Cisco Service Control Engine (SCE) Software Configuration Guide

Version 2.5.7
OL-7827-02

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Customer Order Number: DOC-7827-02
Text Part Number: OL-7827-02

Printed in the USA on recycled paper containing 10% postconsumer waste.

Cisco Service Control Engine (SCE) Software Configuration Guide

CONTENTS

Preface xvii

- Audience xvii
- Organization xvii
- Related Publications xviii
- Document Conventions xix
- Obtaining Documentation xx
  - Cisco.com xx
  - Documentation Feedback xx
- Obtaining Technical Assistance xx
  - Cisco Technical Support Website xxi
  - Submitting a Service Request xxi
  - Definitions of Service Request Severity xxi
- Obtaining Additional Publications and Information xxii

Introduction 1-1

- SCE Platform Management Interfaces 1-2

Command-Line Interface 2-1

- Getting Help 2-2
- Authorization and Command Levels (Hierarchy) 2-2
  - CLI Command Hierarchy 2-3
  - CLI Authorization Levels 2-10
  - Prompt Indications 2-12
  - Syntax and Conventions 2-12
  - Login and User Levels 2-13
- CLI Help Features 2-13
  - Partial Help 2-13
  - Argument Help 2-14
  - The [no] Prefix 2-14

Navigational and Shortcut Features 2-15
Command History 2-15
Keyboard Shortcuts 2-15
Tab Completion 2-16
FTP User Name and Password 2-16
Managing Command Output 2-17
Scrolling the Screen Display 2-17
Filtering Command Output 2-17
Redirecting Command Output to a File 2-18
CLI Scripts 2-18

Operations 3-1
Managing Configurations 3-1
Viewing Configuration 3-1
Saving the Configuration Settings 3-4
Recovering a Previous Configuration 3-6
Upgrading SCE Platform Firmware 3-7
Configuring Applications 3-8
Installing an Application 3-9
Configuring the Currently Installed Application 3-11
Rebooting and Shutting Down the SCE Platform 3-12
Rebooting the SCE Platform 3-12
Shutting Down the SCE Platform 3-12

Utilities 4-1
Setup Utility 4-1
Entering the Setup Utility 4-1
Multiple entry parameters (Lists) 4-2
File-system Operations 4-3
Working with Directories 4-3
Working with Files 4-5
The User Log 4-8
The Logging System 4-8
Generating a File for Technical Support 4-11

Configuring the Management Interface and Security 5-1

Configuring the Available Interfaces 5-1

Configuring Access Control Lists (ACLs) 5-1

Telnet Interface 5-4

SSH Server 5-5

SNMP Interface 5-8

SNMP Configuration and Management 5-9

SNMP Protocol 5-9

Configuration via SNMP 5-10

Security Considerations 5-10

SNMP Community Strings 5-11

Traps 5-13

CLI 5-17

MIBs 5-17

MIB-II 5-18

Service Control Enterprise MIB 5-18

Passwords 5-20

Changing Passwords 5-20

Encryption 5-22

IP Configuration 5-22

IP Routing Table 5-22

IP Advertising 5-24

Setting the IP Address and Subnet Mask of the FastEthernet Management Interface 5-26

Time Clocks and Time Zone 5-26

Showing System Time 5-27

Showing Calendar Time 5-27

Setting the Clock 5-28

Setting the Calendar 5-28

Setting the Time Zone 5-28

Removing Current Time Zone Setting 5-29

Configuring Daylight Saving Time 5-29

SNTP 5-31

Enabling SNTP multicast client 5-32

Disabling SNTP multicast client 5-32
Enabling SNTP unicast client 5-32
Disabling SNTP unicast client 5-33
Defining the SNTP unicast update interval 5-33
Display SNTP information 5-34
Domain Name (DNS) Settings 5-34
Name Servers 5-35
Domain Name 5-36
Host Table 5-36
show hosts 5-37
Management Interface Configuration Mode 5-37
Configuring the Management Interface Speed and Duplex Parameters 5-37

Configuring the Line Interface 6-1
Configuring Tunneling Protocols 6-1
Selecting the Tunneling Mode 6-2
Displaying Tunneling Configuration 6-4
Configuring Traffic Rules and Counters 6-4
Traffic Rules 6-5
Traffic counters 6-5
Configuring Traffic Counters 6-6
Configuring Traffic Rules 6-6
Managing Traffic Rules and Counters 6-9
Configuring TOS Marking 6-11
Enabling and Disabling TOS Marking 6-11
Modifying the TOS Table 6-12
Line Ethernet Interfaces 6-12
Entering Ethernet Line Interface Configuration Mode 6-12
Configuring GigabitEthernet Auto-Negotiation 6-12

Configuring the Connection 7-1
Editing the Connection Mode 7-1
Link Mode 7-2
Forced Failure 7-4
Failure Recovery Mode 7-4

SCE Platform/SM Connection 7-5
Enabling and Disabling Link Failure Reflection 7-5
Enabling and Disabling Link Failure Reflection on All Ports 7-6

Configuring the RDR Formatter 8-1
The RDR Formatter 8-1
RDR Formatter Destinations 8-1
Categories 8-2
Priority 8-3
Forwarding Modes 8-4
Configuring the RDR Formatter 8-4
Displaying RDR Formatter Configuration and Statistics 8-10
Disabling the LineCard from Sending RDRs 8-11

Managing Subscribers 9-1
Subscriber Overview 9-2
Subscriber Modes in Service Control Solutions 9-3
Aging Subscribers 9-4
Anonymous Groups and Subscriber Templates 9-5
Subscriber Files 9-5
Importing/Exporting Subscriber Information 9-6
Importing/Exporting Subscribers 9-7
Importing/Exporting Subscriber Templates 9-7
Removing Subscribers and Templates 9-8
Importing/Exporting Anonymous Groups 9-10
Monitoring Subscribers 9-10
Monitoring the Subscriber Database 9-11
Displaying Subscribers 9-12
Displaying Subscriber Information 9-15
Displaying Anonymous Subscriber Information 9-16
Subscriber Traffic Processor IP Ranges 9-18
Subscriber Mapping Modes 9-19
Subscriber Mapping Conflicts 9-19
Subscriber Rules for TIRs 9-20
Configuring TIRs 9-21

Removing TIRs and Subscriber Mappings 9-22
Importing and Exporting TIRs 9-23
Monitoring TIRs 9-24
Subscriber Aging 9-26
SCE Platform/SM Connection 9-28

Redundancy and Fail-Over 10-1
Terminology and Definitions 10-2
Redundant Topologies 10-2
In-line Dual Link Redundant Topology 10-3
Failure Detection 10-3
Link Failure Reflection 10-4
Hot Standby and Fail-over 10-4
Hot Standby 10-4
Fail-over 10-5
Failure in the Cascade Connection 10-6
Installing a Cascaded System 10-6
Recovery 10-7
Replacing the SCE platform (manual recovery) 10-8
Reboot only (fully automatic recovery) 10-8
CLI Commands 10-9
Topology-Related Parameters for Redundant Topologies 10-9
Configuring the Connection Mode 10-9
Forced Failure 10-10
Monitoring the System 10-10
System Upgrades 10-11
Firmware Upgrade (package installation) 10-11
Application Upgrade 10-12
Simultaneous Upgrade of Firmware and Application 10-12

Identifying And Preventing Distributed-Denial-Of-Service Attacks 11-1
Attack Filtering 11-1
Attack Detection 11-2
Attack Detection Thresholds 11-3
Attack Handling 11-3

Subscriber Notification 11-4
Configuring Attack Detectors 11-5
Enabling Specific-IP Detection 11-7
Default Attack Detector 11-7
Specific Attack Detectors 11-8
Sample Attack Detector Configuration 11-10
Configuring Subscriber Notifications 11-11
Subscriber Notification Ports 11-11
Managing Attack Filtering 11-12
Preventing Attack Filtering 11-13
Forcing Attack Filtering 11-13
Monitoring Attack Filtering 11-14

Proprietary MIB Reference A-1
Service Control Enterprise MIB A-1
Using this Reference A-2
pcubeMgmt: pcubeConfigCopyMIB A-2
Config-Copy MIB Objects A-3
pcubeCopyIndex (pcubeCopyEntry 1) A-3
pcubeCopyEntryRowStatus (pcubeCopyEntry 2) A-3
pcubeCopySourceFileType (pcubeCopyEntry 3) A-3
pcubeCopyDestFileType (pcubeCopyEntry 4) A-3
pcubeWorkgroup: pcubeSeMIB A-4
pcubeSeEvents (pcubeWorkgroup 0) A-4
SCE Events A-4
pcubeSEObjs (pcubeWorkgroup 1) A-6
SCE-MIB Objects A-6
SCE-MIB Structure A-7
SCE Events: pcubeSeEvents A-14
operationalStatusOperationalTrap (pcubeSeEvents 1) A-14
operationalStatusWarningTrap (pcubeSeEvents 2) A-14
operationalStatusFailureTrap (pcubeSeEvents 3) A-14
systemResetTrap (pcubeSeEvents 4) A-14
chassisTempAlarmOnTrap (pcubeSeEvents 5) A-14

chassisTempAlarmOffTrap (pcubeSeEvents 6) A-14
chassisVoltageAlarmOnTrap (pcubeSeEvents 7) A-14
chassisFansAlarmOnTrap (pcubeSeEvents 8) A-14
chassisPowerSupplyAlarmOnTrap (pcubeSeEvents 9) A-15
rdrActiveConnectionTrap (pcubeSeEvents 10) A-15
rdrNoActiveConnectionTrap (pcubeSeEvents 11) A-15
rdrConnectionUpTrap (pcubeSeEvents 12) A-15
rdrConnectionDownTrap (pcubeSeEvents 13) A-15
loggerUserLogIsFullTrap (pcubeSeEvents 18) A-15
sntpClockDriftWarnTrap (pcubeSeEvents 19) A-15
linkModeBypassTrap (pcubeSeEvents 20) A-15
linkModeForwardingTrap (pcubeSeEvents 21) A-15
linkModeCutoffTrap (pcubeSeEvents 22) A-15
pcubeSeEventGenericString1 (pcubeSeEvents 23) A-15
pcubeSeEventGenericString2 (pcubeSeEvents 24) A-16
moduleAttackFilterActivatedTrap (pcubeSeEvents 25) A-16
moduleAttackFilterDeactivatedTrap (pcubeSeEvents 26) A-16
moduleEmAgentGenericTrap (pcubeSeEvents 27) A-17
linkModeSniffingTrap (pcubeSeEvents 28) A-17
moduleRedundancyReadyTrap (pcubeSeEvents 29) A-17
moduleRedundantConfigurationMismatchTrap (pcubeSeEvents 30) A-17
moduleLostRedundancyTrap (pcubeSeEvents 31) A-17
moduleSmConnectionDownTrap (pcubeSeEvents 32) A-17
moduleSmConnectionUpTrap (pcubeSeEvents 33) A-18
moduleOperStatusChangeTrap (pcubeSeEvents 34) A-18
portOperStatusChangeTrap (pcubeSeEvents 35) A-18
chassisLineFeedAlarmOnTrap (pcubeSeEvents 36) A-18
rdrFormatterCategoryDiscardingReportsTrap (pcubeSeEvents 37) A-18
rdrFormatterCategoryStoppedDiscardingReportsTrap (pcubeSeEvents 38) A-18
sessionStartedTrap (pcubeSeEvents 39) A-18
sessionEndedTrap (pcubeSeEvents 40) A-18
sessionDeniedAccessTrap (pcubeSeEvents 41) A-18
sessionBadLoginTrap (pcubeSeEvents 42) A-18
System Group: systemGrp (pcubeSEObjs 1) A-19

sysOperationalStatus (systemGrp 1) A-19
sysFailureRecovery (systemGrp 2) A-19
sysVersion (systemGrp 3) A-19
Chassis Group: chassisGrp (pcubeSEObjs 2) A-20
ChassisSysType (chassisGrp 1) A-20
chassisPowerSupplyAlarm (chassisGrp 2) A-20
chassisFansAlarm (chassisGrp 3) A-21
chassisTempAlarm (chassisGrp 4) A-21
chassisVoltageAlarm (chassisGrp 5) A-21
chassisNumSlots (chassisGrp 6) A-22
chassisSlotConfig (chassisGrp 7) A-22
chassisPsuType (chassisGrp 8) A-22
chassisLineFeedAlarm (chassisGrp 9) A-23
Module Group: moduleGrp (pcubeSEObjs 3) A-24
moduleTable (moduleGrp 1) A-24
moduleEntry (moduleTable 1) A-24
moduleIndex (moduleEntry 1) A-25
moduleType (moduleEntry 2) A-25
moduleNumTrafficProcessors (moduleEntry 3) A-25
moduleSlotNum (moduleEntry 4) A-26
moduleHwVersion (moduleEntry 5) A-26
moduleNumPorts (moduleEntry 6) A-26
moduleNumLinks (moduleEntry 7) A-26
moduleConnectionMode (moduleEntry 8) A-27
moduleSerialNumber (moduleEntry 9) A-27
moduleUpStreamAttackFilteringTime (moduleEntry 10) A-27
moduleUpStreamLastAttackFilteringTime (moduleEntry 11) A-27
moduleDownStreamAttackFilteringTime (moduleEntry 12) A-28
moduleDownStreamLastAttackFilteringTime (moduleEntry 13) A-28
moduleAttackObjectsClearTime (moduleEntry 14) A-28
moduleAdminStatus (moduleEntry 15) A-28
moduleOperStatus (moduleEntry 16) A-29
Link Group: linkGrp (pcubeSEObjs 4) A-30
linkTable (linkGrp 1) A-30

linkEntry (linkTable 1) A-30
linkModuleIndex (linkEntry 1) A-31
linkIndex (linkEntry 2) A-31
linkAdminModeOnActive (linkEntry 3) A-31
linkAdminModeOnFailure (linkEntry 4) A-31
linkOperMode (linkEntry 5) A-32
linkStatusReflectionEnable (linkEntry 6) A-32
linkSubscriberSidePortIndex (linkEntry 7) A-32
linkSubscriberSidePortIndex (linkEntry 8) A-32
Disk Group: diskGrp (pcubeSEObjs 5) A-33
diskNumUsedBytes (diskGrp 1) A-33
diskNumFreeBytes (diskGrp 2) A-33
RDR Formatter Group: rdrFormatterGrp (pcubeSEObjs 6) A-34
rdrFormatterEnable (rdrFormatterGrp 1) A-34
rdrFormatterDestTable (rdrFormatterGrp 2) A-34
rdrFormatterDestEntry (rdrFormatterDestTable 1) A-35
rdrFormatterDestIPAddr (rdrFormatterDestEntry 1) A-35
rdrFormatterDestPort (rdrFormatterDestEntry 2) A-35
rdrFormatterDestPriority (rdrFormatterDestEntry 3) A-36
rdrFormatterDestStatus (rdrFormatterDestEntry 4) A-36
rdrFormatterDestConnectionStatus (rdrFormatterDestEntry 5) A-36
rdrFormatterDestNumReportsSent (rdrFormatterDestEntry 6) A-36
rdrFormatterDestNumReportsDiscarded (rdrFormatterDestEntry 7) A-37
rdrFormatterDestReportRate (rdrFormatterDestEntry 8) A-37
rdrFormatterDestReportRatePeak (rdrFormatterDestEntry 9) A-37
rdrFormatterDestReportRatePeakTime (rdrFormatterDestEntry 10) A-37
rdrFormatterNumReportsSent (rdrFormatterGrp 3) A-37
rdrFormatterNumReportsDiscarded (rdrFormatterGrp 4) A-37
rdrFormatterClearCountersTime (rdrFormatterGrp 5) A-38
rdrFormatterReportRate (rdrFormatterGrp 6) A-38
rdrFormatterReportRatePeak (rdrFormatterGrp 7) A-38
rdrFormatterReportRatePeakTime (rdrFormatterGrp 8) A-38
rdrFormatterProtocol (rdrFormatterGrp 9) A-38
rdrFormatterForwardingMode (rdrFormatterGrp 10) A-39

rdrFormatterCategoryTable (rdrFormatterGrp 11) A-39
rdrFormatterCategoryEntry (rdrFormatterCategoryTable 1) A-40
rdrFormatterCategoryIndex (rdrFormatterCategoryEntry 1) A-40
rdrFormatterCategoryName (rdrFormatterCategoryEntry 2) A-40
rdrFormatterCategoryNumReportsSent (rdrFormatterCategoryEntry 3) A-40
rdrFormatterCategoryNumReportsDiscarded (rdrFormatterCategoryEntry 4) A-41
rdrFormatterCategoryReportRate (rdrFormatterCategoryEntry 5) A-41
rdrFormatterCategoryReportRatePeak (rdrFormatterCategoryEntry 6) A-41
rdrFormatterCategoryReportRatePeakTime (rdrFormatterCategoryEntry 7) A-41
rdrFormatterCategoryNumReportsQueued (rdrFormatterCategoryEntry 8) A-41
rdrFormatterCategoryDestTable (rdrFormatterGrp 12) A-42
rdrFormatterCategoryDestEntry (rdrFormatterCategoryDestTable 1) A-42
rdrFormatterCategoryDestPriority (rdrFormatterCategoryDestEntry 1) A-42
rdrFormatterCategoryDestStatus (rFormatterCategoryDestEntry 2) A-43
Logger Group: loggerGrp (pcubeSEObjs 7) A-44
loggerUserLogEnable (loggerGrp 1) A-44
loggerUserLogNumInfo (loggerGrp 2) A-44
loggerUserLogNumWarning (loggerGrp 3) A-44
loggerUserLogNumError (loggerGrp 4) A-44
loggerUserLogNumFatal (loggerGrp 5) A-45
loggerUserLogClearCountersTime (loggerGrp 6) A-45
Subscribers Group: subscribersGrp (pcubeSEObjs 8) A-46
subscribersInfoTable (subscribersGrp 2) A-46
subscribersInfoEntry (subscribersInfoTable 1) A-47
subscribersNumIntroduced (subscribersInfoEntry 1) A-47
subscribersNumFree (subscribersInfoEntry 2) A-48
subscribersNumIpAddrMappings (subscribersInfoEntry 3) A-48
subscribersNumIpAddrMappingsFree (subscribersInfoEntry 4) A-48
subscribersNumIpRangeMappings (subscribersInfoEntry 5) A-48
subscribersNumIpRangeMappingsFree (subscribersInfoEntry 6) A-48
subscribersNumVlanMappings (subscribersInfoEntry 7) A-49
subscribersNumVlanMappingsFree (subscribersInfoEntry 8) A-49
subscribersNumActive (subscribersInfoEntry 9) A-49
subscribersNumActivePeak (subscribersInfoEntry 10) A-49

