
set command 17-33
if-name parameter
atm command 10-4
ignore-cr-payload parameter
flag command 17-11
IKE 16-18
ike-key parameter
set command 17-33, 17-41
ike-policy parameter
set command 17-33, 17-41
ILMI 10-4
ilmi-enable parameter
atm command 10-4
ilmi-vci parameter
atm command 10-4
ilmi-vpi parameter
atm command 10-4
IMOD parameter
HTTP packet header 18-41
import parameter
neighbor command 20-12

import policy
BGP origin type 16-65
import-policy 16-61, 20-63
import-policy command 16-61, 20-63
inbound ah key parameter
set session key command 17-48
inbound ah spi parameter
set session key command 17-48
inbound esp authenticator parameter
set session key command 17-49
inbound esp cipher parameter
set session key command 17-49
inbound esp spi parameter
set session key command 17-48
Inbound packets
See ds-byte-filter
inbound parameter
VI ds-byte-filter command 21-58
VI packet-filter command 21-53
VR ds-byte-filter command 16-119
VR packet-filter command 16-108
inform parameter
level command 14-4, 18-13, 21-16
initial cell rate
See ICR
initvector 17-5
initvector command 17-5
integrated local management interface
See ILMI
Intenet Protocol version 4 17-34, 17-42
interface command 3-57
interface-dlci parameter
frame-relay command 12-10
Internet Assigned Numbers Authority
See IANA
Internet Protocol Security
See IPSec
Internet Security Association Key Management Protocol
See ISAKMP
Internet service provider
See ISP
internetkey exchange
See IKE
intf-type parameter
frame-relay command 12-3
IP 16-68
address 16-9
packet fragments 16-128
reassembly timeout 16-128
setting parameters on a VI 21-26
IP address 10-5, 16-43, 16-56, 16-112, 16-152, 16-162
IP Address pool 16-68
ip command 16-68, 21-26
IP in IP tunnel 21-7, 21-63
described 21-7, 21-63
IP packet 16-128
IP routing 16-68
ip-address parameter
arp command 16-16
ip-local-pool-name parameter
ppp command 16-93
ipnos-diag command 3-30
ip-pool parameter
pptp command 16-115
IPSec 16-25, 17-1
ipsec command 17-3
IPSec Dialup Crypto Map 16-25
IPSec ISAKMP Crypto Map 16-25
IPSec Manual Crypto Map 16-25
ipsec parameter
area command 14-3, 21-16
IPSec Transform Set 16-25
IPSec transform set 17-4
ipsec transform-set parameter
crypto command 16-25
ipsec-dialup
crypto command 16-25
ipsec-dialup command 17-30
IPSEC-dialup crypto map 17-29
ipsec-dialup parameter
crypto map command 17-29
ipsec-isakmp 17-39
crypto command 16-25
ipsec-isakmp command 17-39
IPSEC-ISAKMP crypto map 17-29
ipsec-isakmp crypto map 17-39
ipsec-isakmp parameter
crypto map command 17-29
ipsec-manual 17-46
crypto map 17-47
ipsec-manual command 17-46
IPSEC-Manual crypto map 17-29
ipsec-manual parameter
crypto map command 17-29
IPSX 9000 Switch 3-1
IPSX 9000™ xxiii
Ip-tunnel parameter
connected-to command 21-7, 21-63
IPv4 address for identifying a crypto map 17-34, 17-42
ipv4 parameter
set local-isakmp-identity command 17-34, 17-42
ISAKMP 16-25, 17-8, 17-41
allowing older certificate request format 17-11
allowing older DA padding scheme 17-11
allowing older ID in hash 17-11
allowing older message padding scheme 17-11
allowing older payload padding scheme 17-11
allowing padding zeros in SPIs 17-11
compatibility with older devices 17-8
ignoring all certificate requests 17-11

multiple policies 17-20
server 17-18
setting anti-clogging tokens 17-14
setting DSS signatures 17-22
Setting older packet encryption format 17-11
setting pre-shared keys 17-22
setting preshared keys with a RADIUS authentication server 17-22
setting priority number for each policy 17-20
setting RSA signatues with a RADIUS authentication server 17-22
setting RSA signatures 17-22
isakmp 17-8
isakmp command 17-8
ISAKMP Parameter 16-25
isakmp parameter
crypto command 16-25
ISAKMP Policy 16-25
isakmp policy
crypto command 16-25
isis parameter
protocol command 16-54
ISP 16-164