subscribersNumActivePeakTime (subscribersInfoEntry 11) A-49
subscribersNumUpdates (subscribersInfoEntry 12) A-49
subscribersCountersClearTime (subscribersInfoEntry 13) A-50
subscribersNumTpIpRangeMappings (subscribersInfoEntry 14) A-50
subscribersNumTpIpRangeMappingsFree (subscribersInfoEntry 15) A-50
subscribersNumAnonymous (subscribersInfoEntry 16) A-50
subscribersNumWithSessions (subscribersInfoEntry 17) A-50
subscribersPropertiesTable (subscribersGrp 2) A-51
subscribersPropertiesEntry (subscribersPropertiesTable 1) A-51
spIndex (subscribersPropertiesEntry 1) A-51
spName (subscribersPropertiesEntry 2) A-51
spType (subscribersPropertiesEntry 3) A-52
subscriberPropertiesValuesTable (subscribersGrp 3) A-52
subscriberPropertiesValueEntry (subscriberPropertiesValueTable 1) A-53
spvIndex (subscriberPropertiesValueEntry 1) A-53
spvSubName (subscriberPropertiesValueEntry 2) A-53
spvPropertyName (subscriberPropertiesValueEntry 3) A-53
spvRowStatus (subscriberPropertiesValueEntry 4) A-54
spvPropertyStringValue (subscriberPropertiesValueEntry 5) A-54
spvPropertyUintValue (subscriberPropertiesValueEntry 6) A-54
spvPropertyCounter64Value (subscriberPropertiesValueEntry 7) A-54
Traffic Processor Group: trafficProcessorGrp (pcubeSEObjs 9) A-55
tpInfoTable (trafficProcessorGrp 1) A-55
tpInfoEntry (tpInfoTable) A-55
tpModuleIndex (tpInfoEntry 1) A-56
tpIndex (tpInfoEntry 2) A-57
tpTotalNumHandledPackets (tpInfoEntry 3) A-57
tpTotalNumHandledFlows (tpInfoEntry 4) A-57
tpNumActiveFlows (tpInfoEntry 5) A-57
tpNumActiveFlowsPeak (tpInfoEntry 6) A-57
tpNumActiveFlowsPeakTime (tpInfoEntry 7) A-58
tpNumTcpActiveFlows (tpInfoEntry 8) A-58
TpNumTcpActiveFlowsPeak (tpInfoEntry 9) A-58
tpNumTcpActiveFlowsPeakTime (tpInfoEntry 10) A-58

tpNumUdpActiveFlows (tpInfoEntry 11) A-58
tpNumUdpActiveFlowsPeak (tpInfoEntry 12) A-58
tpNumUdpActiveFlowsPeakTime (tpInfoEntry 13) A-59
tpNumNonTcpUdpActiveFlows (tpInfoEntry 14) A-59
tpNumNonTcpUdpActiveFlowsPeak (tpInfoEntry 15) A-59
tpNumNonTcpUdpActiveFlowsPeakTime (tpInfoEntry 16) A-59
tpTotalNumBlockedPackets (tpInfoEntry 17) A-59
tpTotalNumBlockedFlows (tpInfoEntry 18) A-60
tpTotalNumDiscardedPacketsDueToBwLimit (tpInfoEntry 19) A-60
tpTotalNumWredDiscardedPackets (tpInfoEntry 20) A-60
tpTotalNumFragments (tpInfoEntry 21) A-60
tpTotalNumNonIpPackets (tpInfoEntry 22) A-60
tpTotalNumIpCrcErrPackets (tpInfoEntry 23) A-61
tpTotalNumIpLengthErrPackets (tpInfoEntry 24) A-61
tpTotalNumIpBroadcastPackets (tpInfoEntry 25) A-61
tpTotalNumTtlErrPackets (tpInfoEntry 26) A-61
tpTotalNumTcpUdpCrcErrPackets (tpInfoEntry 27) A-61
tpClearCountersTime (tpInfoEntry 28) A-61
tpHandledPacketsRate (tpInfoEntry 29) A-62
tpHandledPacketsRatePeak (tpInfoEntry 30) A-62
tpHandledPacketsRatePeakTime (tpInfoEntry 31) A-62
tpHandledFlowsRate (tpInfoEntry 32) A-62
tpHandledFlowsRatePeak (tpInfoEntry 33) A-62
tpHandledFlowsRatePeakTime (tpInfoEntry 34) A-62
tpCpuUtilization (tpInfoEntry 35) A-63
tpCpuUtilizationPeak (tpInfoEntry 36) A-63
tpCpuUtilizationPeakTime (tpInfoEntry 37) A-63
tpFlowsCapacityUtilization (tpInfoEntry 38) A-63
tpFlowsCapacityUtilizationPeak (tpInfoEntry 39) A-63
tpFlowsCapacityUtilizationPeakTime (tpInfoEntry 40) A-64
tpServiceLoss (tpInfoEntry 41) A-64
Port Group: portGrp (pcubeSEObjs 10) A-65
portTable (portGrp 1) A-65
portEntry (portTable 1) A-65

portModuleIndex (portEntry 1) A-66
portIndex (portEntry 2) A-66
portType (portEntry 3) A-66
portNumTxQueues (portEntry 4) A-66
portIfIndex (portEntry 5) A-66
portAdminSpeed (portEntry 6) A-67
portAdminDuplex (portEntry 7) A-67
portOperDuplex (portEntry 8) A-67
portLinkIndex (portEntry 9) A-68
portOperStatus (portEntry 10) A-68
Transmit Queues Group: txQueuesGrp (pcubeSEObjs 11) A-69
txQueuesTable (txQueuesGrp 1) A-69
txQueuesEntry (txQueuesTable 1) A-69
txQueuesModuleIndex (txQueuesEntry 1) A-69
txQueuesPortIndex (txQueuesEntry 2) A-70
txQueuesQueueIndex (txQueuesEntry 3) A-70
txQueuesDescription (txQueuesEntry 4) A-70
txQueuesBandwidth (txQueuesEntry 5) A-70
txQueuesUtilization (txQueuesEntry 6) A-70
txQueuesUtilizationPeak (txQueuesEntry 7) A-71
txQueuesUtilizationPeakTime (txQueuesEntry 8) A-71
txQueuesClearCountersTime (txQueuesEntry 9) A-71
txQueuesDroppedBytes (txQueuesEntry 10) A-71
Global Controllers Group: globalControllersGrp (pcubeSEObjs 12) A-72
globalControllersTable (globalControllersGrp 1) A-72
globalControllersEntry (globalControllersTable 1) A-72
globalControllersModuleIndex (globalControllersEntry 1) A-73
globalControllersPortIndex (globalControllersEntry 2) A-73
globalControllersIndex (globalControllersEntry 3) A-73
globalControllersDescription (globalControllersEntry 4) A-73
globalControllersBandwidth (globalControllersEntry 5) A-73
globalControllersUtilization (globalControllersEntry 6) A-74
globalControllersUtilizationPeak (globalControllersEntry 7) A-74
globalControllersUtilizationPeakTime (globalControllersEntry 8) A-74

globalControllersClearCountersTime (globalControllersEntry 9) A-74
globalControllersDroppedBytes (globalControllersEntry 10) A-74
Application Group: applicationGrp (pcubeSEObjs 13) A-75
appInfoTable (applicationGrp 1) A-75
appInfoEntry (appInfoTable 1) A-75
appName (appInfoEntry 1) A-75
appDescription (appInfoEntry 2) A-75
appVersion (appInfoEntry 3) A-76
appPropertiesTable (applicationGrp 2) A-76
appPropertiesEntry (appPropertiesTable 1) A-76
apIndex (appPropertiesEntry 1) A-76
apName (appPropertiesEntry 2) A-77
apType (appPropertiesEntry 3) A-77
appPropertiesValuesTable (applicationGrp 3) A-77
appPropertiesValueEntry (appPropertiesValueTable 1) A-78
apvIndex (appPropertiesValueEntry 1) A-78
apvPropertyName (appPropertiesValueEntry 2) A-78
apvRowStatus (appPropertiesValueEntry 3) A-78
apvPropertyStringValue (appPropertiesValueEntry 4) A-79
apvPropertyUintValue (appPropertiesValueEntry 5) A-79
apvPropertyCounter64Value (appPropertiesValueEntry 6) A-79
Traffic Counters Group: trafficCountersGrp (pcubeSEObjs 14) A-80
trafficCountersTable (trafficCountersGrp 1) A-80
trafficCountersEntry (trafficCountersTable 1) A-80
trafficCounterIndex (trafficCountersEntry 1) A-80
trafficCounterValue (trafficCountersEntry 2) A-80
trafficCounterName (trafficCountersEntry 3) A-81
trafficCounterType (trafficCountersEntry 4) A-81
Attack Group: attackGrp (pcubeSEObjs 15) A-82
attackTypeTable (attackGrp 1) A-82
attackTypeEntry (attackTypeTable 1) A-82
attackTypeIndex (attackTypeEntry 1) A-82
attackTypeName (attackTypeEntry 2) A-83
attackTypeCurrentNumAttacks (attackTypeEntry 3) A-83

attackTypeTotalNumAttacks (attackTypeEntry 4) A-83
attackTypeTotalNumFlows (attackTypeEntry 5) A-83
attackTypeTotalNumSeconds (attackTypeEntry 6) A-83
attackTypeTableClearTime (attackTypeTable 2) A-84
Supported Standards A-84

Glossary of Terms 1

Index 1

Preface

This preface describes who should read the Cisco Service Control Engine (SCE) Software Configuration Guide, how it is organized, and its document conventions

Audience

This guide is for experienced network administrators who are responsible for configuring and maintaining the ***SCE*** platform.

Organization

The major sections of this guide are as follows:

Chapter	Title	Description
1	*Overview* (on page 1-1)	Overview of SCE platform management.
2	*Command-Line Interface* (on page 2-1)	Detailed explanation of how to use the Cisco SCE Command-line Interface.
3	*Configuring the Management Interface and Security* (on page 5-1)	Explanation of how to configure the various management options: Telnet, SSH, and SNMP. Also how to configure the system time,Domain Name Settings, management IP address, and passwords.
4	*Configuring the Line Interface* (on page 6-1)	Explanation of how to configure tunneling, TOS marking, and traffic rules.
5	*Configuring the Connection* (on page 7-1)	Explanation of how to configure the connection mode, link mode, and failure behaviors.
6	*Configuring the RDR Formatter* (on page 8-1)	Explanation of how to configure the RDR Formatter so that RDRs are sent to the proper destinations
7	*Managing Subscribers* (on page 9-1)	Explanation of how to import and export subscriber information and how to monitor subscribers.
8	*Redundancy and Fail-Over* (on page 10-1)	Explanation of how to configure and manage a redundant system. This chapter applies only to the SCE 2000 platform.

9	*Identifying And Preventing Distributed-Denial-Of-Service Attacks* (on page 11-1)	Explanation of how to configure attack filtering
10	*Operations* (on page 3-1)	Explanation of how to manage configurations, install applications and upgrade the system software.
11	*Utilities* (on page 4-1)	Explanation of the setup wizard and the user log, as well as of file operations.
Appendix A	*Proprietary MIB Reference* (on page A-1)	Definition of the proprietary Service Control Enterprise MIB.

Related Publications

Your ***SCE*** platform and the software running on it contain extensive features and functionality, which are documented in the following resources:

- For further information regarding the service Control CLI and a complete listing of all CLI commands, refer to the *Cisco Service Control Engine (SCE) CLI Command Reference*
- For complete installation information, including initial configuration, refer to the relevant installation guide:
  - *Cisco SCE 2000 4xGBE Installation and Configuration Guide*
  - *Cisco SCE 2000 4/8xFE Installation and Configuration Guide*
  - *Cisco SCE 1000 2xGBE Installation and Configuration Guide*

Note You can access Cisco software configuration and hardware installation and maintenance documentation on the World Wide Web at *Cisco Website URL* http://www.cisco.com. Translated documentation is available at the following URL: *International Cisco Website* (http://www.cisco.com/public/countries_languages.shtml)

- For initial installation and startup information, refer to the relevant quick start guide:
  - *Cisco SCE 2000 4xGBE Quick Start Guide*
  - *Cisco SCE 2000 4/8xFE Quick Start Guide*
  - *Cisco SCE 1000 2xGBE Quick Start Guide*
- For international agency compliance, safety, and statutory information for wide-area network (WAN) interfaces for the ***SCE*** platform, refer to the regulatory and safety information document:
  - *Regulatory Compliance and Safety Information for the Cisco Service Control Engine (SCE)*
- For installation and configuration of the other components of the Service Control Management Suite refer to:
  - *Service Control Management Suite Subscriber Manager User Guide*
  - *Service Control Management Suite Collection Manager User Guide*

- *Service Control Application Suite for Broadband User Guide*
- *Service Control Application Suite for Broadband Reference Guide*

- To view Cisco documentation or obtain general information about the documentation, refer to the following sources:
  - *Obtaining Documentation* (on page xx)
  - The Cisco Information Packet that shipped with your ***SCE*** platform.

Document Conventions

Command descriptions use the following conventions:

boldface font	Commands and keywords are in **boldface**.
italic font	Arguments for which you supply values are in i*talics*.
[ ]	Elements in square brackets are optional.
{x | y | z}	Alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string, or the string will include the quotation marks.

Screen examples use the following conventions:

`screen` font	Terminal sessions and information the system displays are in `screen` font.
`boldface screen` font	Information you must enter is in **`boldface screen`** font.
`italic screen` font	Arguments for which you supply values are in *`italic screen`* font.
^	The symbol ^ represents the key labeled **Control**—for example, the key combination **^D** in a screen display means hold down the **Control** key while you press the **D** key.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (**!**) or a pound sign (**#**) at the beginning of a line of code indicates a comment line.

Notes, cautionary statements, and safety warnings use these conventions.

Note Means *reader take note*. Notes contain helpful suggestions or references to materials not contained in this manual.

Warning Means *reader be careful*. You are capable of doing something that might result in equipment damage or loss of data.

Obtaining Documentation

Cisco documentation and additional literature are available on *Cisco.com* http://www.cisco.com. Cisco also provides several ways to obtain technical assistance and other technical resources. These sections explain how to obtain technical information from Cisco Systems.

Cisco.com

You can access the most current Cisco documentation at this *URL* http://www.cisco.com/univercd/home/home.htm.

You can access the Cisco website at this *URL* http://www.cisco.com.

You can access international Cisco websites at this *URL* (http://www.cisco.com/public/countries_languages.shtml).

Documentation Feedback

You can send comments about technical documentation to this *URL* (http://www.bug-doc@cisco.com).

You can submit comments by using the response card (if present) behind the front cover of your document or by writing to the following address:

Cisco Systems

Attn: Customer Document Ordering

170 West Tasman Drive

San Jose, CA 95134-9883

We appreciate your comments.

Obtaining Technical Assistance

For all customers, partners, resellers, and distributors who hold valid Cisco service contracts, the Cisco Technical Assistance Center (TAC) provides 24-hour-a-day, award-winning technical support services, online and over the phone. *Cisco.com* http://www.cisco.com features the Cisco TAC website as an online starting point for technical assistance. If you do not hold a valid Cisco service contract, please contact your reseller.

Cisco Technical Support Website

The *Cisco TAC website* (http://www.cisco.com/tac) provides online documents and tools for troubleshooting and resolving technical issues with Cisco products and technologies. The Cisco TAC website is available 24 hours a day, 365 days a year.

Accessing all the tools on the Cisco TAC website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a login ID or password, register at this *URL* (http://tools.cisco.com/RPF/register/register.do).

Submitting a Service Request

Using the online *TAC Service Request Tool* (http://www.cisco.com/techsupport/servicerequest) is the fastest way to open S3 and S4 service requests. (S3 and S4 service requests are those in which your network is minimally impaired or for which you require product information.) After you describe your situation, the TAC Service Request Tool automatically provides recommended solutions. If your issue is not resolved using the recommended resources, your service request will be assigned to a Cisco TAC engineer.

For S1 or S2 service requests or if you do not have Internet access, contact the Cisco TAC by telephone. (S1 or S2 service requests are those in which your production network is down or severely degraded.) Cisco TAC engineers are assigned immediately to S1 and S2 service requests to help keep your business operations running smoothly.

To open a service request by telephone, use one of the following numbers:

Asia-Pacific: +61 2 8446 7411 (Australia: 1 800 805 227)

EMEA: +32 2 704 55 55

USA: 1 800 553-2447

A complete listing of *Cisco TAC contacts* (http://www.cisco.com/techsupport/contacts) is available online.

Definitions of Service Request Severity

To ensure that all cases are reported in a standard format, Cisco has established severity definitions.

Severity 1 (S1)—Your network is “down” or there is a critical impact to your business operations. You and Cisco will commit all necessary resources around the clock to resolve the situation.

Severity 2 (S2)—Operation of an existing network is severely degraded, or significant aspects of your business operation are negatively affected by inadequate performance of Cisco products. You and Cisco will commit full-time resources during normal business hours to resolve the situation.

Severity 3 (S3)—Operational performance of your network is impaired, but most business operations remain functional. You and Cisco will commit resources during normal business hours to restore service to satisfactory levels.

Severity 4 (S4)—You require information or assistance with Cisco product capabilities, installation, or configuration. There is little or no effect on your business operations.

Obtaining Additional Publications and Information

Information about Cisco products, technologies, and network solutions is available from various online and printed sources.

- *Cisco Marketplace* (http://www.cisco.com/go/marketplace/) provides a variety of Cisco books, reference guides, and logo merchandise.
- The *Cisco Product Catalog* describes the networking products offered by Cisco Systems, as well as ordering and customer support services.
- *Cisco Press* (http://www.ciscopress.com) publishes a wide range of general networking, training and certification titles. Both new and experienced users will benefit from these publications. For current Cisco Press titles and other information, go to *Cisco Press* (http://www.ciscopress.com).
- *Packet* (http://www.cisco.com/packet) magazine is the Cisco Systems technical user magazine for maximizing Internet and networking investments. Each quarter, Packet delivers coverage of the latest industry trends, technology breakthroughs, and Cisco products and solutions, as well as network deployment and troubleshooting tips, configuration examples, customer case studies, certification and training information, and links to scores of in-depth online resources.
- *iQ Magazine* (http://www.cisco.com/go/iqmagazine) is the quarterly publication from Cisco Systems designed to help growing companies learn how they can use technology to increase revenue, streamline their business, and expand services. The publication identifies the challenges facing these companies and the technologies to help solve them, using real-world case studies and business strategies to help readers make sound technology investment decisions.
- *Internet Protocol Journal* (http://www.cisco.com/ipj) is a quarterly journal published by Cisco Systems for engineering professionals involved in designing, developing, and operating public and private internets and intranets.
- World-class networking training is available from Cisco. You can view current offerings at this *URL* (http://www.cisco.com/en/US/learning/index.html).

CHAPTER 1

Introduction

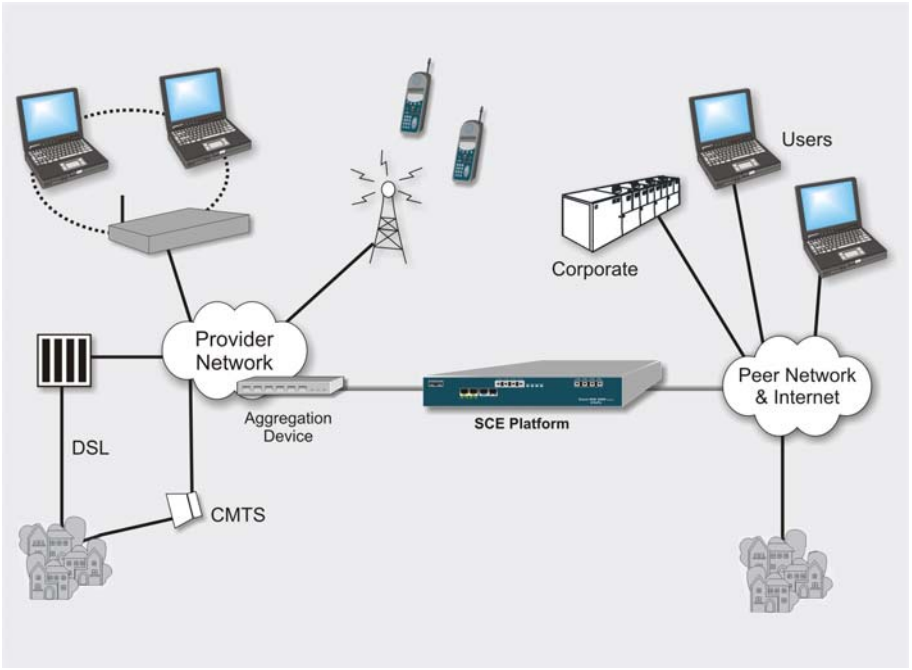
The Service Control Engine family of programmable network devices is capable of performing stateful flow inspection of IP traffic, and controlling that traffic based on configurable rules. The Service Control Engine platforms provide a real-time classification of network usage through programmable, stateful inspection of bi-directional traffic flows and the mapping of these flows with user ownership.

This chapter contains the following sections:

- SCE Platform Management Interfaces 1-2

The following diagram demonstrates a deployment of an ***SCE*** platform in the network.

Figure 1-1: SCE Platform in the Network



SCE Platform Management Interfaces

You can manage the ***SCE*** platform through either of its management interfaces, CLI or SNMP. Both these interfaces provide management access to the same database of the ***SCE*** platform; any configuration changes made through one interface are also reflected through the other interface.

- **CLI** (Command Line Interface). The CLI is accessible through the Console port or through a Telnet connection. The CLI is the interface described throughout this manual. *Command Line Interface* (see "Command-Line Interface" on page 2-1) further discusses the CLI.
- **SNMP** (Simple Network Management Protocol). You can use SNMP as an interface for monitoring the variables as defined in the MIB-II and Cisco's propriety MIB specifications. For information on enabling SNMP, see *SNMP Interface* (on page 5-8)

Command-Line Interface

This chapter describes how to use the SCE platform Command-Line Interface (CLI), its hierarchical structure, authorization levels and its help features. The Command-Line Interface is one of the ***SCE*** Platform management interfaces.

This chapter contains the following sections:

- Getting Help 2-2
- Authorization and Command Levels (Hierarchy) 2-2
- CLI Help Features 2-13
- Navigational and Shortcut Features 2-15
- Managing Command Output 2-17
- CLI Scripts2-18

The CLI is accessed through a Telnet session or directly via the console port on the front panel of the SCE platform. When you enter a Telnet session, you enter as the simplest level of user, in the User Exec mode.

The SCE platform supports up to six concurrent CLI sessions; five sessions initiated by Telnet connection, and one session on the console port.

In this chapter, the procedures shown are examples of how to perform typical SCE Platform management functions using the CLI. The *CLI Command Reference* chapter gives you examples of how to implement the most common of these commands, and general information on the interrelationships between the commands and the conceptual background of how to use them.

Getting Help

To obtain a list of commands that are available for each command mode, enter a question mark (?) at the system prompt. You also can obtain a list of any command's associated keywords and arguments with the context-sensitive help feature.

The following table lists commands you can enter to get help that is specific to a command mode, a command, a keyword, or an argument.

Table 2-1 Getting Help

Command	Purpose
abbreviated-command-entry?	Obtain a list of commands that begin with a particular character string. (Do not leave a space between the command and question mark.)
abbreviated-command-entry<Tab>	Complete a partial command name.
?	List all commands available for a particular command mode.
command ?	List a command's associated keywords. Leave a space between the command and question mark.
command keyword ?	List a keyword's associated arguments. Leave a space between the keyword and question mark.

Authorization and Command Levels (Hierarchy)

When using the CLI there are two important concepts that you must understand in order to navigate:

- **Authorization Level:** Indicates the level of commands you can execute. A user with a simple authorization level can only view some information in the system, while a higher level administrator can actually make changes to configuration. Almost all of the procedures in this manual require an Admin authorization level. See CLI Command Hierarchy.
- **Command Hierarchy Level:** Provides you with a context for initiating commands. Commands are broken down into categories and you can only execute each command within the context of its category. For example, in order to configure parameters related to the Line Card, you need to be within the LineCard Interface Configuration Mode. See CLI Command Hierarchy.

The following sections describe the available Authorization and Command Hierarchy Levels and how to maneuver within them.

The on-screen prompt indicates both your authorization level and your command hierarchy level, as well as the assigned host name. See *Prompt Indications* (on page 2-12).

Note Throughout the manual, ***SCE*** is used as the sample host name.

CLI Command Hierarchy

The set of all CLI commands is grouped in hierarchical order, according to the type of the commands. The first two levels in the hierarchy are the User Exec and the Privileged Exec modes. These are non-configuration modes in which the set of available commands enables the monitoring of the SCE platform, file system operations, and other operations that cannot alter the configuration of the SCE platform.

The next levels in the hierarchy are the Global and Interface configuration modes, which hold a set of commands that control the global configuration of the SCE platform and its interfaces. Any of the parameters set by the commands in these modes should be saved in the startup configuration, such that in the case of a reboot, the SCE platform restores the saved configuration.

The following table shows the available CLI modes.

Table 2-2 CLI Modes

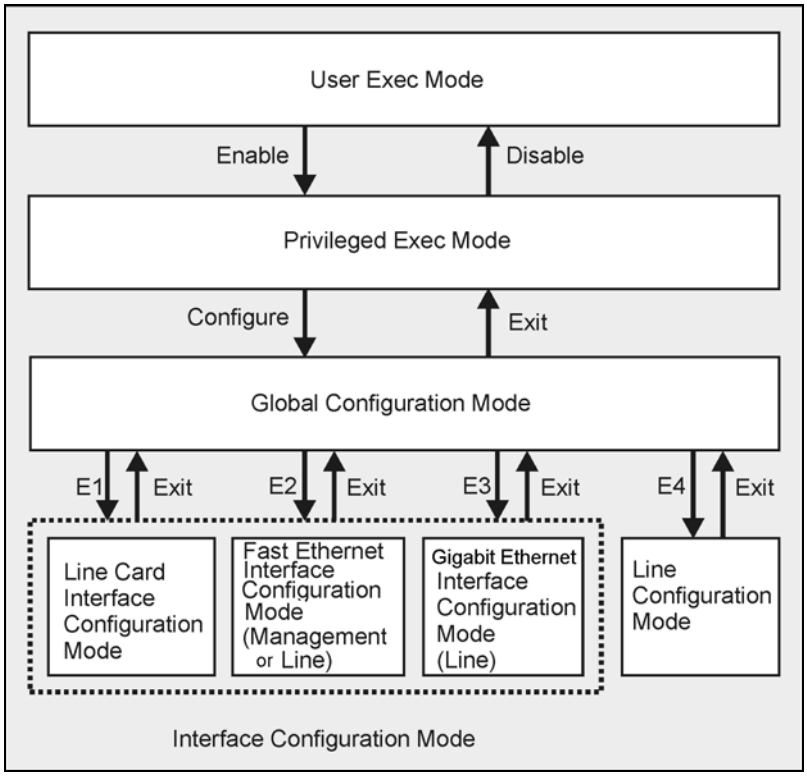
Mode	Description	Level	Prompt indication
User Exec	Initial mode with very limited functionality.	User	***SCE***>
Privileged Exec	General administration; file system manipulations and control of basic parameters that do not change the configuration of the SCE platform.	Admin	***SCE***#
Global Configuration	Configuration of general system parameters, such as DNS, host name, and time zone.	Admin	***SCE***`(config)#`
Interface Configuration	Configuration of specific system interface parameters, such as the Line Card and the Ethernet interfaces.	Admin	***SCE***`(config if)#`
Line Configuration	Configuration of Telnet lines, such as an access-list.	Admin	***SCE***`(config-line)#`

When you login to the system, you have the User authorization level and enter User Exec mode. Changing the authorization level to Admin automatically moves you to Privileged Exec mode. In order to move to any of the configuration modes, you must enter commands specific to that mode.

The list of available commands in each mode can be viewed using the question mark '?' at the end of the prompt.

The figure below, illustrates the hierarchical structure of the CLI modes, and the CLI commands used to enter and exit a mode.

Figure 2-1: CLI Command Hierarchy



The following commands are used to enter the different configure interface modes and the Line Configuration Mode:

- E1 **`interface LineCard 0`**
- E2 **`interface FastEthernet 0/0`** (management port, all platforms)
- E2 **`interface FastEthernet`** *`0/1, 0/2, 0/3, or 0/4`* (line ports, SCE 2000 4/8xFE platform)
- E3 **`interface GigabitEthernet`** *`0/1, 0/2, 0/3, or 0/4`* (line ports, SCE 2000 4xGBE platform)
- E3 **`interface GigabitEthernet`** *`0/1, 0/,`* (line ports, SCE 1000 2xGBE platform)
- E4 **`line vty 0`**

To move from one interface configuration mode to another you must exit the current interface configuration mode (as illustrated in the above figure).

Note Although the system supports up to five concurrent Telnet connections, you cannot configure them separately. This means that any number you enter in the **`line vty`** command (**0**, **1**, **2**, **3** or **4**) will act as a **0** and configure all five connections together.

EXAMPLE:

This example illustrates moving into and out from Interface configuration mode as follows:

- Configure the SCE platform time zone (global configuration)
- Enter **`FastEthernet`** Interface configuration mode for Mng port
- Configure the speed of the management interface
- Define the link mode.
- Exit Interface configuration mode

```
SCE#>configure
SCE(config)#>clock timezone PST -10
SCE(config)#>interface FastEthernet 0/0
SCE(config if)#>speed 100
SCE(config)#>exit
SCE(config)#>interface LineCard 0
SCE(config if)#>link-mode forwarding
SCE(config if)#>exit
```

Entering and Exiting Global Configuration Mode

To enter the Global Configuration Mode:

Step 1 At the ***SCE***# prompt, type **`configure`**, and press **Enter**.

The ***SCE***(config)# prompt appears.

To exit the Global Configuration Mode:

Step 1 At the ***SCE***(config)# prompt, type exit and press **Enter**.

The ***SCE***# prompt appears.

Interface Configuration Modes

The components that are configured by the Interface Configuration Modes are:

- Card
  - LineCard: **`Interface LineCard 0`**

    The LineCard interface configures the main functionality of viewing and handling traffic on the line.
- Ports
  - See *Configuring the Physical Ports* (on page 2-6)
- Telnet
  - Line Configuration Mode: **`Line vty 0`**

    The Line Configuration Mode enables you to configure Telnet parameters.

Configuring the Physical Ports

The SCE platform system contains the following physical port interfaces:

- Fast Ethernet Management:

  `Interface FastEthernet 0/0`

  The FastEthernet Management Interface configures the settings for the interface to other network elements within the system. This interface should be connected to the internal Ethernet within the operator's site.
- Fast Ethernet (SCE 2000 4/8xFE):

  `Interface FastEthernet 0/1, 0/2, 0/3, or 0/4`

  The FastEthernet Interface mode configures the settings for the FastEthernet interface to the Internet traffic on the wire. Each of the four ports can be set individually.
- Gigabit Ethernet (SCE 1000 platform):

  `Interface GigabitEthernet 0/1,` `or` **`0/2`**

  The GigabitEthernet Interface mode configures the settings for the GigabitEthernet interface to the Internet traffic on the wire. Each of the two ports can be set individually.
- Gigabit Ethernet (SCE 2000 4xGBE platform):

  `Interface GigabitEthernet 0/1, 0/2, 0/3, or 0/4`

  The GigabitEthernet Interface mode configures the settings for the GigabitEthernet interface to the Internet traffic on the wire. Each of the four ports can be set individually.

Note You need to specify the slot number and the interface number when referencing any interface. The slot number is always 0, and the interfaces are numbered as follows:
Ethernet Line Interfaces:
SCE 1000 platform: **1,2**
SCE 2000 platform: **1,2,3,4**
FastEthernet Management Interface: **0**

Configuring the Management Port

The following commands are used to configure the management port for all platforms:

- duplex
- ip address
- speed

Configuring the Fast Ethernet Line Ports

The commands that are used to configure the Fast Ethernet line ports are:

- bandwidth
- duplex
- queue
- speed

Configuring the Gigabit Ethernet Line Ports

The commands that are used to configure the Gigabit Ethernet line ports are:

- auto-negotiate (GigabitEthernet only)
- bandwidth
- queue

Entering FastEthernet (Management) Interface Configuration Mode

Before you can configure the FastEthernet parameters for the management interface, you must be in the FastEthernet Management Interface Configuration Mode.

To enter FastEthernet Management Interface Configuration Mode:

Step 1 To enter Global Configuration Mode, type **`configure`** and press **Enter**.

The ***SCE***`(config)#` prompt appears.

Step 2 Type **`interface FastEthernet 0/0`** and press **Enter**.

The ***SCE***`(config if)#` prompt appears.

The system prompt changes to reflect the higher level mode.

To return to the Global Configuration mode:

Step 1 Type **`exit`**.

Entering LineCard Interface Configuration Mode

The following procedure is for entering Line Card Interface Configuration mode. The procedures for entering the other interfaces are the same except for the interface command as described above and in CLI Command Reference.

To enter LineCard Interface Configuration mode:

Step 1 To enter Global Configuration Mode, at the ***SCE***# prompt, type **`configure`**, and press **Enter**.

The ***SCE***`(config)#` prompt appears.

Step 2 Type **`interface LineCard`** 0, and press **Enter**.

The ***SCE***`(config if)#` prompt appears.

Step 3 To return to Global Configuration Mode, type **`exit`** and press **Enter**.

The ***SCE***`(config)#` prompt appears.

Step 4 To exit Global Configuration Mode, type **`exit`** and press **Enter**.

The ***SCE***# prompt appears.

Entering Ethernet Line Interface Configuration Mode

Entering the Fast Ethernet Line Interface Configuration Mode

To enter the FastEthernet Interface Configuration Mode:

Step 1 To enter Global Configuration Mode, type **`configure`** and press **Enter**.

The ***SCE***`(config)#` prompt appears.

Step 2 For the SCE 2000, type **`interface FastEthernet [0/1|0/2|0/3|0/4]`** and press **Enter**.

The ***SCE***`(config if)#` prompt appears.

EXAMPLE:

The following example shows how to enter Configuration Mode for the FastEthernet Interface number 3.

```
SCE(config)#interface FastEthernet 0/3
SCE(config if)#
```

Entering the Gigabit Ethernet Line Interface Configuration Mode

To enter the GigabitEthernet Interface Configuration Mode:

Step 1 To enter Global Configuration Mode, type **configure** and press **Enter**.

The *SCE*(config)# prompt appears.

Step 2 For the SCE 1000, type **interface GigabitEthernet [0/1|0/2]** and press **Enter**.

Step 3 For the SCE 2000, type **interface GigabitEthernet [0/1|0/2|0/3|0/4]** and press **Enter**.

The *SCE*(config if)# prompt appears.

EXAMPLE:

The following example shows how to enter Configuration Mode for the GigabitEthernet Interface number 2.

```
SCE(config)#interface GigabitEthernet 0/2
SCE(config if)#
```

Navigating between the Interface Configuration Modes

To navigate from one Interface Configuration Mode to another:

Step 1 Type **exit**.

You are returned to the Global Configuration Mode.

Step 2 Type the appropriate command to enter a different Interface Configuration Mode.

Exiting Modes

This section describes how to revert to a previous mode. When you use the exit command you revert to the general level above the current level, as shown in the figure in *CLI Command Hierarchy* (on page 2-3).

To exit from the Privileged Exec mode and revert to the User Exec mode:

Step 1 At the ***SCE***# prompt, type **`disable`**, and press **Enter**.

The ***SCE***> prompt for the User Exec mode appears.

Exiting from any configuration mode and revert to the previous mode is done in the same manner, as in the following procedure.

To exit from the Global Configuration Mode:

Step 1 At the ***SCE***`(config)#` prompt, type **`exit`**, and press **Enter**.

The appropriate prompt for the previous level appears.

EXAMPLE:

The following example shows the system response when you exit the Interface Configuration mode.

```
SCE(config if)#exit
SCE(config)#
```

CLI Authorization Levels

The SCE platform system has three authorization levels, which represent the user's access permissions. When you initially connect to the SCE platform, you automatically have the most basic authorization level, that is User, which allows minimum functionality.

In order to perform administrative functions on the SCE platform, you must have Admin or Root authorization, which means changing the level by logging in with an Admin or Root password, as described in the procedure "To log in with Admin level authorization," below. This manual covers the functions that can be performed by the Admin level user.

The commands available in each authorization level are all the commands of the lower authorization layers plus commands that are authorized only to this level.

Note This manual covers the functions that can be performed by the Admin level user, unless otherwise noted.

The following CLI commands are related to authorization levels:

- `enable`
- `disable`

Each authorization level has a value (number) corresponding to it. When using the CLI commands, use the values, not the name of the level, as shown in the following table.

Table 2-3 Authorization Levels

Level	Description	Value	Prompt
User	Password required. This level enables basic operational functionality.	0	>
Admin	Password required. For use by general administrators, the Admin authorization level enables configuration and management of the SCE platform.	10	#
Root	Password required. For use by technical field engineers, the Root authorization level enables configuration of all advanced settings, such as debug and disaster recovery. The Root level is used by technical engineers only and is not documented in this manual.	15	#>

A telnet session begins with a request for password, and will not continue until the proper user password is supplied. This enhances the security of the system by not revealing its identity to unauthorized people.

To log in with Admin level authorization:

Step 1 Initiate a telnet connection.

Step 2 A `Password:` prompt appears. Type in the user level password and press **Enter**.

The ***SCE***> prompt appears.

You now have user level authorization.

Step 3 From the ***SCE***> prompt, type **`enable 10`** and press **Enter**.

The system prompts for a password by showing the prompt `Password`:

Step 4 Type in the password for the Admin level and press **Enter**.

Note that the password is an access-level authorization setting, not an individual user password.

The system prompt changes to ***SCE***# to show you are now in Admin level.

EXAMPLE:

The following example illustrates how to change the authorization level from User to Admin, and then revert back to User. No password is required for moving to a lower authorization level.

```
SCE>enable 10
Password:  cisco
SCE#disable
```

Prompt Indications

The on-screen prompt indicates your authorization level, your command hierarchy level, and the assigned host name. The structure of the prompt is:
<`hostname(mode-indication)level-indication`>

Authorization levels are indicated as follows:

This prompt...	**Indicates this...**
>	indicates User and Viewer levels
#	indicates Admin level
#>	indicates Root level

Command hierarchy levels are indicated as follows:

This command hierarchy...	**Is indicated as...**
User Exec	***SCE***>
Privileged Exec	***SCE***#
Global Configuration	***SCE***`(config)`#
Interface Configuration	***SCE***`(config if)`#
Line Configuration	***SCE***`(config-line)`#

EXAMPLE:

The prompt `My`***SCE***`(config if)`# indicates:

- The name of the ***SCE*** platform is `My`***SCE***
- The current CLI mode is Interface configuration mode
- The user has Admin authorization level

Syntax and Conventions

The CLI commands are written in the following format:

command *required-parameter* [*optional-parameter*]

[no] is an optional parameter that may appear before the command name.

- When typing commands, you may enclose parameters in double-quote marks, and you *must* do so when there is a space within a parameter name.
- Examples are shown in courier style. **Bold courier** is used to show the commands as you type them and regular courier is used for system prompts and responses.

Note The command prompt, SCE*xxxx*, in the examples and in other sections of the CLI commands represents the type of platform of the SCE, where *xxxx* denotes either 1000 for the SCE1000 platform or 2000 for the SCE2000 platform.

Login and User Levels

To log in to the SCE platform, start a Telnet session from your computer to connect to the Command-Line Interface (CLI). When you initially connect to the SCE platform, you are automatically in the User authorization level, which is the most basic mode with minimum functionality.

In order to perform administrative functions on the SCE platform, you must enter the password-protected Admin or Root authorization levels. The password is not a personal password, but rather it is a password that gives you and others access to these levels.