J

J0 byte
used with SONET section tracing 9-2
J0 parameter
oc3 command 9-2
J1 byte
used with OC-3 path tracing 9-2, 9-3
Java 18-37
JAVA parameter
HTTP feature 18-37
JavaScript 18-37

K

KANJ parameter
HTTP feature 18-37
Kanji 18-37
keepalive parameter
ppp command 14-7, 16-93
keep-alive-period command 19-17
key point, explanation of xxv

L

L2TP 16-78, 16-80
setting the username command 16-165
setting up IPSX switch hostname 16-60
l2tp command 16-78, 16-82
L2TP Network Server
See LNS
LAN emulation
See LANE
LANE 10-56
largest possible unit of data transmitted from a VI 21-26
LDAP
setting up a firewall proxy 18-47
ldap-proxy command 18-47
ldap-proxy parameter
fw command 18-47
LEGK parameter
HTTP feature 18-37
length parameter
ping command 16-112
Level 2 tunneling protocol
See L2TP
level command 18-13
level parameter
syslogd command 16-152
lifetime command 17-27
lightweight directory access protocol
See LDAP
linecode command 7-10
link parameter
area command 14-3, 21-16
Link state metric on a VI 21-26
LIST parameter
FTP function 18-22
FTP transaction 18-26
list parameter
radius-server command 16-122
listen command 18-17, 18-34, 18-54, 18-61, 18-68, 18-75, 18-89, 18-97
listId parameter
access-list command 16-5
LLC 10-34
LMI
configuring in Frame Relay 12-3
lmi-n391dte parameter
frame-relay command 12-3
lmi-n392dce parameter
frame-relay command 12-3
lmi-n392dte parameter
frame-relay command 12-3
lmi-n393dce parameter
frame-relay command 12-3
lmi-n393dte parameter
frame-relay command 12-3
lmi-t391dce parameter
frame-relay command 12-3
lmi-t391dte parameter
frame-relay command 12-3
lmi-type parameter
frame-relay command 12-3
LMOD parameter
HTTP packet header 18-41
LNK parameter

HTTP packet header 18-41
lnotes-proxy command 18-48
lnotes-proxy parameter
fw command 18-48
LNS 16-90, 16-91
lns command 16-91
load command 2-10
local commands compared to global commands 1-3
local management iInterface
See LMI
Local Management Interface
See LMI
local parameter
protocol command 16-54
local pool parameter
ip command 16-69
local secret 17-14, 17-19
local-domain command 18-78
local-isakmp-identity parameter
set command 17-34, 17-42
local-secret-size command 17-14
local-vi command 19-6
logged command 18-26, 18-43
logging command 18-80
logical interfaces 3-1
Logical Link Control
See LLC
logical link parameter
connected-to command 21-7, 21-63
login 2-17
logout command 2-11
log-relay command 18-79
log-spam command 18-80
loop parameter
oc3 command 9-2
loopback command 5-10, 6-7, 7-11, 8-7
loopback parameter
connected-to command 21-7, 21-63
Lotus Notes
setting up a firewall proxy 18-48

M

MAC addresses 16-16
mail-hub command 18-81
main-mode parameter
set flag command 17-33, 17-41
major parameter
level command 14-3, 18-13, 21-16
management information base
See MIB
map 17-29
map command 17-29
match 17-32, 17-40, 17-47
match command 17-32, 17-40, 17-47
max-bytes command 18-81
max-conn parameter
pptp command 16-115
maxhops parameter
dhcp-relay command 16-29
Maximum Burst Size
See MBS
maximum burst size
See MBS
maximum transmittal unit
See MTU
max-packet-size parameter
oc3 command 9-2
max-recip command 18-82
max-retry-timer command 17-14
max-sa-count command 17-15
max-sessions-per-tunnel parameter
l2tp command 16-82
max-transport command 19-7
max-tunnels parameter
l2tp command 16-82
maxvci-bits parameter
atm command 10-4
maxvc-number parameter
atm command 10-4
maxvpi-bits parameter
atm command 10-5
maxvp-number parameter
atm command 10-4
MBS 10-38, 10-57
mbs command 10-38
MCR 10-40, 10-56, 10-57
setting QoS to ABR 10-40
mcr command 10-40
MD5 17-3
used with ISAKMP policies 17-26
using with anti-clogging tokens 17-20
md5 parameter
hash command 17-26
token-hash-type command 17-20
mdl 8-9
media access control
See MAC
Message Digest 5
See CHAPMD5
See MD5
message padding 17-11
metric 16-52
value 16-52
metric command 16-52
MIB 16-145
Microsoft 1-2
Microsoft Challenge Handshake Authentication Protocol, Version 1
See MSCHAPV1