During the course of a Telnet session, you can change your current access level by enabling or disabling the access level and giving the correct system password. There are four authorization levels, as described in the following table.

Table 2-4 Authorization Levels

Level	Value	Description
User	0	By default, password required. This level provides minimum functionality.
Admin	10	By default, password required. For use by general administrators, the Admin authorization level enables configuration of the SCE platform.
Root	15	By default, password required. For use by technical field engineers, the Root authorization level enables configuration of all advanced settings, such as debug and disaster recovery.

When setting the authorization level in the CLI commands, you must use the value number rather than the level name.

CLI Help Features

CLI provides context sensitive help. Two types of context sensitive help are supported:

- Partial help
- Argument help

Partial Help

To obtain a list of commands that begin with a particular character string, enter the abbreviated command entry immediately followed by a question mark (?). This form of help is called partial help, because it lists only the keywords or arguments that begin with the abbreviation you entered.

EXAMPLE:

The following example illustrates how typing **c?** displays all available arguments that start with the letter c.

```
SCE(config)#snmp-server c?
Community         contact
SCE(config)#snmp-server c
```

Argument Help

To obtain a list of command's associated keywords or parameters, type a question mark (?) in place of a keyword or parameter on the command line.

Note that if <**Enter**> is acceptable input, the symbol <cr> represents the **Enter** key.

EXAMPLE:

The following example illustrates how to get a list of all arguments or keywords expected after the command **snmp-server**.

```
SCE(config)#snmp-server ?
Community   Define community string
Contact           Set system contact
Enable            Enable the SNMP agent
Host              Set traps destination
Location          Set system location
SCE(config)#
```

When asking for help on particular parameter, the system informs you of the type of data that is an accepted legal value. The types of parameters supported are:

STRING — When a String is expected, you can enter any set of characters or digits. If the string has a space as one of its characters, use double-quote (") marks to enclose the string.

DECIMAL — Any decimal number. Positive number is assumed, for negative numbers use the "–" symbol.

HEX — A hexadecimal number; must start with either 0x or 0X.

EXAMPLE:

The following example illustrates the use of **?** to get help on commands syntax. In this example, you can enter either the word **running-config**, or any name of a file, after the word **copy**.

```
SCE#copy ?
     running-config            Copy running configuration file
     STRING             Source file name
SCE#
```

The [no] Prefix

Many CLI commands offer the option of adding the word **no** before the command to disable the feature controlled by the command or revert it to its default configuration. This notation is shown in the CLI Command Reference as **[no]** to denote it is optional.

For example, **no service telnetd** disables the telnet server. Enabling the telnet server is done by typing **service telnetd**.

Navigational and Shortcut Features

Command History

CLI maintains a history buffer of the most recent commands you used in the current CLI session for quick retrieval. Using the keyboard, you can navigate through your last commands, one by one, or all commands that start with a given prefix. By default, the system saves the last 30 commands you typed. You can change the number of commands remembered using the **`history size`** command.

To use the history functions, use the keys shown in the following table.

Table 2-5 Keyboard Shortcuts for History Functions

Arrow	Shortcut	Description
Up arrow	Ctrl-P	Moves cursor to the previous command with the same prefix.
Down arrow	Ctrl-N	Moves cursor to the next command with the same prefix as original.
	Ctrl-L Ctrl-R	Re-display the current command line.

Keyboard Shortcuts

The SCE platform has a number of keyboard shortcuts that make it easier to navigate and use the system. The following table shows the keyboard shortcuts available.

You can get a display the keyboard shortcuts at any time by typing **`help bindings`**.

Table 2-6 Keyboard Shortcuts

Description	Shortcut Key
Navigational shortcuts	
Move cursor one character to the right.	CTRL-F /->
Move cursor one character to the left.	CTRL-B /<-
Move cursor one word to the right (forward).	ESC-F
Move cursor one word to the left (backward.	ESC-B
Move cursor to the start of the line.	CTRL-A
Move cursor to the end of the line.	CTRL-E
Editing shortcuts	
Delete the character where the cursor is located.	CTRL-D
Delete from the cursor position to the end of the word.	ESC-d
Delete the character before the current location of the cursor.	Backspace
Delete the character before the current location of the cursor.	CTRL-H
Deletes from the cursor position to the end of the line	CTRL-K

Description	Shortcut Key
Deletes all characters from the cursor to the beginning of the line	CTRL-U
Deletes all characters from the cursor to the beginning of the line. (Same functionality as CTRL-U.)	CTRL-X
Delete the word to the left of the cursor.	CTRL-W
Recall the last item deleted.	CTRL-Y
Completes the word when there is only one possible completion.	<Tab>
Completes the word when there is only one possible completion. (Same functionality as <Tab>.)	CTRL-I

Tab Completion

The CLI interface features tab completion. When you type in the first letters of a command and type **<Tab>**, the system automatically fills in the rest of the command or keyword. This feature works only when there is one possible command that could be possible using the starting letters.

EXAMPLE:

The letters **snm** followed by **<Tab>** will be completed to the command **snmp-server.**

```
SCE(config)#snm<Tab>
SCE(config)#snmp-server
```

If you type **<Enter>** instead of **<Tab>**, and there is no ambiguity, the system actually carries out the command which would be filled in by the rest of the word.

EXAMPLE:

The following example displays how the system completes a partial (unique) command for the **enable** command. Because **enable** does not require any parameters, the system simply carries out the **enable** command when the user presses **Enter**.

```
SCE>en<Enter>
Password:
SCE#
```

FTP User Name and Password

CLI enables saving ftp user name and password to be used in FTP operations—download and upload, per session.

These settings are effective during the current CLI session.

EXAMPLE:

The following example illustrates how to set FTP password and user name and the use in these settings for getting a file named *config.tmp* from a remote station using FTP protocol.

```
SCE#ip ftp password vk
SCE#ip ftp username vk
SCE#copy ftp://@10.1.1.253/h:/config.tmp myconf.txt
connecting 10.1.1.253 (user name vk password vk) to retrieve config.tmp
SCE#
```

Managing Command Output

Some commands, such as many **show** commands, may have many lines of output. There are several ways of managing the command output:

- Scrolling options: When the command output is too large to be displayed all at once, you can control whether the display scrolls line by line or refreshes the entire screen.
- Filtering options: You can filter the output so that output lines are displayed only if they include or exclude a specified expression.
- Redirecting to a file: You can send the output to a specified file

Scrolling the Screen Display

The output of some **show** and **dir** commands is quite lengthy and cannot all be displayed on the screen at one time. Commands with many lines of output are displayed in chunks of 24 lines. You can choose to scroll the display line by line or refresh the entire screen. At the prompt after any line, you can type one of the following keys for the desired action:

- **<Enter>**– show one more line
- **<Space>** – show 24 more lines (a new chunk)
- **<g>** – Stop prompting for more
- **<?>** – Display a help string showing possible options
- Any other key – quit showing the file

Filtering Command Output

You can filter the output of certain commands, such as **show**, **more**, and **dir,** so that output lines are displayed only if they include or exclude a specified expression. The filtering options are as follows:

- **include**: Shows all lines that include the specified text.
- **exclude**: Does not show any lines that include the specified text.
- **begin**: Finds the first line that includes the specified text, and shows all lines beginning with that line. All previous lines are excluded.

The syntax of filtered commands is as follows:

- **<command> | include <expression>**
- **<command> | exclude <expression>**
- **<command> | begin <expression>**

The <expression> in these commands is case sensitive.

EXAMPLE

Following is an example of how to filter the **show version** command to display only the last part of the output, beginning with the version information.

```
SCE# show version begin revision
```

Redirecting Command Output to a File

You can redirect the output of commands, such as **show**, **more**, and **dir**, to a file. When writing the output of these commands to a file, you can specify either of the following options:

- **redirect**: The new output of the command will overwrite the existing contents of the file.
- **append**: The new output of the command will be appended to the existing contents of the file.

The syntax of redirection commands is as follows:

- **<command> | redirect <file-name>**
- **<command> | append <file-name>**

EXAMPLE

Following is an example of how to do the following:

- Filter the **more** command to display from a *csv* subscriber file only the gold package subscribers.
- Redirect that output to a file named *current_gold_subscribers*. The output should not overwrite existing entries in the file, but should be appended to the end of the file.

```
SCE# more subscribers_10.10.2004 include gold append
current_gold_subscribers
```

CLI Scripts

The CLI scripts feature allows you to record several CLI commands together as a script and play it back. This is useful for saving repeatable sequence of commands , such as software upgrade. For example, if you are configuring a group of SCE platforms and you want to run the same configuration commands on each platform, you could create a script on one platform and run it on all the other SCE platforms.

The available script commands are:

- `script capture`
- `script stop`
- `script print`
- `script run`

To create a script:

Step 1 At the *SCE#* prompt, type **script capture** *sample1.scr* where *sample1.scr* is the name of the script.

Step 2 Perform the actions you want to be included in the script.

Step 3 Type **script stop**.

The system saves the script.

EXAMPLE:

The following is an example of recording a script for upgrading software.

```
SCE#script capture upgrade.scr
SCE#configure
SCE(config)#boot system new.pkg
Verifying package file...
Package file verified OK.
SCE(config)#exit
SCE#copy running-config startup-config
Writing general configuration file to temporary location...
Extracting files from '/tffs0/images/new.pkg'...
Verifying package file...
Package file verified OK.
Device '/tffs0/' has 81154048 bytes free, 21447973 bytes are
needed for extraction, all is well.
Extracting files to temp locations...
Renaming temp files...
Extracted OK.
Backing-up general configuration file...
Copy temporary file to final location...
SCE#script stop
SCE#
```

To run the script recorded above, type:

```
SCE#script run upgrade.scr
```

Operations

This chapter contains the following sections:

- Managing Configurations 3-1
- Upgrading SCE Platform Firmware 3-7
- Configuring Applications 3-8
- Rebooting and Shutting Down the SCE Platform 3-12

Managing Configurations

Viewing Configuration

When you enter configuration commands, it immediately effects the ***SCE*** platform operation and configuration. This configuration, referred to as the *`running-config`*, is saved in the ***SCE*** platform volatile memory and is effective while the ***SCE*** platform is up. After reboot, the ***SCE*** platform loads the *`startup-config`*, which includes the non-default configuration as saved by the user, into the *`running-config`*.

The ***SCE*** platform provides commands for:

- Viewing the running configuration
- Viewing the startup configuration

After configuring the ***SCE*** platform, you may query for the running configuration using the command **`show running-config`**. This command displays the non-default running configuration. To view the complete ***SCE*** platform running configuration, including both default and user-configured configuration, you may use the option **`all-data`** in the **`show running-config`** command.

To view the running configuration, complete the following steps:

Step 1 At the ***SCE#*** prompt, type **`show running-config`**.

The system shows the running configuration.

```
SCE#show running-config
#This is a general configuration file (running-config).
#Created on 15:50:56  CET  MON  February  11  2002
#cli-type 1
#version 1

clock timezone CET 1
snmp-server community “public” ro
snmp-server host 10.1.1.253 traps version 1 “public”
interface LineCard 0
connection-mode active
no silent
no shutdown
flow-aging default-timeout UDP 60
interface FastEthernet 0/0
ip address 10.1.5.109 255.255.0.0
interface FastEthernet 0/1
interface FastEthernet 0/2
exit
line vty 0 4
no timeout
exit
SCE#
```

One of the useful show commands is the **`show version`** command. This command displays global static information on the ***SCE*** platform as software and hardware version, image build time, system uptime, last open packages names and information on the SLI application assigned.

To show the version information for the ***SCE*** platform software and hardware, complete the following steps:

Step 1 At the ***SCE#*** prompt, type **`show version`**.

The system shows the version information.

```
SCE#show version
System version: Version 2.5.2 Build 240
Build time: Jan 11 2005, 07:34:47
Software version is: Version 2.5.2 Build 240
Hardware information is:
rx            : 0x0075
dp            : 0x1808
tx            : 0x1708
ff            : 0x0077
cls           : 0x1721
cpld          : 0x0025
Lic           : 0x0176
rev           : G001
Bootrom       : 2.1.0
L2 cache      : Samsung 0.5
lic type      : MFE
optic mode    :
Part number: 53AA-BXC1-AAAA
Revision: A02A
Software revision: G001
Serial number: 043P6982
Power Supply type: AC

SML Application information is:
Application file: /tffs0/temp.sli
Application name:
Application help:
Original source file:
H:\work\Emb\jrt\V2.5\sml\actions\drop\drop_basic_anyflow.san
Compilation date: Wed, September 22, 2004 at 21:25:21
Compiler version: SANc v2.50 Build 32 gcc_codelets=true built on: Tue
September 22 2004 09:51:57 AM.;SME plugin v1.1
Default capacity option used.


Logger status: Enabled


Platform: SCE 2000 - 4xFE
Management agent interface version: SCE Agent 2.5.1 Build 18
Software package file:
ftp://vk:vk@10.1.8.22/P:/EMB/LatestVersion/2.5.2/se1000.pkg

SCE 2000 uptime is 21 minutes, 37 seconds
SCE#
```

Another useful show commands is the **`show system-uptime`** command. This command displays information similar to the last line above, which indicates how long the system has been running since the last reboot.

To show the **s**ystem uptime for the ***SCE*** platform software and hardware, complete the following steps:

Step 1 At the ***SCE***# prompt, type **show system-uptime.**

The system shows how long the system has been running since the last reboot.

```
SCE#show system-uptime
SCE uptime is 21 minutes, 37 seconds
SCE#
```

Saving the Configuration Settings

When you make changes to the current running configuration and you want those changes to continue to be valid when the system restarts, you must save the changes before leaving the management session, that is, you must save the running configuration to the startup configuration file.

As mentioned before, ***SCE*** platform provides multiple interfaces for the purpose of configuration and management. All interfaces supply an API to the same database of the ***SCE*** platform and any configuration made through one interface is reflected through all interfaces. Furthermore, when saving the running configuration to the startup configuration from any management interface, all configuration settings are saved regardless of the management interface used to set the configuration.

To save configuration changes, complete the following steps:

Step 1 At the ***SCE***# prompt, type **show running-config** to view the running configuration.

The running configuration is displayed.

Step 2 Check the displayed configuration to make sure that it is set the way you want. If not, make the changes you want before saving.

Step 3 Type **copy running-config startup-config**.

The system saves all running configuration information to the configuration file, which is used when the system reboots.

The configuration file holds all information that is different from the system default in a file called config.txt located in the directory: tffs0:system.

EXAMPLE:

The following example shows the running configuration file.

```
SCE#show running-config
#This is a general configuration file (running-config).
#Created on 15:50:56  CET  MON  February  11  2002
#cli-type 1
#version 1

clock timezone CET 1
snmp-server community “public” ro
snmp-server host 10.1.1.253 traps version 1 “public”
interface LineCard 0
connection-mode active
no silent
no shutdown
flow-aging default-timeout UDP 60
interface FastEthernet 0/0
ip address 10.1.5.109 255.255.0.0
interface FastEthernet 0/1

interface FastEthernet 0/2

exit
line vty 0 4
no timeout
exit
SCE#
SCE#copy running-config startup-config
Writing general configuration file to temporary location...
Backing-up general configuration file...
Copy temporary file to final location...
SCE#
```

For backup purposes, the old startup-config file is saved under the directory: `tffs0:system/prevconf`. Refer to *Recovering a Previous Configuration* (on page 3-6) for an explanation on how to recover previous configuration.

To remove a configuration command from the running-config, use the no form of the command.

EXAMPLE:

The following example illustrates how to remove all DNS settings from the running configuration.

```
SCE(config)#no ip name-server
SCE(config)#
```

Recovering a Previous Configuration

When you save a new configuration, the system automatically backs up the old configuration in the directory *tffs0:system/prevconf/*. Up to nine versions of the startup configuration file are saved, namely *config.tx1-config.tx9*, where *config.tx1* is the most recently saved file.

You can view the old startup configuration files using the CLI command **more**.

Restoring a previous startup configuration means renaming the file so it overwrites the startup configuration (*config.txt*) file.

To restore a previous startup configuration, complete the following steps:

Step 1 At the ***SCE#*** prompt, type **more tffs0:system/prevconf/config.txt** to view the configuration file.

The system displays the configuration information stored in the file.

Step 2 Read the configuration information to make sure it is the configuration you want to restore.

Note that you cannot undo the configuration restore command.

Step 3 Type
copy tffs0:system/prevconf/config.tx1 tffs0:system/config.txt.

The system sets the startup configuration to the configuration from config.tx1.

EXAMPLE:

The following example displays a saved configuration file and then restores the file to overwrite the current configuration.

```
SCE#more tffs0:system/prevconf/config.tx1
#This is a general configuration file (running-config).
#Created on 19:36:07 UTC THU February 14 2002

#cli-type 1
#version 1

interface LineCard 0
no silent
no shutdown

interface FastEthernet 0/0
ip address 10.1.5.109 255.255.0.0

interface FastEthernet 0/1

interface FastEthernet 0/2

exit

line vty 0 4
exit
SCE#copy tffs0:system/prevconf/config.tx1 tffs0:system/config.txt
SCE#
```

Upgrading SCE Platform Firmware

Cisco distributes upgrades to the software and firmware on the ***SCE*** platform. Cisco distributes upgrade software as a file with the extension .pkg that is installed directly from the ftp site without being copied to the disk. This procedure walks you through installation and rebooting of the ***SCE*** platform with the new firmware.

To upgrade your ***SCE*** platform software:

Step 1 Type **`configure`** to enter Global Configuration mode.

The SCE prompt changes to ***SCE***`(config)#`.

Step 2 Type **`boot system ftp://<user:password@host/drive:dir/seNum.pkg>`**, where <seNum.pkg> is the file name on the ftp site.

The boot command verifies that the package is a legal, appropriate update for the ***SCE*** platform and that the file was not corrupted. It does not perform an upgrade, but does keep in the system memory that a pkg file is available.

Step 3 Type **`exit`** to leave the Global Configuration mode.

The SCE prompt changes to ***SCE***#.

Step 4 Type **`copy running-config startup-config`**.

This command re-verifies that the package is valid, and extracts the upgrade to the Flash file system.

The system notifies you that it is performing the extraction as follows:

```
Backing–up configuration file…
Writing configuration file…
Extracting new system image…
Extracted OK.
SCE#
```

Step 5 Type reload to reboot the system.

The ***SCE*** prompts you for confirmation by asking `Are you sure?`

Step 6 Type **Y** and press **Enter**.

The system sends the following message and reboots.

```
the system is about to reboot, this will end your CLI session
```

EXAMPLE:

The following example shows the full procedure for performing a software update.

```
SCE#configure
SCE(config)# boot system ftp://vk:vk@10.1.1.230/downloads/SENum.pkg
SCE(config)#exit
SCE#copy running-config startup-config
Backing–up configuration file…
Writing configuration file…
Extracting new system image…
Extracted OK.
SCE#>reload
Are you sure? y
the system is about to reboot, this will end your CLI session
```

Configuring Applications

The ***SCE*** platform can be configured to run with different Service Control applications by installing the appropriate file. All ***SCE*** platform application files are **pqi** files, that is, the filename must end with the *pqi* extension.

Once a specific Service Control application is installed it can be configured by applying a configuration file. The configuration file is application-specific, and is produced by application-specific means, not covered in this documentation. Configuration files have no specific extension.

Note These configuration changes are automatically saved to the start-up configuration after execution, and therefore do not appear when the running configuration is displayed (**`more running-config`** command).
These configurations cannot be manipulated by changing the *system/config.txt* file

Installing an Application

Use the following commands to install, uninstall, and upgrade an application. You can use the **`show pqi file`** command before installing or upgrading an application to display the options that are available when installing the pqi file. These options can then be specified in the **install** or **upgrade** command as needed.

The documentation of the application will tell the user whether the application is stand-alone (in which case **`install`** should be used), or an upgrade to an existing application that is assumed to be installed already (in this case **`upgrade`** should be used). Currently all Cisco Service Control applications are stand-alone.

You should always run the pqi unistall command before installing a new pqi file. This prevents old files from accumulating on the disk.

The following commands are relevant for installing and uninstalling an application:

- `pqi install file`
- `pqi uninstall file`
- `pqi upgrade file`
- `pqi rollback file`
- `show pqi file`
- `show pqi last-installed`

To display information about an application file:

Step 1 From the ***SCE#*** prompt, type **`show pqi file`** *`filename`* **`info`** and press **Enter**.

Information regarding the pqi file, such as installation options, is displayed and the ***SCE#*** prompt appears.

To install an application:

Step 1 From the ***SCE***`(config if)#` prompt, type **`pqi install file`** *`filename [options]`* and press **Enter**.

The specified pqi file is installed using the installation options specified (if any) and the ***SCE***`(config if)#` prompt appears.

Note that this may take up to 5 minutes.

Note Always run the pqi uninstall command before installing a new pqi file.

To uninstall an application:

Step 1 From the ***SCE***`(config if)#` prompt, type **`pqi uninstall file`** *`filename`* and press **Enter**.

The specified *pqi* file is uninstalled and the ***SCE***`(config if)#` prompt appears.

You must specify the same *pqi* file that was installed.

Note that this may take up to 5 minutes.

To upgrade an application:

Step 1 From the ***SCE***`(config if)#` prompt, type **`pqi upgrade file`** *`filename [options]`* and press **Enter**.

The specified pqi file is upgraded using the options specified (if any) and the ***SCE***`(config if)#` prompt appears.

You must specify the *pqi* file that was last used for upgrade.

Note that this may take up to 5 minutes.

To undo an upgrade of an application:

Step 1 From the ***SCE***`(config if)#` prompt, type **`pqi rollback file`** *`filename`* and press **Enter**.

The upgrade of the specified pqi file is undone and the ***SCE***`(config if)#` prompt appears.

Note that this may take up to 5 minutes.

To display the last pqi file that was installed:

Step 1 From the ***SCE***# prompt, type **`show pqi last-installed`** and press **Enter**.

The name of the last pqi file that was installed is displayed and the ***SCE***# prompt appears.

Configuring the Currently Installed Application

Use the following commands to:

- Validate the configuration file
- Configure the currently installed application by applying the configuration file
- Display the name of the last configuration file that was applied

The following commands are relevant for configuring the currently installed application:

- `scm apply file`
- `scm validate file`
- `show scm last-applied`

To validate a configuration file:

Step 1 From the ***SCE#*** prompt, type **`scm validate file`** *`filename`* and **press** Enter.

The specified configuration file is checked and the ***SCE#*** prompt appears.

To apply a configuration file:

Step 1 From the ***SCE***`(config if)#` prompt, type **`scm apply file`** *`filename`* and press **Enter**.

The specified configuration file is applied and the ***SCE***`(config if)#` prompt appears.

To display the last configuration file that was applied:

Step 1 From the ***SCE#*** prompt, type **`show scm last-applied`** and press **Enter**.

The name of the last configuration file that was applied is displayed and the ***SCE#*** prompt appears.

Rebooting and Shutting Down the SCE Platform

Rebooting the SCE Platform

Rebooting the ***SCE*** platform is required after installing a new firmware, in order for that firmware to take effect. There might be other occasions where rebooting the ***SCE*** platform is necessary.

Note When the SCE restarts, it loads the startup configuration, so all changes made in the running configuration will be lost. You are advised to save the running configuration before performing reload, as described in *Saving the Configuration Settings* (on page 3-4).

To reboot your ***SCE*** platform, complete the following steps:

Step 1 At the ***SCE***# prompt, type **`reload`** and press **Enter**.

A confirmation message appears.

Step 2 Type **Y** to confirm the reboot request and press **Enter**.

EXAMPLE:

The following example shows the commands for system reboot.

```
SCE# reload
Are you sure? y
the system is about to reboot, this will end your CLI session
```

Shutting Down the SCE Platform

Shutting down the ***SCE*** platform is required before turning the power off. This helps to ensure that non-volatile memory devices in the ***SCE*** platform are properly flushed in an orderly manner.

Note When the SCE restarts, it loads the startup configuration, so all changes made in the running configuration will be lost. You are advised to save the running configuration before performing reload, as described in *Saving the Configuration Settings* (on page 3-4).

To shut down your ***SCE*** platform, complete the following steps:

Step 1 Connect to the serial console port (The CON connector on the ***SCE*** platform front panel, 9600 baud).

The ***SCE***# prompt appears.

Step 2 Type **`reload shutdown`**.

A confirmation message appears.

Step 3 Type **Y** to confirm the shutdown request and press **Enter**.

EXAMPLE:

The following example shows the commands for system shutdown.

```
SCE#reload shutdown
You are about to shut down the system.
The only way to resume system operation after this
is to cycle the power off, and then back on.
Continue?
y

IT IS NOW SAFE TO TURN THE POWER OFF.
```

Note Since the ***SCE*** platform can recover from the power-down state only by being physically turned off (or cycling the power), this command can only be executed from the serial CLI console. This limitation helps prevent situations in which a user issues this command from a Telnet session, and then realizes he/she has no physical access to the ***SCE*** platform.

Utilities

This chapter contains the following sections:

- Setup Utility 4-1
- File-system Operations 4-3
- The User Log 4-8

Setup Utility

The setup utility is an interactive wizard that guides the user through the basic configuration process. This utility runs automatically upon initial connection to the local terminal. It may also be invoked explicitly via Telnet or via the local terminal to make changes to the system configuration.

Entering the Setup Utility

Step 1 Press **Enter** several times until the Cisco logo appears on the local terminal and the setup configuration dialog is entered.

```
                --- System Configuration Dialog ---

At any point you may enter a question mark '?' followed by 'Enter' for help.
Use ctrl-C to abort configuration dialog at any prompt.
Use ctrl-Z to jump to the end of the configuration dialog at any prompt.
Default settings are in square brackets '[]'.

Would you like to continue with the System Configuration Dialog? [yes/no]: y
```

Step 2 Type y and press **Enter**.

The system configuration dialog begins.

Multiple entry parameters (Lists)

When explicitly invoked, the setup utility offers the option of multiple entries (lists) for certain parameters.

Several parameters, such as the Access Control Lists, are actually lists containing a number of entries. If these lists are empty (initial configuration) or contain only one entry, they act the same as any scalar parameter, except that you can add additional entries to the list.

If these lists already contain more than one entry, the entire list is displayed, and you are then presented with several options. Following is an excerpt from the SNMP trap manager menu, illustrating how to configure list entries.

To configure a list parameter when more than one entry already exists in the list:

Step 1 The entries in the list are displayed.

```
There are 2 SNMP trap managers in the current configuration as follows:
IP address: 10.10.10.10  Community: private  Version: 1
IP address: 10.11.10.1   Community: pcube    Version: 2c
```

Note If only one entry exists in the table, it is displayed as the default [ ] to be either accepted or changed. The three list options are not displayed.

Step 2 Three options are presented.

```
Please choose one of the following options:
1. Leave the running configuration unchanged.
2. Clear the existing lists and configure new ones.
3. Add new entries.

Enter your choice:
```

Step 3 You are prompted to continue the setup, depending on the choice you entered:

- 1. Leave the running configuration unchanged:

  The dialog proceeds to the next question. The list remains unchanged.

- 2. Clear the existing entries and configure new ones:

  The dialog prompts you for a new entry in the list.

  After completing the first entry, you are asked whether you would like to add another entry.

```
Would you like to add another SNMP trap manager? [no]:y
```

  Since the list was empty, you may enter the maximum number of entries.

- 3. Add new entries:

  The dialog prompts you for a new entry in the list.

  After the completing one entry, you are asked whether you would like add another new entry.

```
Would you like to add another SNMP trap manager? [no]:y
```

  You may enter only enough additional entries to reach the maximum number.

File-system Operations

The CLI commands include a complete range of file management commands. These commands allow you to create, delete, copy, and display both files and directories.

Note Regarding disk capacity: While performing disk operations, the user should take care that the addition of new files that are stored on the SCE disk do not cause the disk to exceed 70% utilization.

Working with Directories

The following file-system operations commands are relevant to directories:

- `cd`
- `delete`
- `dir`
- `mkdir`
- `pwd`
- `rmdir`

Creating a Directory

To create a directory:

Step 1 From the ***SCE#*** prompt, type **`mkdir`** *`directory-name`* and press **Enter**.

The specified directory is created and the ***SCE#*** prompt appears.

Deleting a Directory

There are two different commands for deleting a directory, depending on whether the directory is empty or not.

Use this command to delete a directory along with all of its contents.

To delete a directory and all its files and sub-directories:

Step 1 From the ***SCE#*** prompt, type **`delete`** *`directory-name`* **`/recursive`** **and** press Enter.

The specified directory, including all files and sub-directories, is deleted, and the ***SCE#*** prompt appears.

Use this command to remove an empty directory.

To delete an empty directory:

Step 1 From the ***SCE#*** prompt, type **`rmdir`** *`directory-name`* and press **Enter**.

The specified directory is deleted and the ***SCE#*** prompt appears.

Changing Directories

To change the path of the current working directory:

Step 1 From the ***SCE#*** prompt, type **`cd`** *`new path`* and press **Enter**.

The specified directory becomes the working directory and the ***SCE#*** prompt appears.

Displaying Working Directory

To display the current working directory:

Step 1 From the ***SCE#*** prompt, type **`pwd`** and press **Enter**.

The name of the working directory is displayed and the ***SCE#*** prompt appears.

Listing Files in Current Directory

You can display a listing of all files in the current working directory. This list may be filtered to include only application files. The listing may also be expanded to include all files in any sub-directories.

To list all the files in the current directory:

Step 1 From the ***SCE#*** prompt, type **`dir`** and press **Enter**.

A listing of all files in the working directory is displayed and the ***SCE#*** prompt appears.

To list all the applications in the current directory:

Step 1 From the ***SCE#*** prompt, type **`dir applications`** and press **Enter**.

A listing of all application files in the working directory is displayed and the ***SCE#*** prompt appears.

To include files in all sub-directories in the listing of the current directory:

Step 1 From the ***SCE#*** prompt, type **`dir -r`** and press **Enter**.

A listing of all files in the working directory, including all files in all sub-directories, is displayed and the ***SCE#*** prompt appears.

Working with Files

The following file-system operations commands are relevant to files:

- `copy`
- `copy-passive`
- `delete`
- `more`
- `rename`
- `unzip`

Renaming a File

To rename a file:

Step 1 From the ***SCE#*** prompt, type **`rename`** *`current-file-name new-file-name`* and press **Enter**.

The specified file is renamed and the ***SCE#*** prompt appears.

Deleting a File

To delete a file:

Step 1 From the ***SCE#*** prompt, type **`delete`** *`file-name`* and press **Enter**.

The specified file is deleted and the ***SCE#*** prompt appears.

Copying a File

You can copy a file from the current directory to a different directory.

You can also copy a file (upload/download) to or from an FTP site. In this case, either the source or destination filename must begin with *ftp://*. To copy a file using passive FTP, use the **`copy-passive command.`**

To copy a file:

Step 1 From the ***SCE#*** prompt, type **`copy`** *`source-file-name destination-file-name`* and press **Enter**.

The file is copied to the specified directory and the ***SCE#*** prompt appears.

EXAMPLE:

The following example copies the local *`analysis.sli`* file located in the root directory to the *`applications`* directory.

```
SCE#copy analysis.sli applications/analysis.sli
SCE#
```

To download a file from an FTP site:

Step 1 From the ***SCE#*** prompt, type **`copy`** *`ftp://source destination-file-name`* and press **Enter**.

The file is downloaded from the FTP site to the specified directory and the ***SCE#*** prompt appears.

To upload a file to an FTP site using Passive FTP:

Step 1 From the ***SCE#*** prompt, type **copy-passive** *source-file-name ftp://destination* and press **Enter**.

The file is uploaded to the specified FTP site and the ***SCE#*** prompt appears.

EXAMPLE:

The following example uploads the *analysis.sli* file located on the local flash file system to the host 10.1.1.105, specifying Passive FTP.

```
SCE#copy-passive /appli/analysis.sli
ftp://myname:mypw@10.1.1.105/p:/appli/analysis.sli
SCE#
```

Displaying File Contents

To display the contents of a file:

Step 1 From the ***SCE#*** prompt, type **more** *file-name* and press **Enter**.

The contents of the specified file are displayed and the ***SCE#*** prompt appears.

Unzipping a File

Use this command to unzip a file. The specified file must be a zip file.

Files are extracted to the current directory.

To unzip a file:

Step 1 From the ***SCE#*** prompt, type **unzip** *file-name* and press **Enter**.

The specified file is extracted to the current directory and the ***SCE#*** prompt appears.

The User Log

The user log is an ASCII file that can be viewed in any editor. It contains a record of system events, including startup, shutdown and errors. You can view the user log to determine whether or not the system is functioning properly, as well as for technical support purposes.

The Logging System

Events are logged to one of two log files. After a file reaches maximum capacity, the events logged in that file are then temporarily archived. New events are then automatically logged to the alternate log file. When the second log file reaches maximum capacity, the system then reverts to logging events to the first log file, thus overwriting the temporarily archived information stored in that file.

Basic operations include:

- Copying the User Log to an external source
- Viewing or clearing the User Log
- Viewing/clearing the User Log counters

Enabling and Disabling the User Log

By default, the user log is enabled. You can disable the user log by configuring the status of the logger.

To disable the user log:

Step 1 From the ***SCE***# prompt, type **`configure`** and press **Enter**.

The ***SCE***`(config)#` prompt appears indicating that you are in Global Configuration mode.

Step 2 Type **`logger device User-File-Log disabled`** and press **Enter**.

The ***SCE***`(config)#` prompt appears.

To enable the user file log:

Step 1 From the ***SCE***# prompt, type **`configure`** and press **Enter**.

The ***SCE***`(config)#` prompt appears.

Step 2 Type **`logger device User-File-Log enabled`** and press **Enter**.

The ***SCE***`(config)#` prompt appears.

Copying the User Log

You can view the log file by copying it to an external source or to disk. This command copies both log files to the local ***SCE*** platform disk or any external host running a FTP server.

To copy the user log to an external source, complete the following steps:

Step 1 From the ***SCE#*** prompt, type **`logger get user-log file-name`** *`ftp://username:password@ipaddress/path`* and press **Enter**.

The ***SCE#*** prompt appears.

To copy the user log to an internal location, complete the following steps:

Step 1 From the ***SCE#*** prompt, type **`logger get user-log file-name`** *`target-filename`* and press **Enter**.

Viewing the User Log Counters

There are two types of log counters:

- User log counters: count the number of system events logged from the ***SCE*** platform last reboot.
- Non-volatile counters: are not cleared during boot time

To view the user log counters for the current session, complete the following steps:

Step 1 From the ***SCE#*** prompt, type **`show logger device user-file-log counters`** and press **Enter**.

The logger lines information appears, followed by the ***SCE#*** prompt.

To view the non-volatile logger counters for both the User log file and the debug log file, complete the following steps:

Step 1 From the ***SCE#*** prompt, type **`show logger nv-counters`** and press **Enter**.

The non-volatile log counter information appears, followed by the ***SCE#*** prompt.

To view the non-volatile counter for the user-file-log only, complete the following steps:

Step 1 From the ***SCE***# prompt, type **`show logger device user-file-log nv-counters`** and press **Enter**.

The user-file-log non-volatile log counter information appears, followed by the ***SCE***# prompt.

Viewing the User Log

Note This command is not recommended when the user log is large. Copy a large log to a file to view it (see *Copying the User Log* (on page 4-9))

To view the user log, complete the following steps:

Step 1 From the ***SCE***# prompt, type **`more user log`** and press **Enter**.

The user log appears, followed by the ***SCE***# prompt.

Clearing the User Log

You can clear the contents of the user log at any time. The user log contains important information regarding the functioning of the system. It is recommended that a copy be made before the log is cleared.

To clear the user log, complete the following steps:

Step 1 From `the` ***SCE***# prompt, type **clear logger device user-file-log** and press **Enter**.

Step 2 The system asks `Are you sure?`

Step 3 Type **`Y`** and press **Enter**.

The ***SCE***# prompt appears.

Generating a File for Technical Support

In order for technical support to be most effective, the user should provide them with the information contained in the system logs. Use the **logger get support-file** command to generate a support file for the use of Cisco technical support staff.

To generate a log file for technical support, complete the following steps:

Step 1 From the ***SCE#*** prompt, type **logger get support-file** *filename* and press **Enter**.

The support information file is created using the specified filename, and the ***SCE#*** prompt appears. This operation may take some time.

CHAPTER 5

Configuring the Management Interface and Security

This chapter contains the following sections:

- Configuring the Available Interfaces 5-1
- SNMP Configuration and Management 5-9
- Passwords 5-20
- IP Configuration 5-22
- Time Clocks and Time Zone 5-26
- SNTP 5-31
- Domain Name (DNS) Settings 5-34
- Management Interface Configuration Mode 5-37

Configuring the Available Interfaces

The system allows you to configure the Telnet and SNMP interfaces according to how you are planning to manage the ***SCE*** platform and the external components of the system.

Configuring Access Control Lists (ACLs)

The ***SCE*** platform can be configured with Access Control Lists (ACLs), which are used to permit or deny incoming connections on any of the management interfaces. An access list is an ordered list of entries, each consisting of an IP address and an optional wildcard "mask" defining an IP address range, and a permit/deny field.

The order of the entries in the list is important. The default action of the first entry that matches the connection is used. If no entry in the Access List matches the connection, or if the Access List is empty, the default action is `deny`.

Configuration of system access is done in two stages:

Step 1 Creating an access list. (See *Adding Entries to an Access List* (on page 5-3)).

Step 2 Associating the access list with a management interface. (See *Defining the Global Access List* (on page 5-4) and *Associating an Access List to Telnet Interface.* ("Associating an Access List to Telnet Interface" on page 5-5))

Creating an access list is done entry by entry, from the first to the last.

When the system checks for an IP address on an access list, the system checks each line in the access list for the IP address, starting at the first entry and moving towards the last entry. The first match that is detected (that is, the IP address being checked is found within the IP address range defined by the entry) determines the result, according to the permit/deny flag in the matched entry. If no matching entry is found in the access list, access is denied.

You can create up to 99 access lists. Access lists can be associated with system access on the following levels:

- Global (IP) level. If a global list is defined using the `ip access-class` command, when a request comes in, the ***SCE*** platform first checks if there is permission for access from that IP address. If not, the SCE does not respond to the request. Configuring the ***SCE*** platform to deny a certain IP address would preclude the option of communicating with that address using any IP-based protocol including Telnet, FTP, ICMP and SNMP. The basic IP interface is low-level, blocking the IP packets before they reach the interfaces.
- Interface level. Access to each management interface (Telnet, SNMP, etc.) can be restricted to an access list. Interface-level lists are, by definition, a subset of the Global list defined. If access is denied at the global level, the IP will not be allowed to access using one of the interfaces. Once an access list is associated with a specific management interface, that interface checks the access list to find out if there is permission for a specific external IP address trying to access the management interface.