Microsoft SQL Server
setting up a firewall proxy 18-87
MIME 18-41
MIME parameter
HTTP packet header 18-41
mime-check command 18-82
minimum cell rate
See MCR
minor parameter
level command 14-3, 18-13, 21-16
MKD parameter
FTP function 18-22
FTP transaction 18-26
MLPPP 14-9
link bundle to VI 21-7, 21-63
MLPPP Bundle
assign to a VI 21-7, 21-63
mode 17-6
mode command 17-6
MODE parameter
FTP function 18-22
FTP transaction 18-27
MPF parameter
HTTP feature 18-37
MSCHAPV1 16-115
used in PPP 14-7
mschapv1 parameter
ppp authentication command 14-7
MTU
setting on a VI 21-26
mtu parameter
ip command 21-26
Multi Part Forms 18-37
multi-exit discriminator
preference command 16-67
Multilink PPP
See MLPPP
multimedia 10-56
multipurpose internet mail extension
See MIME

N

N parameter
arp command 16-16
name 15-7
name command 15-7, 16-106
nat
inside destination static parameter
ip command 16-69
inside source static parameter
ip command 16-69
list parameter
ip command 16-70
pool parameter
ip command 16-70
national bit command 7-13
neighbor command 20-12
neighbor-ip-address parameter
atm command 10-5
neighbor-name parameter
atm command 10-5
NetMeeting
setting up a firewall proxy 18-49
netm-proxy command 18-49
netm-proxy parameter
fw command 18-49
Netscape 18-37
Network Management System
See NMS
Network News Transfer Protocol
See NNTP
network operations personnel xxiii
next-hop-self parameter
neighbor command 20-12
NLST parameter
FTP function 18-22
FTP transaction 18-27
NMS
used with OC-3 blades 9-2
NNTP
setting up a firewall proxy 18-50
nntp-proxy command 18-50
nntp-proxy parameter
fw command 18-50
no 8-9, 16-151
no parameter
flag command 17-10
ip command 21-27
no-advertise parameter
bgp-community command 16-49
no-export parameter
bgp-community command 16-49
no-export-sub parameter
bgp-community command 16-49
none parameter
frame-relay lmi-type command 12-3
logging command 18-80
log-relay command 18-79
log-spam command 18-80
non-real-time variable bit rate
See NRT-VBR
NOOP parameter
FTP function 18-22
FTP transaction 18-27
normal parameter
logging command 18-80
note, explanation of xxv
NRT-VBR 10-38, 10-55, 10-56, 10-57

O

OAM 10-41
oam-loopback command 10-41
OC-3
    enable data scrambling 9-3
    enable physical interface 9-2
    sending alarms to NMS 9-2
    setting clock source 9-2
    setting CRC 9-2
    setting J0 trace byte 9-2
    setting local PTE physical path trace label 9-2
    setting maximum POS packet size 9-2
    setting remote PTE physical path trace label 9-3
    setting SONET or SDH 9-2
    setting to loopback 9-2
oc3 command 9-2
OC-3 performance monitoring parameters 10-17
OC-3 POS
    setting CRC 9-2
    setting maximum packet size 9-2
OC-3c 9-1
old-cr-format parameter
    flag command 17-11
old-da-padding parameter
    flag command 17-11
old-id-in-hash parameter
    flag command 17-11
old-message-padding parameter
    flag command 17-11
old-payload-padding parameter
    flag command 17-11
old-pk-encryption parameter
    flag command 17-11
Open Shortest Path First
    See OSPF
operations administration and maintenance
    See OAM
Optical Carrier 3
    See OC-3c
origin 16-65
origin command 16-65
OSI 10-56
OSPF 20-46
    area identification number 20-46
    export-policy 20-54
    set area on a VI 21-26
    setting password on a VI 21-26
    setting route cost on VI 21-26
ospf 20-46
ospf command 20-46
OSPF interface priority on a VI 21-26
ospf parameter
    ip command 21-26
    protocol command 16-54
ourname command 18-45
outbound ah key parameter
    set session key command 17-48
outbound ah spi parameter
    set session key command 17-48
outbound esp authenticator parameter
    set session key command 17-49
outbound esp cipher parameter
    set session key command 17-49
outbound esp spi parameter
    set session key command 17-49
Outbound packets
    See ds-byte-filter
Outbound packets from a VI 21-53
outbound parameter
    VI ds-byte-filter command 21-58
    VI packet-filter command 21-53
    VR ds-byte-filter command 16-119
    VR packet-filter command 16-108