It is possible to configure several management interfaces to the same access list, if this is the desired behavior of the ***SCE*** platform.

If no ACL is associated to a management interface or to the global IP level, access is permitted from all IP addresses.

Note The SCE Platform will respond to `ping` commands only from IP addresses that are allowed access. Ping from a non-authorized address will not receive a response from the SCE unit, as ping uses ICMP protocol

The following commands are relevant to access lists:

- `access-list`
- `access-class number in`
- `ip access-class`
- `no access-list`
- `no ip access-class`
- `show ip access-class`

Adding Entries to an Access List

To add an address to an access list allowing access to a particular address:

Step 1 To enter the Global Configuration Mode, type **`configure`** and press **Enter**.

Step 2 The ***SCE***`(config)#` prompt appears.

Step 3 To configure one IP address type:

`access-list` `number` **`permit`** `x.x.x.x` and press **Enter** where `x.x.x.x` is the IP address.

Step 4 To configure more than one IP address type:

`access-list` `number` **`permit`** `x.x.x.x y.y.y.y` and press **Enter**.

This command configures a range of addresses in the format `x.x.x.x y.y.y.y` where `x.x.x.x` specifies the prefix bits common to all IP addresses in the range, and `y.y.y.y` is a wildcard-bits mask specifying the bits that are ignored. In this notation, '1' means bits to ignore.

EXAMPLE:

The following example adds an entry to the access list number 1, that permits access only to IP addresses in the range of 10.1.1.0–10.1.1.255.

```
SCE(config)#access-list 1 permit 10.1.1.0 0.0.0.255
```

You can also add addresses from which you deny service, by using the **`deny`** rather than the **`permit`** switch. You can create up to 99 different address lists, which can be associated with access to the interfaces.

When you add a new entry to an ACL, it is always added to the end of the Access-List.

Removing an Access List

To remove an Access List (with all its entries):

Step 1 From the ***SCE***`(config)#` prompt, type **`no access-list`** *number permit/deny*, and press **Enter**.

The Access List and all of its entries are removed.

Defining the Global Access List

To define an Access List as the global list for permitting or denying all traffic to the ***SCE*** platform:

Step 1 From the ***SCE***`(config)#` prompt, type `ip access-class` *number*, and press **Enter**.

Telnet Interface

This section discusses the Telnet interface of the ***SCE*** platform. A Telnet session is the most common way to connect to the ***SCE*** CLI interface.

You can set the following parameters for the Telnet interface:

- Enable/disable the interface
- Associate an access list to permit or deny incoming connections. (Access lists)
- Timeout for Telnet sessions, that is, if there is no activity on the session, how long the ***SCE*** platform waits before automatically cutting off the Telnet connection.

The following commands are relevant to Telnet interface:

- `access-class number in`
- `line vty`
- `[no] access list`
- `[no] service telnetd`
- `[no] timeout`
- `show line vty access-class in`
- `show line vty timeout`

Preventing Telnet Access

You can disable access by Telnet altogether.

To disable Telnet access:

Step 1 From the ***SCE***`(config)#` prompt, type **`no service telnetd`**.

Telnet service is no longer allowed on the ***SCE*** platform. Current Telnet sessions are not disconnected, but no new Telnet sessions are allowed.

Associating an Access List to Telnet Interface

To restrict the ***SCE*** platform management via Telnet to a specific access list:

Step 1 From the ***SCE***`(config)#` prompt, enter the Line Configuration mode by typing line vty 0.

Step 2 Type **`access-class`** *access-list-number* **`in`** (where `access-list-number` is an index of an existing access list).

The following example associates the access list number 1 to the Telnet interface.

```
SCE#configure
SCE (config)#line vty 0
SCE(config-line)#access-class 1 in
```

Step 3 Type **`exit`** and press **Enter**.

This returns you to Global Configuration Mode.

Telnet Timeout

The ***SCE*** platform supports timeout of inactive Telnet sessions. The default timeout is 30 minutes.

To configure the timeout for a telnet session when the line is idle:

Step 1 From the ***SCE***`(config-line)#` prompt, type **`timeout`** time, where time is the time in minutes.

SSH Server

A shortcoming of the standard telnet protocol is that it transfers password and data over the net unencrypted, thus compromising security. Where security is a concern, using a Secure Shell (SSH) server rather than telnet is recommended.

An SSH server is similar to a telnet server, but it uses cryptographic techniques that allow it to communicate with any SSH client over an insecure network in a manner which ensures the privacy of the communication. CLI commands are executed over SSH in exactly the same manner as over telnet.

The SSH server supports both the SSH-1 and SSH-2 protocols.

An Access Control List (ACL) can be configured for SSH as for any other management protocol, limiting SSH access to a specific set of IP addresses (see *Configuring Access Control Lists* ("Configuring Access Control Lists (ACLs)" on page 5-1)).

Key Management

Each SSH server should define a set of keys (DSA2, RSA2 and RSA1) to be used when communicating with various clients. The key sets are pairs of public and private keys. The server publishes the public key while keeping the private key in non-volatile memory, never transmitting it to SSH clients. Note that the keys are kept on the tffs0 file system, which means that a person with knowledge of the 'enable' password can access both the private and public keys. The SSH server implementation provides protection against eaves-droppers who can monitor the management communication channels of the ***Service Control Application for Broadband*** platform, but it does not provide protection against a user with knowledge of the 'enable' password.

Key management is performed by the user via a special CLI command. A set of keys must be generated at least once before enabling the SSH server.

Size of the encryption key is always 2048 bits.

Managing the SSH Server

Use these commands to manage the SSH server. These commands do the following:

- Generate an SSH key set
- Enable/disable the SSH server
- Assign/remove an ACL to the SSH server
- Delete existing SSH keys

Remember that you must generate a set of SSH keys before you enable the SSH server.

To generate a set of SSH keys:

Step 1 From the ***SCE***(config)# prompt, type **`ip ssh key generate`** and press **Enter**.

A new SSH key set is generated and immediately saved to non-volatile memory. (Key set is not part of the configuration file). Key size is always 2048 bits.

To enable the SSH server:

Step 1 From the ***SCE***(config)# prompt, type **`ip ssh`** and press **Enter**.

To disable the SSH server:

Step 1 From the ***SCE***(config)# prompt, type **`no ip ssh`** and press **Enter**.

To assign an ACL to the SSH server:

Step 1 From the ***SCE***(config)# prompt, type **`ip ssh access-class`** *`access-list number`* and press **Enter**.

The specified ACL is assigned to the SSH server, so that access the SSH server is limited to the IP addresses defined in the ACL.

To remove the ACL assignment from the SSH server:

Step 1 From the ***SCE***(config)# prompt, type **`no ip ssh access-class`** and press **Enter**.

The ACL assigment is removed from the SSH server, so that any IP address may now access the SSH server.

To delete the existing SSH keys:

Step 1 From the ***SCE***(config)# prompt, type **`ip ssh key remove`** and press **Enter**.

The existing SSH key set is removed from non-volatile memory.

If the SSH server is currently enabled, it will continue to run, since it only reads the keys from non-volatile memory when it is started. However, if the startup-configuration specifies that the SSH server is enabled, the ***SCE*** platform will not be able to start the SSH server on startup if the keys have been deleted. To avoid this situation, after executing this command, always do one of the following before the ***SCE*** platform is restarted (using reload):

- Generate a new set of keys.
- Disable the SSH server and save the configuration.

Monitoring the Status of the SSH Server

Use this command to monitor the status of the SSH sever, including current SSH sessions.

This command is a Privileged Exec command. Make sure that you are in Privileged Exec command mode by exiting any other modes.

To display the SSH server status:

Step 1 From the ***SCE***# prompt, type **`show ip ssh`** and press **Enter**.

SNMP Interface

To enable the SNMP interface, use the **snmp-server** command. You can also configure any of the SNMP parameters: hosts, communities, contact, location, and trap destination host. When you enable the SNMP agent, these four parameters are filled in with their most recent values before the agent was disabled. To disable the SNMP interface, use the **no snmp-server** command.

This section guides you through enabling and disabling the SNMP interface. Complete information on SNMP is found in *SNMP Configuration and Management* (on page 5-9).

The following commands are relevant to enabling and disabling the SNMP interface:

- `[no] snmp-server`
- `[no] snmp-server community`
- `[no] snmp-server contact`
- `[no] snmp-server host`
- `[no] snmp-server location`

Enabling SNMP

To enable SNMP by setting a community string:

Step 1 To enter the Global Configuration Mode, at the ***SCE***# prompt, type `configure` and press **Enter**.

The ***SCE***`(config)`# prompt appears.

Step 2 Type **snmp-server community** `community-string`, where the `community string` is a security string that identifies a community of managers that are able to access the SNMP server.

You must define at least one community string in order to allow SNMP access. For complete information on community strings see *Configuring SNMP Community Strings* (on page 5-11).

Disabling SNMP

To disable SNMP access:

Step 1 From the ***SCE***`(config)`# prompt, type **no snmp-server**.

SNMP Configuration and Management

The ***SCE*** platform operating system includes a Simple Network Management Protocol (SNMP) agent that supports the RFC 1213 standard (MIB-II) and Cisco's enterprise MIBs. This section explains how to configure the SNMP agent parameters. It also describes the SNMP traps and the Cisco proprietary MIB, and explains the order in which the MIB must be loaded.

Note Throughout this manual, the terms SNMP server and SNMP agent are used interchangeably, as equivalents.

SNMP Protocol

SNMP (Simple Network Management Protocol) is a set of protocols for managing complex networks. SNMP works by sending messages, called protocol data units (PDUs), to different parts of a network. SNMP-compliant devices, called agents, store data about themselves in Management Information Bases (MIBs) and return this data to the SNMP requesters.

SCE platform supports the original SNMP protocol (also known as SNMPv1), and a newer version called Community-based SNMPv2 (also known as SNMPv2C).

- **SNMPv1:** is the first version of the Simple Network Management Protocol, as defined in RFCs 1155 and 1157, and is a full Internet standard. SNMPv1 uses a community-based form of security.
- **SNMPv2c:** is the revised protocol, which includes improvements to SNMPv1 in the areas of protocol packet types, transport mappings, and MIB structure elements but using the existing SNMPv1 administration structure. It is defined in RFC 1901, RFC 1905, and RFC 1906.

SCE platform implementation of SNMP supports all MIB II variables, as described in RFC 1213, and defines the SNMP traps using the guidelines described in RFC 1215.

The SNMPv1 and SNMPv2C specifications define the following basic operations that are supported by ***SCE*** platform:

Table 5-1 Request Types

Request Type	Description	Remarks
Set Request	Writes new data to one or more of the objects managed by an agent.	Set operations immediately affect the ***SCE*** platform running-config but do not affect the startup config.
Get Request	Requests the value of one or more of the objects managed by an agent.	
Get Next Request	Requests the Object Identifier(s) and value(s) of the next object(s) managed by an agent.	
Get Response	Contains the data returned by an agent.	
Trap	Sends an unsolicited notification from an agent to a manager, indicating that an event or error has occurred on the agent system	***SCE*** platform may be configured to send either SNMPv1 or SNMPv2 style traps.
Get Bulk Request	Retrieves large amounts of object information in a single Request / response transaction. GetBulk behaves as if many iterations of GetNext request/responses were issued, except that they are all performed in a single request/response.	This is newly defined SNMPv2c message.

Configuration via SNMP

SCE platform supports a limited set of variables that may be configured via SNMP (read-write variables). Setting a variable via SNMP (as via the CLI) takes effect immediately and affects only the running-configuration. To make this configuration stored for next reboots (startup-configuration) the user must specify it explicitly via CLI or via SNMP using the Cisco enterprise MIB objects (see the figure in *Cisco Enterprise MIB* (on page 5-18)).

It should be noted also that the ***SCE*** platform takes the approach of a single configuration database with multiple interfaces that may change this database. Therefore, activating the `copy running-config startup-config` command via CLI or SNMP makes permanent all the changes made by either SNMP or CLI.

Security Considerations

By default, the SNMP agent is disabled for both read and write operations. When enabled, SNMP is supported over the management port only (in-band management is not supported).

In addition, ***SCE*** platform supports the option to configure community of managers for read-write accessibility or for read-only accessibility. Furthermore, an ACL (Access List) may be associated with a community to allow SNMP management to a restricted set of managers IP addresses.

SNMP Community Strings

An SNMP community string is a text string that acts like a password to permit access to the agent on the ***SCE*** platform. The community string is used to authenticate messages that are sent between the management station (the SNMP manager) and the device (the SNMP agent). The community string is included in every message transmitted between the SNMP manager and the SNMP agent.

Configuring SNMP Community Strings

In order to enable SNMP management, you must configure SNMP community strings to define the relationship between the SNMP manager and the agent.

After receiving an SNMP request, the SNMP agent compares the community string in the request to the community strings that are configured for the agent. The requests are valid under the following circumstances:

- SNMP *Get* and *Get-next*, *Get-bulk* requests are valid if the community string in the request matches the read-only community.
- SNMP *Get*, *Get-next*, *Get-bulk* and *Set* requests are valid if the community string in the request matches the agent's read-write community.

You may specify the following characteristics associated with the community string:

- An access list of IP addresses of the SNMP managers permitted to use the community string to gain access to the agent
- Read-write or read-only accessibility for the community.

Note If no access list is configured, all IP addresses can access the agent using the defined community string. For more information about Access Lists, see *Configuring Access Control Lists (ACLs)* (on page 5-1)

Note When defining a community if it is not specified explicitly, the default accessibility is read-only.

The following describes how to configure a community string, as well as how to remove a community string.

To configure a community string:

Step 1 At the ***SCE***`(config)#` prompt, type **`snmp-server community`** *`community-string`* **`[ro|rw]`** `[acl-number]`, and press **Enter**.

The ***SCE***`(config)#` prompt appears.

Step 2 If needed, repeat steps 1 to configure additional community strings.

EXAMPLE:

The following example shows how to configure a community string called “mycommunity” with read-only rights and access list number “1”.

```
SCE(config)#snmp-server community mycommunity 1
```

Note ACL-number is an index to an access list. For further information about access lists, see *Configuring Access Control Lists (ACLs)* (on page 5-1)

To remove a community string:

Step 1 At the ***SCE***`(config)#` prompt, type **`no snmp-server community`** *`community-string`*, and press **Enter**.

The community string is removed.

EXAMPLE:

The following example displays how to remove a community string called “mycommunity”.

```
SCE(config)#no snmp-server community mycommunity
```

To display the configured communities:

Step 1 At the ***SCE***`#` prompt, type **`show snmp community`** and press **Enter**.

The configured SNMP communities appear.

EXAMPLE:

The following example shows the SNMP communities.

```
SCE#show snmp community
Community: public, Access Authorization: RO, Access List Index: 1
```

Traps

Traps are unsolicited messages that are generated by the SNMP agent that resides inside the ***SCE*** platform when an event occurs. When the Network Management System receives the trap message, it can take suitable actions, such as logging the occurrence or ignoring the signal.

Configuring Traps

By default, the ***SCE*** platform is not configured to send any SNMP traps. You must define the Network Management System to which the ***SCE*** platform should send traps. (See the table below, Configurable Traps, for a list of configurable traps). Whenever one of the events that trigger traps occurs in the ***SCE*** platform, an SNMP trap is sent from the ***SCE*** platform to the list of IP addresses that you define.

SCE platform supports two general categories of traps:

- Standard SNMP traps: As defined in RFC1157 and using the conventions defined in RFC1215.
- Proprietary SCE enterprise traps: As defined in the SCE proprietary MIB.

After a host is configured to receive traps, by default, the ***SCE*** platform sends to this host all the traps supported by the ***SCE*** platform except for the AuthenticationFailure trap. The ***SCE*** platform provides the option to enable or disable the sending of this trap, as well as some of the SCE enterprise traps, explicitly.

SCE platform can be configured to generate either SNMPv1 style or SNMPv2c style traps. By default, the ***SCE*** platforms sends SNMPv1 traps.

Following the table are sample procedures displaying how to configure a host (NMS) to which the SNMP agent should send traps; how to enable the SNMP agent to send authentication-failure traps; how to reset all traps to the default setting, and how to remove/disable a host (NMS) from receiving traps.

Table 5-2 Configurable Traps

Traps	Description	Trap Names	Default
Standard Traps			
Authentication Failure	An authenticationFailure trap is sent when the ***SCE*** platform is the addressee of a protocol message that is not properly authenticated.	authenticationFailure	Disabled
Enterprise Traps			
attack filter	An attack filter trap is sent when an attack filter has been activated or deactivated. The type of attack-filter that was activated is returned in pcubeSeEventGenericString1	moduleAttackFilterActivatedTrap moduleAttackFilter DeactivatedTrap	Disabled

Traps	Description	Trap Names	Default
chassis	A chassis trap is sent when an environmental alarm condition occurs in the ***SCE*** platform or is resolved.	chassisTempAlarmOnTrap chassisTempAlarmOffTrap chassisVoltageAlarmOnTrap chassisFansAlarmOnTrap chassisPowerSupplyAlarmOnTrap	Enabled
link-bypass	A link-bypass trap is sent when the ***SCE*** platform recognizes that the link-bypass mode has changed (bypass, no bypass, cutoff).	linkModeBypassTrap linkModeNoBypassTrap linkModeCutoffTrap	Enabled
logger	A logger trap is sent when the ***SCE*** platform recognizes that the User log is full. The ***SCE*** platform rolls over to the next log file.	loggerUserLogIsFullTrap	Enabled
operational-status	An operational-status trap is sent when the ***SCE*** platform recognizes that the operational status has changed (the ***SCE*** platform fails, resumes operation, or detects a warning).	OperationalStatusOperationalTrap operationalStatusWarningTrap operationalStatusFailureTrap	Enabled
rdr-formatter	An rdr-formatter trap is sent when the ***SCE*** platform recognizes a change in the status of the connection of the rdr-formatter to the data collector (up, down, active, not active).	rdrActiveConnectionTrap rdrNoActiveConnectionTrap rdrConnectionUpTrap rdrConnectionDownTrap	Enabled
sntp	An sntp trap is sent when the ***SCE*** platform recognizes that the SNTP agent has not updated the time in a long enough interval that time drift may occur in the system.	sntpClockDriftWarnTrap	Enabled
system-reset	A system-reset trap is sent before the ***SCE*** platform performs a system reset, due either to user request or fatal event.	systemResetTrap	Enabled
telnet	A telnet trap is sent when the ***SCE*** platform recognizes that a telnet session has started or ended. A telnet trap is also sent when an attempt is made to logon from an unauthorized source, or with the wrong password.	SessionStartedTrap SessionEndedTrap SessionDeniedAccessTrap SessionBadLoginTrap	Enabled

To configure the ***SCE*** platform to send traps to a host (NMS):

Step 1 At the ***SCE***`config)#` prompt, type **`snmp-server host`** *`IP-address community-string`*, and press **Enter**.

The ***SCE***`(config)#` prompt appears.

EXAMPLE:

The following example shows how to configure the ***SCE*** platform to send SNMPv1 traps to a host with the IP Address: `192.168.0.83` and community string named `mycommunity`.

```
SCE(config)#snmp-server host 192.168.0.83 mycommunity
```

To enable the SNMP server to send AuthenticationFailure traps:

Step 1 At the ***SCE***`(config)#` prompt, type **`snmp-server enable traps snmp authentication`**, and press **Enter**.

The SNMP server is enabled to send **`authentication failure`** traps.

EXAMPLE:

The following example shows how to configure the SNMP server to send the Authentication failure trap.

```
SCE(config)#snmp-server enable traps snmp authentication
```

You may enable or disable a specific enterprise trap or all enterprise traps.