P

packet 16-7, 16-52, 16-163
packet filters 16-108
Packet Over SONET
    See POS
Packet over SONET
    See POS
packet-filter 16-108, 21-53
packet-filter command 16-108, 21-53
    assigning an ACL 16-108
padding scheme 17-11
PAP 16-115
    used in PPP 14-7
pap parameter
    ppp authentication command 14-7
parameter command 17-8
PASS parameter
    FTP function 18-22
    FTP transaction 18-27
passive parameter
    neighbor command 20-12
password
    setting on VI for OSPF 21-26
Password Authentication Protocol
    See PAP
password parameter
    neighbor command 20-12
PASV parameter
    FTP function 18-22
    FTP transaction 18-27
path terminating equipment
    See PTE
path-trace parameter
    oc3 command 9-2

PCR 10-36, 10-41, 10-55, 10-56, 10-57
pcr command 10-41
peak cell rate
    See PCR
peer 16-53
    peer-as 16-53, 16-66
peer-as command 16-53, 16-66
peer-group parameter
    neighbor command 20-12
Perl 18-37
permanent virtual circuit
    See PVC
permit 16-47
permit parameter
    access-list command 16-5
    action command 18-6
    VI packet-filter command 21-53
    VR packet-filter command 16-108
permit-bad-event command 19-18
permitting packets into a VI 21-53
pfs parameter
    set flag command 17-33, 17-41
pfs-group parameter
    set command 17-34, 17-42
physical commands 1-3
ping 10-41, 16-112
ping command 16-112
plug-proxy command 18-51
plug-proxy parameter
    fw command 18-51
Point to Point Protocol
    See PPP
Point to point tunneling protocol
    See PPTP
Point-to-Point Protocol
    See PPP
Point-to-Point Tunneling Protocol
    See PPTP
policy command
    access policy name 18-7
    crypto ISAKMP policy 17-20
    FTP proxy name 18-18
    HTTP proxy name 18-35
    plug proxy name 18-54
    pop3 proxy name 18-63
    rsh proxy name 18-69
    SMTP proxy name 18-76
    SQL proxy name 18-90
    telnet proxy name 18-98
POP3
    setting up a firewall proxy 18-59
pop3 18-64
pop3-proxy command 18-59
pop3-proxy parameter
    fw command 18-59
pop3-server command 18-64
port
    shutting down on channelized DS3 blade 5-12
    shutting down on channelized E3 blade 7-14
port command 18-8
PORT parameter
    FTP function 18-22
    FTP transaction 18-27
port parameter
    route command 19-16
POS 11-1
    provisioning a Frame Relay PVC 12-7
pos command 11-2, 12-7
POS monitoring statistics 11-2
PPP 16-93
    enable or disable 14-7
    keepalive 14-7
    setting authentication types 14-7
    setting idle timeout 14-7
    setting number of retransmission attempts 14-7
    setting number of termination requests 14-7
    setting reauthentication value 14-7
    setting the username command 16-165
ppp command 14-7, 16-93
PPTP 16-113
    setting the username command 16-165
    setting up IPSX switch hostname 16-60
pptp 16-113
pptp command 16-113, 16-115
preference 16-67
preference command 16-67
pre-share parameter
    authentication command 17-22
pre-shared key 17-21
PRGM parameter
    HTTP packet header 18-41
private-key 16-20
private-key command 16-20
protocol 16-11, 16-54
protocol command 10-50, 16-11, 16-54
protocol parameter
    area command 14-3, 21-16
    frame-relay command 12-10
protocol value 16-11
proxy 18-16
PTE
    local end 9-2
    remote end 9-3
PVC 10-40, 12-3
    provision on a POS blade 12-7
    provisioning on a serial interface 12-8
PWD parameter
    FTP function 18-22
    FTP transaction 18-27

Q

Q933A
    setting Frame Relay LMI type 12-3
q933a parameter
    frame-relay lmi-type command 12-3
QoS 16-118, 21-57
    setting to ABR 10-56
qos command 16-118, 21-57
Quality of Service 10-56
quality of service
    See QoS
    See qos
QUIT parameter
    FTP function 18-22
    FTP transaction 18-27

R

RADIUS 16-115
    used with authenticating ISAKMP pre-shared keys 17-22
    used with authenticating ISAKMP RSA signatures 17-22
RADIUS protocol 16-122
RADIUS Server 16-115
radius-server 16-122
radius-server command 16-122
range parameter
    area command 20-47
rate parameter
    rate-limit command 21-61
rate-limit command 21-61
rate-police parameter
    frame-relay command 12-10
RC4 16-115
READ parameter
    HTTP function 18-39
    HTTP transaction 18-44
Real-Time Variable Bit Rate
    See RT-VBR
real-time variable bit rate
    See RT-VBR
reassembly 16-128
reassembly command 16-128
reauth parameter
    ppp command 14-7, 16-93
reboot command 2-12
receive-size command 10-54
reference, explanation of xxv
REFR parameter
    HTTP packet header 18-41
REIN parameter
    FTP function 18-22
    FTP transaction 18-27
reject parameter
    log-relay command 18-79
    log-spam command 18-80
relay-host command 18-83
relay-reply command 18-84
remote 21-63
    See also connected-to
remote as parameter
    neighbor command 20-12
remote command 21-63
remote server 17-17, 17-18
remote shell
    See RSH
remote VPN clients 17-29
remoteauthentication dial in user service
    See RADIUS
remote-path-trace parameter
    oc3 command 9-3
remote-port command 18-56, 18-92
remote-server command 18-57, 18-71, 18-92
rename command 3-55
reset-blade command 3-7
REST parameter
    FTP function 18-22
    FTP transaction 18-27
RETA parameter
    HTTP packet header 18-41
RETR parameter
    FTP function 18-23
    FTP transaction 18-27
    HTTP function 18-39
    HTTP transaction 18-44
retransmission 16-122
retransmit parameter
    radius-server command 16-122
retransmit-interval on a VI 21-26
retransmit-interval parameter
    ip ospf command 21-26
retry a transmission 17-16
retry timer for ISAKMP 17-14
retry-limit 17-16
retry-timer-sec 17-17
retry-timer-usec 17-18
REV1
    setting Frame Relay LMI type 12-3
rev1 parameter
    frame-relay lmi-type command 12-3
RFC 1828 17-3
RFC 1829 17-3
RFC 1853 21-7, 21-63
RFC 2663 16-68
RIP 20-56
    rip-ip 16-56
rip 20-56
rip command 20-56
rip parameter

protocol command 16-54
RIP-IP 16-56
rip-ip command 16-56
Rivest-Shamir-Adleman
See RSA
RMD parameter
FTP function 18-23
FTP transaction 18-27
RNFR parameter
FTP function 18-23
FTP transaction 18-27
RNTO parameter
FTP function 18-23
FTP transaction 18-27
route command 19-16
route parameter
frame-relay command 12-10
ip command 16-70
router 16-16, 16-129, 20-3, 21-66
router command 16-129, 21-66
Router Information Protocol 16-56
Routing Information Protocol
See RIP
routing parameter
IP command 16-70
RSA 17-22
digital signature 17-22
rsa-sig parameter
authentication command 17-22
RSH 18-56
setting up a firewall proxy 18-66
rsh-proxy command 18-66
rsh-proxy parameter
fw command 18-66
RT-VBR 10-38, 10-55, 10-57

S

SA 17-15
SA-Per-Host 17-41
sa-per-host parameter
set flag command 17-41
sa-per-net parameter
set flag command 17-33, 17-41
save command 2-14
SCR 10-55, 10-57
scr command 10-55
scramble parameter
oc3 command 9-3
SCRP parameter
HTTP feature 18-37
SDH B-7
SDH parameter
oc3 framing command 9-2
SDU 10-54, 10-65
secret-recreate command 17-19
section trace byte
See J0 byte
secure hash algorithm
See SHA-1
security flag 16-20
Security Parameter Index
see SPI
security parameter index
See SPI
security-association parameter
set command 17-34, 17-42
serial command 5-17, 6-2, 7-15, 8-2, 12-8
SERV parameter
HTTP packet header 18-41
server parameter
dhcp-relay command 16-29
pptp command 16-113
service data units
See SDU
Service Management System
See SMS
Service Processing Switch
See SPS xxiii, 1-1
service provider parameter
type command 16-164
service-category command 10-56
service-vi parameter
l2tp command 16-82
session command 16-98
session parameter
l2tp command 16-83, 16-84
set 17-48
set command 17-33, 17-40, 17-48
Setting ISAKMP Phase I rules 17-20
Setvalue
See ds-byte-filter
setvalue parameter
VI ds-byte-filter command 21-58
VR ds-byte-filter command 16-119
sha parameter
hash command 17-26
SHA-1 17-3
used with ISAKMP policies 17-26
using with anti-clogging tokens 17-20
sha1 parameter
token-hash-type command 17-20
show command 2-15
show mode 1-3
shutdown 5-12, 6-8, 7-14, 8-9
shutdown command 5-12, 6-8, 7-14, 8-9
shutdown parameter
neighbor command 20-12
Simple Network Management Protocol
See SNMP