To enable the SNMP server to send all Enterprise traps:

Step 1 At the ***SCE***`(config)#` prompt, type **`snmp-server enable traps enterprise`**, and press **Enter**.

The SNMP server is enabled to send all **`enterprise`** traps.

EXAMPLE:

The following example shows how to configure the SNMP server to send all enterprise traps.

```
SCE(config)#snmp-server enable traps enterprise
```

To enable the SNMP server to send a specific Enterprise trap:

Step 1 At the ***SCE***(config)# prompt, type **snmp-server enable traps enterprise** [*chassis*|*link-bypass*|*logger*|*operational-status*|*RDR-formatter*|*sntp*|*system-reset*|*telnet*] and press **Enter**.

The SNMP server is enabled to send the specified enterprise trap(s).

EXAMPLE:

The following example shows how to configure the SNMP server to send the logger enterprise trap only.

```
SCE(config)#snmp-server enable traps enterprise logger
```

To restore all traps to the default status:

Step 1 At the ***SCE***(config)# prompt, type **default snmp-server enable traps**, and press **Enter**.

All traps supported by the ***SCE*** platform are reset to their default status.

EXAMPLE:

The following example shows how to restore all SNMP traps to their default status.

```
SCE(config)# default snmp-server enable traps
```

To configure the SCE to stop sending traps to an NMS:

Step 1 At the ***SCE***(config)# prompt, type **no snmp-server host** *IP-address*, and press **Enter**.

The ***SCE***(config)# prompt appears.

EXAMPLE:

The following example shows how to remove the host with the IP Address: “192.168.0.83”.

```
SCE(config)#no snmp-server host 192.168.0.83
```

CLI

The ***SCE*** platform supports the CLI commands that control the operation of the SNMP agent. All the SNMP commands are available in Admin authorization level. The SNMP agent is disabled by default and any SNMP configuration command enables the SNMP agent (except where there is an explicit disable command).

Privileged Exec Mode Commands

The following SNMP commands are available in Exec mode when the SNMP agent is enabled:

- `show snmp (also available when SNMP agent is disabled)`
- `show snmp community`
- `show snmp contact`
- `show snmp enabled`
- `show snmp host`
- `show snmp location`
- `show snmp mib`
- `show snmp traps`

Global Configuration Mode Commands

The following SNMP commands are available in Global Configuration Mode:

- `snmp-server enable`
- `no snmp-server`
- `snmp-server community`
- `no snmp-server community all`
- `[no | default] snmp-server enable traps`
- `[no] snmp-server host`
- `no snmp-server host all`
- `[no] snmp-server contact`
- `[no] snmp-server location`

MIBs

MIBs (Management Information Bases) are databases of objects that can be monitored by a network management system (NMS). SNMP uses standardized MIB formats that allow any SNMP tools to monitor any device defined by a MIB.

The ***SCE*** `platform` supports the following MIBs:

- MIB-II as defined in RFC 1213, Management Information Base for Network Management of TCP/IP-based Internets.
- Cisco enterprise MIB, which is described by a number of MIB files. *Proprietary MIB Reference* (on page A-1).)

MIB-II

SCE platform fully supports MIB-II (RFC1213), including the following groups:

- System
- Interface (for both the management and line ports)
- AT (management port)
- IP (management port)
- ICMP (management port)
- TCP (management port)
- UDP (management port)
- SNMP (management port)

Service Control Enterprise MIB

The SCE proprietary MIB enables external management systems to retrieve general information regarding the ***SCE*** platform operating status and resources utilization, extract real time measurements of bandwidth utilization and network statistics, and receive notifications of critical events and alarms.

Note The following object identifier represents the Service Control Enterprise MIB: *1.3.6.1.4.1.5655*, or *iso.org.dod.internet.private.enterprise.pcube*

The Service Control Enterprise MIB splits into four main groups: Products, Modules, Management, and Workgroup. The Service Control enterprise tree structure is defined in a MIB file named `pcube.mib`.

Refer to the *Proprietary MIB Reference* (on page A-1) for a complete description of the `pcube` enterprise MIB.

- The `pcubeProducts` sub-tree contains the sysObjectIDs of Cisco products.

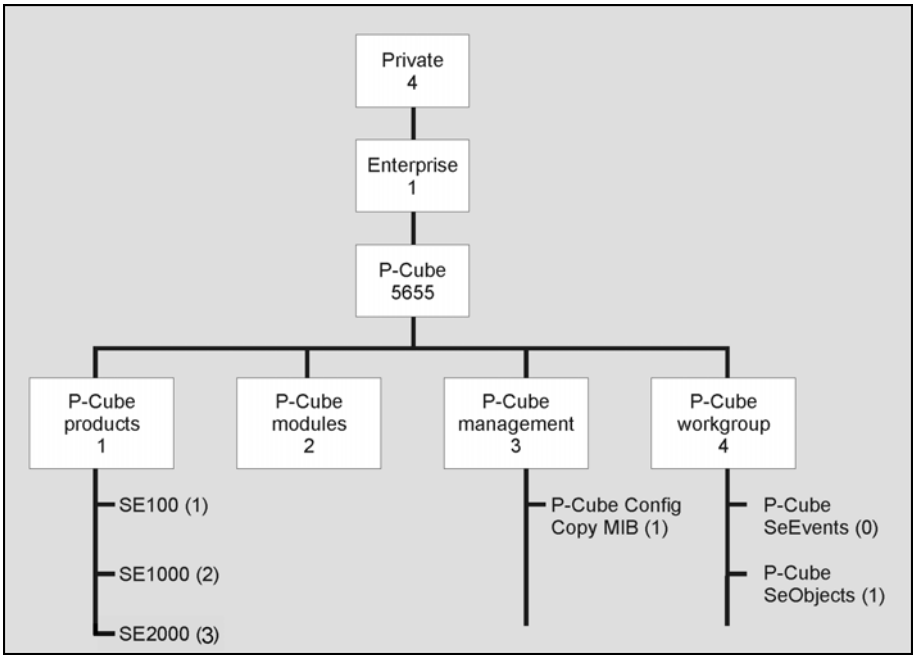
  Cisco product sysObjectIDs are defined in a MIB file named `Pcube-Products-MIB`
- The `pcubeModules` sub-tree provides a root object identifier from which MIB modules can be defined.
- The `pcubeMgmt` sub-tree contains the configuration copy MIB.
- The `pcubeWorkgroup` sub-tree contains the SCE MIB, which is the main SNMP MIB for the Cisco OS products.

  The SCE MIB is divided into two main groups:

  - **pcubeSeEvents**
  - **pcubeSEObjs**

The figure below, illustrates the Service Control Enterprise MIB structure.

Figure 5-1: Service Control MIB Structure



Loading the MIB Files

The SCE proprietary MIB uses definitions that are defined in other MIBs, such as Pcube MIB (pcube.mib), and the SNMPv2-SMI. Therefore, the order in which the MIBs are loaded is important. To avoid errors, the MIBs must be loaded in the proper order.

To load the MIBs:

Step 1 Load the SNMPv2-SMI.

Step 2 Load the SNMPv2-TC.

Step 3 Load pcube.mib.

Step 4 Load pcubeSEMib.mib.

Passwords

Cisco CLI passwords are an access-level authorization setting, not individual user passwords. All Admin users, for example, log in with the same password. This means that the system does not identify you as an individual, but as a user with certain privileges.

Passwords are needed for all authorization levels in order to prevent unauthorized users from accessing the ***SCE*** platform. It is highly recommended that you change the default password upon initial installation, and that you change the passwords periodically to secure the system.

Note The default password for all levels is either "**pcube**" or "**cisco**".

When a telnet user logs on, he sees only a Password: prompt, no logo is displayed. This provides extra security by not revealing the system identity to users that do not know the password.

Password guidelines:

- Password length must be between 4 and 100 characters long.
- Passwords can contain any visible keyboard character.
- Passwords must begin with a letter.
- Passwords cannot contain spaces.
- Passwords are case-sensitive.

Users with Admin or higher authorization level can view the configured passwords using the show running-config or the show startup-config commands. Therefore, if you want passwords to remain completely confidential, you must activate the encryption feature, described in *Encryption* (on page 5-22)

Changing Passwords

Use the **`enable password`** command to change the password. Note that if the password has been changed, the default password will no longer be accepted.

To change the password for a specified level:

Step 1 At the ***SCE***> prompt, to access the Admin authorization level, type **`enable`** and press **Enter**.

The `Password:` prompt appears.

Step 2 Type **`cisco`** (the default password for the Admin level) and press **Enter**.

The ***SCE***# prompt appears.

Step 3 To enter the Global Configuration Mode, type **`configure`** and press **Enter**.

The ***SCE***`(config)`# prompt appears.

Step 4 Type **`enable password level <level> <password>`**, and press **Enter**.

Use the appropriate value for the *level* parameter as follows:

- 0: user
- 10: admin
- 15: root

Your new password for the specified level is entered into the system.

The ***SCE***(config)# prompt appears.

Step 5 Type **exit** to exit the Global Configuration Mode and press **Enter**.

The ***SCE***# prompt appears.

Step 6 At this point, the Network Administrator should record passwords in a secure location.

To verify that you configured your passwords correctly:

Step 1 Initiate a new telnet connection, while maintaining the one you used to set the password.

This is needed so that if the verification fails, you would still have admin level authorization in order to re-enter the password.

Step 2 At the ***SCE***# prompt, do one of the following, according to the password level you are checking:

- Type **enable**.

  OR

- Type **enable 15**. (Root level)

Step 3 Press **Enter**.

Step 4 Type your new password and press **Enter**.

If your new password has been entered successfully, then the ***SCE*** Admin or Root prompt appears.

If you enter an incorrect password , the following error message appears: "Error—The supplied password is simply not right."

Step 5 Repeat steps 1 to 3 to check additional passwords.

The encryption feature will encrypt the passwords in the platform configuration files.

Encryption

Once the encryption feature is activated, passwords entered into the system are encrypted to the startup configuration file the next time the configuration is saved. When encryption feature is turned off, passwords previously encrypted to the startup configuration file are not deciphered.

By default, the password encryption feature is turned off.

To enable password encryption:

Step 1 From the ***SCE***`(config)#` prompt, type **`service password encryption`**.

Password encryption is enabled.

To disable password encryption:

Step 1 From the ***SCE***`(config)#` prompt, type **`no service password encryption`**.

This does not remove the encryption from the configuration file. You must save to the startup configuration file if you want the password to be stored un-encrypted on the startup configuration file.

Note Once the system is secured, you cannot recover a lost or forgotten password. Contact your Cisco customer support center if the password is lost.

IP Configuration

IP Routing Table

For handling IP packets on the out of band FE port, the ***SCE*** platform maintains a static routing table. When a packet is sent, the system checks the routing table for proper routing, and forwards the packet accordingly. In cases where the ***SCE*** platform cannot determine where to route a packet, it sends the packet to the default gateway.

SCE platform supports the configuration of the default gateway as the default next hop router, as well as the configuration of the routing table to provide different next hop routers for different subnets (for maximum configuration of 10 subnets).

The following sections illustrate how to use CLI commands to configure various parameters.

The following commands are relevant to IP Routing tables:

- `ip route prefix mask next-hop`
- `no ip route all`

- `no ip route prefix mask`
- `show ip route`
- `show ip route prefix`
- `show ip route prefix mask`

Default Gateway

To configure the default gateway:

Step 1 From the ***SCE***`(config)#` prompt, type **`ip default-gateway <address>`**, and press **Enter**.

The default gateway for the ***SCE*** platform is set.

EXAMPLE:

The following example shows how to set the default gateway IP of the ***SCE*** platform to 10.1.1.1.

```
SCE(config)#ip default-gateway 10.1.1.1
```

Adding IP Routing Entry to Routing Table

To add an IP routing entry to the routing table:

Step 1 From the ***SCE***`(config)#` prompt, use the **`ip route <prefix> <mask> <next-hop>`** command, and press **Enter**.

The IP routing entry is added to the routing table. (All addresses must be in dotted notation. The next-hop must be within the Fast-Ethernet interface subnet.)

EXAMPLE:

The following example shows how to set the router 10.1.1.250 as the next hop to subnet 10.2.0.0.

```
SCE(config)#ip route 10.2.0.0 255.255.0.0 10.1.1.250
```

Show IP Route

To use show ip route command to display the entire routing table:

Step 1 From the ***SCE***# prompt, type **`show ip route`** and press **Enter**.

The entire routing table and the destination of last resort (default-gateway) appear.

EXAMPLE:

```
SCE#show ip route
gateway of last resort is        10.1.1.1

|    prefix       |     mask          |    next hop     |
|-----------------|-------------------|-----------------|
|        10.2.0.0 |     255.255.0.0   |      10.1.1.250 |
|        10.3.0.0 |     255.255.0.0   |      10.1.1.253 |
|        198.0.0.0 |       255.0.0.0 |      10.1.1.251 |
|        10.1.60.0 |   255.255.255.0 |      10.1.1.5   |
```

To use show ip route prefix command to display routing entries from the subnet specified by the prefix and mask pair:

Step 1 From the ***SCE***# prompt, type show **`ip route prefix mask`** and press **Enter**.

Routing entries with this prefix and mask pair appear.

EXAMPLE:

```
SCE#show ip route 10.1.60.0 255.255.255.0
|    prefix       |     mask          |    next hop     |
|-----------------|-------------------|-----------------|
|       10.1.60.0 |   255.255.255.0 |        10.1.1.5 |
SCE#
```

IP Advertising

IP advertising is the act of periodically sending Ping requests to a configured address at configured intervals. This maintains the ***SCE*** platform IP/MAC addresses in the memory of adaptive network elements, such as switches, even during a long period of inactivity.

The following commands are relevant to IP advertising:

- `[no] ip advertising`
- `ip advertising destination`
- `ip advertising interval`
- `default ip advertising destination`
- `default ip advertising interval`
- `show ip advertising`
- `show ip advertising destination`
- `show ip advertising interval`

Configuring IP Advertising

In order to configure IP advertising, you must first enable IP advertising. You may then specify a destination address to which the ping request is to be sent and/or the frequency of the ping requests (interval). If no destination or interval is explicitly configured, the default values are assumed.

To enable IP advertising:

Step 1 From the ***SCE***(config)# prompt, type **ip advertising**, and press **Enter**.

IP advertising is enabled.

To configure the IP advertising destination:

Step 1 From the ***SCE***(config)# prompt, type **ip advertising destination <destination>**, and press **Enter**.

The specified IP address is the destination for the ping requests.

To configure the IP advertising interval in seconds:

Step 1 From the ***SCE***(config)# prompt, type **ip advertising interval <interval>**, and press **Enter**.

The ping requests are sent at the specified intervals.

EXAMPLE:

The following example shows how to configure IP advertising, specifying 10.1.1.1 as the destination and an interval of 240 seconds.

```
SCE(config)#ip advertising destination 10.1.1.1 interval 240
```

Show IP Advertising

To display the current IP advertising configuration:

Step 1 From the ***SCE***# prompt, type **`show ip advertising`** and press **Enter**.

The status of IP advertising (enabled or disabled), the configured destination, and the configured interval are displayed.

Setting the IP Address and Subnet Mask of the FastEthernet Management Interface

Warning Changing the IP address of the management interface via telnet will result in loss of the telnet connection and inability to reconnect with the interface.

To set the IP address and subnet mask of the FastEthernet Management Interface:

Step 1 From the ***SCE***`(config if)#` prompt, type **`ip address`** new-*`address subnet-mask`* and press **Enter**.

The command might fail if there is a routing table entry that is not part of the new subnet, defined by the new IP address and subnet mask.

EXAMPLE:

The following example shows how to set the IP address of the ***SCE*** platform to 10.1.1.1 and the subnet mask to 255.255.0.0.

```
SCE(config if)#ip address 10.1.1.1 255.255.0.0
```

Time Clocks and Time Zone

The ***SCE*** platform has three types of time settings, which can be configured: the clock, the calendar, and the time zone. It is important to synchronize the clock and calendar to the local time, and to set the time zone properly. The ***SCE*** platform does not track Daylight Saving Time automatically, so you must update the time zone when the time changes bi-annually.

The ***SCE*** platform has the following two time sources:

- A real-time clock, called the calendar, that continuously keeps track of the time, even when the ***SCE*** platform is not powered up. When the ***SCE*** platform reboots, the calendar time is used to set the system clock. The calendar is not used for time tracking during system operation.

- A system clock, which creates all the time stamps during normal operation. This clock clears if the system shuts down. During a system boot, the clock is initialized to show the time indicated by the calendar.

It does not matter which clock you set first, as long as you use the clock and calendar read commands to ensure they are synchronized.

The time zone settings are important because they allow the system to communicate properly with other systems in other time zones. The system is configured based on Greenwich Mean Time (GMT), which is standard in the industry for coordination with other manufacturers' hardware and software. For example, Pacific Standard Time would be written as PST-10, meaning that the name of the time zone is PST, which is 10 hours behind Greenwich Mean Time.

When setting and showing the time, the time is always typed or displayed according to the local time zone configured.

Showing System Time

To display the current time of the system clock:

Step 1 From the ***SCE***`(config)#` prompt, type **`show clock`** and press **Enter**.

The time maintained by the system clock appears.

EXAMPLE:

The following example shows the current system clock.

```
SCE#show clock
12:50:03  UTC  MON  November 13 2001
```

Showing Calendar Time

To display the current time and date of the system calendar:

Step 1 From the ***SCE#*** prompt, type **`show calendar`** and press **Enter**.

The current system calendar appears.

EXAMPLE:

The following example shows the current system calendar.

```
SCE#show calendar
12:50:03  UTC  MON  November 13 2001
```

Setting the Clock

To set the clock:

Step 1 From the ***SCE***# prompt, type **`clock set <hh:mm:ss day month year>`**, where *<hh:mm:ss day month year>* is the time and date you want to set, and press **Enter**.

EXAMPLE:

The following example shows how to set the clock to 20 minutes past 10 AM, October 13, 2001, updates the calendar and then displays the time.

```
SCE#clock set 10:20:00 13 oct 2001
SCE#clock update-calendar
SCE#show clock
10:21:10  UTC  THU  October  13  2001
```

Setting the Calendar

To set the calendar:

Step 1 From the ***SCE***# prompt, type calendar **`set <hh:mm:ss day month year>`**, where <hh:mm:ss day month year> is the time and date you want to set.

This sets the system calendar, displaying the time and date.

Step 2 Synchronize the clock with the calendar time you just set by typing **`clock read-calendar`**.

The time specified in this command is relative to the configured time zone.

EXAMPLE:

The following example shows that the calendar is set to 20 minutes past 10 AM, October 13, 2001.

```
SCE#calendar set 10:20:00 13 oct 2001
SCE#clock read-calendar
SCE#show calendar
10:20:00  UTC  THU  October  13  2001
```

Setting the Time Zone

To set the current time zone:

Step 1 From the ***SCE***`(config)`# prompt, type **`clock timezone <zone> <hours>`**, where <zone> is the name of the time zone and <hours> is the offset from GMT.

EXAMPLE:

The following example shows how to set the time zone to Pacific Standard Time with an offset of 10 hours behind GMT.

```
SCE(config)#clock timezone PST –10
```

Note You can configure time zones that do not differ from GMT by a multiple of one hour. Consult the CLI Command Reference regarding the clock timezone global configuration command.

Removing Current Time Zone Setting

To remove the current time zone setting:

Step 1 From the ***SCE***`(config)#` prompt, type **`no clock timezone`** and press **Enter**.

The default time zone is UTC (GMT).