SITE parameter
FTP function 18-23
FTP transaction 18-27
SIZE parameter
FTP function 18-23
FTP transaction 18-27
slot parameter
route command 19-16
SMDS 10-56
SMNT parameter
FTP function 18-23
FTP transaction 18-27
SMS 1-2
SMTP
setting up a firewall proxy 18-73
smtp-proxy command 18-73
smtp-proxy parameter
fw command 18-73
SNMP 16-144
SNMP server 16-145
snmp-server 16-144
snmp-server command 16-144
SONET 11-1
SONET parameter
oc3 framing command 9-2
source command 18-9
source ip address
See srcaddr
source parameter
ping command 16-112
traceroute command 16-162
source port
See srcport
spam-host command 18-84
spam-reply command 18-85
speed 4-7
speed command 4-7
SPI 17-11, 17-48
Spreading VR workload across multiple blades. 15-2
SPS xxiii, 1-1
sql-proxy command 18-87
sql-proxy parameter
fw command 18-87
srcaddr 16-12
srcaddr command 16-12
srcport 16-14
srcport command 16-14
start-address parameter
access-list command 16-5
STAT parameter
FTP function 18-23
FTP transaction 18-27
Static NAT 16-68
static parameter
protocol command 16-54
Static route 16-68
status command 4-8
status panel command 3-27
status parameter
dhcp-relay command 16-29
STOR parameter
FTP function 18-23
FTP transaction 18-27
HTTP function 18-39
HTTP transaction 18-44
STOU parameter
FTP function 18-23
FTP transaction 18-27
STRU parameter
FTP function 18-23
FTP transaction 18-27
subnet mask 16-9, 16-12
SUN Microsystem 18-37
sustainable cell rate
See SCR
switched multimegabit data service
See SMDS
Sybase SQL Server
setting up a firewall proxy 18-87
symbols xxv
Synchronous Digital Hierarchy
See SDH
see SDH
Synchronous Optical Network
See SONET
syntax xxiii
syslogd 16-151
syslogd command 16-151
sysname command 2-16
SYST parameter
FTP function 18-23
FTP transaction 18-27
system parameter
area command 14-3, 21-16
system-up-time command 3-59
sysuser command 2-17

T

t1 5-13
t1 command 5-13
T1 framer 5-10, 6-7, 7-11
T1 interface 5-13
TCP/IP 10-57
telnet 16-157
telnet command 16-157
telnet-proxy command 18-95
telnet-proxy parameter
fw command 18-95
terminal emulation 1-2

term-req-retransmit parameter
ppp command 14-7, 16-93
TEXT parameter
HTTP function 18-39
HTTP transaction 18-44
TFTP 3-8, 3-10
tftpget command 3-8
tftpput command 3-10
threshold parameter
dhcp-relay command 16-29
time command 3-60
time to live
See TTL
time to transmit a link-state update packet over the VI 21-26
timeout command 18-29, 18-46, 18-57, 18-65, 18-71, 18-85, 18-93, 18-100
timeout parameter
ping command 16-112
ppp command 14-7, 16-93
radius-server command 16-122
timers parameter
neighbor command 20-12
time-to-live
See TTL
TITL parameter
HTTP packet header 18-41
token-hash-type 17-20
top command 2-18
TOS 16-14
tos 16-14
tos command 16-14
compared to VR ds-byte command 16-44
traceroute 16-162
traceroute command 16-162
transform-set command 17-3
transform-set parameter
set command 17-34, 17-42
set session key command 17-49
Transmit-delay on a VI 21-26
transmit-delay parameter
ip ospf command 21-26
transmit-size command 10-65
transport command 19-8
transport mode 17-6
trap parameter
area command 14-3, 21-16
Triple-DES 17-3
triple-DES encryption 17-24
Trivial File Transfer Protocol
See TFTP
TTL 16-27, 16-112, 16-163
ttl 16-163
ttl command 16-163
ttl parameter
ping command 16-112
tunnel command 16-96, 16-99
tunnel parameter
connected-to command 21-7, 21-63
l2tp command 16-85
tunnel-keepalive command 21-69
type command 16-164
Type Of Service
setting the octet on a VI using ds-byte-filter 21-58
type of service
See TOS
TYPE parameter
FTP function 18-23
FTP transaction 18-27
HTTP function 18-39
HTTP transaction 18-44
type parameter
arp command 16-16
typographical conventions xxv

U

UBR 10-56, 10-57
Unchannelized E3 Blade 8-1
uniform resource locator
See URL
universal time code
See UTC
UNIX 1-2
unnumbered parameter
ip command 21-26
unspecified bit rate
See UBR
up command 2-19
update-source parameter
neighbor command 20-12
URL parameter
HTTP packet header 18-41
USER parameter
FTP function 18-23
FTP transaction 18-27
HTTP packet header 18-42
username 16-165
username command 16-165
userpriority command 4-10
use-zero-spi parameter
flag command 17-11
UTC 3-60

V

VC 10-34, 10-69
VCC 10-4, 10-5
VCI 10-5, 10-66
VCI bits 10-5
vci command 10-66

verbose-load command 3-62
version command 2-20
VI 3-37, 16-24, 16-28, 16-108, 16-167
connected-to
GRE tunnel connections 21-7, 21-63
IP in IP tunnel connections 21-7, 21-63
logical interface connections 21-7, 21-63
loopback connections 21-7, 21-63
tunnel connections 21-7, 21-63
virtual interface connections 21-7, 21-63
display current configuration 21-11
OSPF
retransmit-interval 21-26
set area on a VI 21-26
setting dead-interval 21-26
setting hello-interval 21-26
setting password on a VI 21-26
setting priority on a VI 21-26
setting route cost 21-26
transmit-delay 21-26
packet-filter
Deny a packet into a VIDeny a packet into a VI 21-53
Inbound packetsInbound packets to a VI 21-53
Outbound packets from a VIOutbound packets from a VI 21-53
permit a packet into a VIpermitting packets into a VI 21-53
Use a Firewall proxyFirewall proxies on a VI 21-53
using ACLs 21-53
set broadcast address 21-26
setting IP address and Netmask 21-26
setting MTU 21-26
setting OSPF on a VI 21-26
use an unnumbered IP address 21-26
vi 16-167
VI command 21-1
vi command 16-167
viId parameter
access-list command 16-5
arp command 16-16
virtual channel connection
See VCC
virtual channel identifier
See VCI
virtual commands 1-3
virtual connection
See VC
virtual interface
See VI
see VI
virtual interface parameter
connected-to command 21-7, 21-63
Virtual Local Area Network
See VLAN
virtual path connection
See VPC
virtual path identifier
See VPI
virtual router
See VR
VLAN 4-9, 4-10, 4-11
vlanid command 4-9
vlanname command 4-11
voice over IP 10-57
VPC 10-4, 10-5
VPI 10-5, 10-69
VPI bits 10-5
vpi command 10-69
VPN 3-37
vpn command 3-63
VR 3-37, 15-11, 16-1, 16-24, 16-28, 16-47, 16-108
creating a VR 15-2
vr 15-11
vr command 15-11
vsio command 3-28

W

warning, explanation of xxv
web extensions 18-37
Windows NT 1-2
window-size command 19-19
window-timeout command 19-19
World Wide Web
See WWW
WRTE parameter
HTTP function 18-39
HTTP transaction 18-44
WWW parameter
HTTP packet header 18-42

X

X509 distinguished name 17-34, 17-42
xauth-init-preshared parameter
authentication command 17-22
xauth-init-rsa-sig parameter
authentication command 17-22
XCUP parameter
FTP function 18-23
FTP transaction 18-27
XCWD parameter
FTP function 18-23
FTP transaction 18-27
XMKD parameter
FTP function 18-23
FTP transaction 18-27
XPWD parameter
FTP function 18-23

FTP transaction 18-27
XRMD parameter
FTP function 18-23
FTP transaction 18-27