EXAMPLE:

The following example shows how to remove the time zone setting.

```
SCE(config)#no clock timezone
```

Configuring Daylight Saving Time

The ***SCE*** platform can be configured to automatically switch to daylight savings time on a specified date, and also to switch back to standard time. In addition, the three-letter time zone code can be configured to vary with daylight savings time if required. (For instance, in the eastern United States, standard time is designated EST, and daylight savings time is designated EDT).

The transition times into and out of daylight savings time may be configured in one of two ways, depending on how the dates for the beginning and end of daylight savings time are determined for the particular location:

- recurring: If daylight savings time always begins and ends on the same day every year, (as in the United States), the **`clock summer-time recurring`** command is used. The beginning and ending days for daylight savings time can be configured once, and the system will automatically perform the switch every year.
- not recurring: If the start and end of daylight savings time is different every year, (as in Israel), the **`clock summer-time`** command is used. In this case, the transitions must be configured every year for that particular year. (Note that "year" is not necessarily a calendar year. If the transition days are determined in the fall, the transitions for that fall and the next spring may be configured.)

The day on which the transition takes place may be defined in several ways:

- Specific date: For example: March 29, 2004. A specific date, including the year, is defined for a not recurring configuration.

- First/last occurrence of a day of the week in a specified month: For example: the last Sunday in March. This is used for a recurring configuration.
- Day of the week in a specific week in a specified month: For example: Sunday of the fourth week of March. (This would be different from the last Sunday of the month whenever there were five Sundays in the month). This is used for a recurring configuration.

General guidelines for configuring daylight savings time transitions:

- Specify the three letter time zone code for daylight savings time.
- recurring: specify a day of the month (week#|first|last/day of the week/month).
- not recurring: specify a date (month/day of the month/year).
- Define two days:
  - Day1 = beginning of daylight savings time.
  - Day2 = end of daylight savings time.

  In the Southern hemisphere, month2 must be before month1, as daylight savings time begins in the fall and ends in the spring.
- Specify the exact time that the transition should occur (24 hour clock).
  - Time of transition into daylight savings time: according to local standard time.
  - Time of transition out of **`clock summer-time recurring`** : according to local daylight savings time.
- Offset: specify the difference in minutes between standard time and daylight savings time.

  Default = 60 minutes
- For the **`clock summer-time recurring`** command, the default values are the United States transition rules:
  - Daylight savings time begins: 2:00 (AM) on the first Sunday of April.
  - Daylight savings time ends: 2:00 (AM) on the last Sunday of October.

To define recurring daylight savings time transitions:

Step 1 From the ***SCE***`(config)#` prompt, type **`clock summer-time`** *`<zone>`* **`recurring`** [*`<week1> <day1> <month1> <time1> <week2> <day2> <month2> <time2> [<offset>]`*] and press **Enter**.

EXAMPLE:

The following example shows how to configure recurring daylight savings time for a time zone designated "DST" as follows:

- Daylight savings time begins: 0:00 on the last Sunday of March.
- Daylight savings time ends: 23:59 (AM) on the Saturday of fourth week of November.
- Offset = 1 hour (default)

```
SCE(config)# clock summer-time DST recurring last Sunday March 00:00   4
Saturday November 23:59
```

To define non-recurring daylight savings time transitions:

Step 1 From the ***SCE***(config)# prompt, type **clock summer-time** *<zone>* [*<date1>* *<month1>* *<year1>* *<time1>* *<date2>* *<month2>* *<year2>* *<time2>* *[<offset>]*] and press **Enter**.

EXAMPLE:

The following example shows how to configure non-recurring daylight savings time for a time zone designated "DST" as follows:

- Daylight savings time begins: 0:00 on April 16, 2004.
- Daylight savings time ends: 23:59 October 23, 2004.
- Offset = 1 hour (default)

```
SCE(config)# clock summer-time DST April 16 2004 00:00 October 23 2004 23:59
```

To cancel the daylight savings time transitions configuration:

Step 1 From the ***SCE***(config)# prompt, type **no clock summer-time** and press **Enter**.

To display the current daylight savings time configuration:

Step 1 From the ***SCE***(config)# prompt, type **show timezone** and press **Enter**.

The current time zone and daylight saving time configuration is displayed.

SNTP

The Simple Network Timing Protocol (SNTP) is a simple solution to the problem of synchronizing the clocks in the various elements of the network. SNTP provides access to a time source via the network. The system clock and calendar are then set in accordance with this external source.

There are two options for the SNTP client. These functions are independent, and the system employ either one or both.

- Multicast SNTP client: Listens to SNTP broadcasts and updates the system clock accordingly.
- Unicast SNTP client: Sends a periodic request to a configured SNTP server, and updates the system clock according to the server response.

Note It is recommended that an IP access control list be configured in order to prevent access from unauthorized SNTP or NTP multicast servers.

The following commands are relevant to SNTP configuration:

- `[no] sntp broadcast client`
- `[no] sntp server address`
- `no sntp server all`
- `sntp update-interval interval in seconds`
- `show sntp`

Enabling SNTP multicast client

To enable the SNTP multicast client:

Step 1 From the ***SCE***`(config)#` prompt, type **`sntp broadcast client`**, and press **Enter**.

The SNTP multicast is enabled, and will accept time updates from any broadcast server.

Disabling SNTP multicast client

To disable the SNTP multicast client:

Step 1 From the ***SCE***`(config)#` prompt, type **`no sntp broadcast client`**, and press **Enter**.

The SNTP multicast client is disabled, and will not accept any broadcast time updates.

Enabling SNTP unicast client

To define the SNTP unicast server to be queried:

Step 1 From the ***SCE***`(config)#` prompt, type **`sntp server <address>`**, and press **Enter**, where <address> is the IP address of the SNTP server.

The SNTP unicast server is defined, and SNTP client is enabled to query that server.

EXAMPLE:

The following example shows how to enable an SNTP server at IP address 128.182.58.100.

```
SCE(config)# sntp server 128.182.58.100
```

Disabling SNTP unicast client

To disable the SNTP unicast client and remove all servers from the client list:

Step 1 From the ***SCE***`(config)#` prompt, type **`no sntp server all`**, and press Enter.

All SNTP unicast servers are removed, preventing unicast SNTP query.

To remove one SNTP servers from the client list:

Step 1 From the ***SCE***`(config)#` prompt, type **`no sntp server <address>`**, and press **Enter**, where <address> is the IP address of the SNTP server.

The specified SNTP unicast server is removed.

Defining the SNTP unicast update interval

To define the interval for SNTP update queries:

Step 1 From the ***SCE***`(config)#` prompt, type **`sntp update-interval <interval>`**, where <interval> is the time in seconds between updates (64 through 1024), and press **Enter**.

The SNTP unicast client will query the server at the defined intervals.

EXAMPLE:

The following example shows how to set the SNTP update interval for 100 seconds.

```
SCE(config)# sntp update-interval 100
```

Display SNTP information

To get information about SNTP servers and updates:

Step 1 From the ***SCE***`(config)#` prompt, type **`show sntp`**, and press **Enter**.

The configuration of both the SNTP unicast client and the SNTP multicast client is displayed.

EXAMPLE:

```
SNTP broadcast client: disabled
last update time: not available

SNTP unicast client: enabled
SNTP unicast server: 128.182.58.100
last update time: Feb 10 2002, 14:06:41
update interval: 100 seconds
```

Domain Name (DNS) Settings

When a name of a host is given as a parameter to a CLI command that expects a host name or an IP address, the system translates the name to an IP address according to the following:

Step 1 If the name is in a dotted decimal notation (that is, in the format x.x.x.x), it is directly translated to an IP address it represents.

Step 2 If the name does not contain the dot character (.), the system looks it up in the IP Host table. If the name is found on the table, it is mapped to the corresponding IP address. The IP host table can be configured using the command `ip host`.

Step 3 If the name does not contain the dot (.) character, and the domain name function is enabled (See the ip domain-lookup command), and a default domain name is specified (See the ip domain-name command), the default domain name is appended to the given name to form a fully qualified host name. This, in turn, is used to perform a DNS query translating the name to an IP address.

Step 4 Otherwise, if the domain name function is enabled, the name is considered to be fully qualified, and is used to perform a DNS query translating the name to an IP address.

The following commands are relevant to DNS settings:

- `ip name-server`
- `ip domain-name`
- `no ip domain-name`
- `ip domain-lookup`
- `show hosts`

To enable DNS lookup:

Step 1 From the ***SCE***(config)# prompt, type **ip domain-lookup**.

To disable DNS lookup:

Step 1 From the ***SCE***(config)# prompt, type **no ip domain-lookup**.

Name Servers

To specify the address of one or more name servers to use for name and address resolution:

Step 1 From the ***SCE***(config)# prompt, type **ip name-server <server-address1> [<server-address2> [<server-address3>]]**, and press **Enter**.

The addresses of the name servers are set.

EXAMPLE:

The following example shows how to configure the two name server (DNS) IP addresses.

```
SCE(config)#ip name-server 10.1.1.60 10.1.1.61
```

To remove the name server address:

Step 1 From the ***SCE***(config)# prompt, type **no ip name-server <server-address1> [<server-address2> [<server-address3>]]**, and press **Enter**.

The addresses of the name servers are removed.

EXAMPLE:

The following example shows how to remove the name server (DNS) IP address.

```
SCE(config)#no ip name-server 10.1.1.60 10.1.1.61
```

To clear the name server table all addresses :

Step 1 From the ***SCE***(config)# prompt, type **no ip name-server**, and press **Enter**.

Domain Name

To define a default domain name:

Step 1 From the ***SCE***(config)# prompt, type **ip domain-name domain-name**, and press **Enter**.

The default domain name is defined. The default domain name is used to complete unqualified host names. Do not include the initial period that separates an unqualified name from the domain name.

EXAMPLE:

The following example shows how to configure the domain name.

Now, if the hostname “Cisco” is entered, the default domain name “com” is appended, to produce “Cisco.com”.

```
SCE(config)#ip domain-name com
```

EXAMPLE:

The following example shows how to remove the configured domain name.

```
SCE(config)#no ip domain-name
```

Host Table

To add a hostname and address to the host table:

Step 1 From the ***SCE***(config)# prompt, type **ip host hostname ip-address**, and press **Enter**.

EXAMPLE:

The following example shows how to add a host to the host table.

```
SCE(config)#ip host PC85 10.1.1.61
```

EXAMPLE:

The following example shows how to remove a hostname together with all of its IP mappings.

```
SCE(config)#no ip host PC85
```

show hosts

To display current DNS settings:

Step 1 From the ***SCE#*** prompt, type **`show hosts`**.

EXAMPLE:

The following example shows how to display current DNS information.

```
SCE#show hosts
Default domain is Cisco.com
Name/address lookup uses domain service
Name servers are 10.1.1.60, 10.1.1.61
Host                 Address
----                 -------
PC85                 10.1.1.61
SCE#
```

Management Interface Configuration Mode

This interface has a transmission rate of 100 Mbps and is used for management operations and for transmitting RDRs, which are the output of traffic analysis and management operations. The parameters that can be configured for this interface include:

- IP address of the interface, see *Setting the IP Address and Subnet Mask of the FastEthernet Management Interface* (on page 5-26).
- Speed and duplex, see *Configuring the Speed of the FastEthernet Interface* (on page 5-38) and *Configuring the Duplex Operation of the FastEthernet Interface* (on page 5-37).

Configuring the Management Interface Speed and Duplex Parameters

This section presents sample procedures that describe how to configure the speed and the duplex of the Management Interface.

Configuring the Duplex Operation of the FastEthernet Interface

To configure the duplex operation of the FastEthernet Management Interface:

Step 1 From the ***SCE*****`(config if)#`** prompt, type **`duplex [auto | full |half]`** **and** press Enter.

Configures the duplex operation of the FastEthernet Management Interface to either half duplex, or full duplex. **auto** means auto-negotiation (do not force duplex on the link).

The default of this command is set to **auto**. Changing this configuration takes effect only if the **speed** is not configured to **auto**.

EXAMPLE:

The following example shows how to use this command to configure the FastEthernet Management port to half duplex mode.

```
SCE(config if)#duplex half
```

Configuring the Speed of the FastEthernet Interface

To configure the speed of the FastEthernet Management Interface:

Step 1 From the ***SCE*(config if)#** prompt, type **speed** *speed*, where *speed* can be **10**, **100** (Mbps) or **auto** and press **Enter**.

Configures the speed of the FastEthernet Management Interface to either 10 Mbps or 100 Mbps. **auto** means auto-negotiation (do not force speed on the link).

The default of this command is set to **auto**. Changing this configuration takes effect only if the **duplex** mode is not configured to **auto**.

EXAMPLE:

The following example shows how to use this command to configure the FastEthernet Management port to 100 Mbps speed.

```
SCE(config if)#speed 100
```

Table 5-3 Interface State Relationship to Speed and Duplex

Speed	Duplex	Actual FEI state
Auto	Auto	Auto negotiation
Auto	Full	Auto negotiation
Auto	Half	Auto negotiation
10	Auto	Auto-negotiation (duplex only)
10	Full	10 Mbps and Full duplex
10	Half	10 Mbps and half duplex
100	Auto	Auto-negotiation (speed only)
100	Full	100 Mbps and full duplex
100	Half	100 Mbps and half duplex

Configuring the Line Interface

This chapter contains the following sections:

- Configuring Tunneling Protocols 6-1
- Configuring Traffic Rules and Counters 6-4
- Configuring TOS Marking 6-11
- Line Ethernet Interfaces 6-12

Configuring Tunneling Protocols

Tunneling technology is used across various telecommunications segments in order to solve a wide variety of networking problems. The ***SCE*** platform is designed to recognize various tunneling protocols. When the ***SCE*** platform is installed in an L2TP, MPLS or VLAN environment, it is able to ignore the tunnel headers and skip into the higher L3 layer for protocol classification

A tunneling protocol adds headers to the basic protocol stack in order to route the packet across the telecommunications segments. Therefore, the system must be aware that the packets contain additional tunnel headers. Based on the selected protocol, the system skips the tunnel (outer IP headers and tunnel headers) and processes only to the internal IP header and the data.

Since VLAN and MPLS constitute headers at layers just above L2 and below any L3 layer, they are automatically recognized as tunnels regardless of the system configuration, with the exception that MPLS label stacks have a maximum depth of 15 labels.

In addition to skipping the tunnel VLAN and MPLS tunnel headers, the ***SCE*** platform is also able to differentiate flows and differentiate subscribers (i.e. use the VLAN information for classification purposes) according to the VLAN tag when configured in the correct mode. VLAN classification is possible only for symmetric tunnels, that is, when the VLAN tags of every tunnel are identical for both the upstream and downstream direction (of that tunnel).

The default system mode is the following:

- Skip VLAN headers, do-not use them for classification.

  The VLAN environment is assumed to be symmetric.

- Skip MPLS headers.

  The MPLS environment is assumed to be Traffic-Engineering.

- No IP-tunnel support – L2TP tunnels will not be skipped and therefore all flows within a single L2TP tunnel will be classified as the same flow.

Selecting the Tunneling Mode

Use these commands to configure tunneling:

- `ip tunnel`
- `vlan`
- `mpls`
- `L2PT identify-by`

Configuring IP Tunnels

By default, IP tunnel recognition is disabled. Use this command to configure recognition of L2TP tunnels and skipping into the internal IP packet.

An IP tunnel is mutually exclusive with using VLAN for classification.

To configure IP tunnels:

Step 1 From the ***SCE***(config if)# prompt, type:

`ip tunnel L2TP skip` and press **Enter**.

To disable identification of IP tunnels:

Step 1 From the ***SCE***(config if)# prompt, type:

`no ip tunnel` and press **Enter**.

Configuring the VLAN Environment

Use this command to configure the VLAN environment. There are three options:

- symmetric classify
- symmetric skip (default)
- a-symmetric skip

Setting the mode to classify means that subscriber and flow classification will use the VLAN tag. Using VLAN classification is mutually exclusive with any IP tunnels.

Note that using The *a-symmetric skip* value incurs a performance penalty.

To configure the VLAN environment

Step 1 From the ***SCE***(config if)# prompt, type:

`vlan [symmetric {classify|skip}] [a-symmetric skip]` and press **Enter.**

EXAMPLE:

The following example selects *symmetric skip* VLAN tunnel environment.

```
SCE(config if)#vlan symmetric skip
```

Configuring the MPLS Environment

Use this command to set the MPLS environment. Use the *VPN* keyword when the labels are mandatory in the traffic, otherwise use *Traffic-Engineering* (default).

Note that using the *VPN* value incurs a performance penalty.

To configure the MPLS environment

Step 1 From the ***SCE***(config if)# prompt, type:

`mpls [vpn|Traffic-Engineering] skip` and press **Enter**.

EXAMPLE:

The following example selects the VPN MPLS tunnel environment.

```
SCE(config if)#mpls vpn skip
```

Configuring the L2TP Environment

Use this command to set the port number that the LNS and LAC use for L2TP tunnels. The default port number is 1701.

To configure the L2TP port number

Step 1 From the ***SCE***(config if)# prompt, type:

`L2TP identify-by port-number` *<number>* and press **Enter.**

Displaying Tunneling Configuration

You can display the tunnel configuration.

To display the tunneling configuration:

Step 1 From the ***SCE***# prompt, type:

`show interface lineCard 0 [MPLS|VLAN|L2TP|IP-tunnel]` and press **Enter**.

Configuring Traffic Rules and Counters

Traffic rules and counters may be configured by the user. This functionality enables the user to define specific operations on the traffic flowing through the SCE Platform, such as blocking or ignoring certain flows or counting certain packets. The configuration of traffic rules and counters is independent of the application loaded by the ***SCE*** platform, and thus is preserved when the application being run by the ***SCE*** platform is changed.

Possible uses for traffic rules and counters include:

- Enabling the user to count packets according to various criteria. Since the traffic counters are readable via the SCE SNMP MIB, these might be used to monitor up to 32 types of packets, according to the requirements of the installation.
- Ignoring certain types of flows. When a traffic rules specifies an “ignore” action, packets matching the rule criteria will not open a new flow, but will pass through the ***SCE*** platform without being processed. This is useful when a particular type of traffic should be ignored by the ***SCE*** platform.

  Possible examples include ignoring traffic from a certain IP range known to require no service, or traffic from a certain protocol.
- Blocking certain types of flows. When a traffic rules specifies a “block” action, packets matching the rule criteria (and not belonging to an existing flow) will be dropped and not passed to the other interface. This is useful when a particular type of traffic should be blocked by the ***SCE*** platform.

  Possible examples include performing ingress source address filtering (dropping packets originating from a subscriber port whose IP address does not belong to any defined subscriber-side subnet), or blocking specific ports.

It should be noted that using traffic rules and counters does not affect performance. It is possible to define the maximum number of both traffic rules and counters without causing any degradation in the ***SCE*** platform performance.

Traffic Rules

A traffic rule specifies that a defined action should be taken on packets processed by the SCE Platform that meet certain criteria. The maximum number of rules is 128. Each rule is given a name when it is defined, which is then used when referring to the rule.

Packets are selected according to user-defined criteria, which may be any combination of the following:

- **IP address**: A single address or a subnet range can be specified for each of the line ports (Subscriber / Network).
- **Protocol**: TCP/UCP/ICMP/IGRP/EIGRP/IS-IS/OSPF/Other
- **TCP/UDP Ports**: A single port or a port range can be specified for each of the line ports (Subscriber / Network). Valid for the TCP/UDP protocols only.
- **TCP flags** (TCP only).
- **Direction** (Upstream/Downstream).

The possible actions are:

- **Count** the packet by a specific traffic counter
- **Block** the packet (do not pass it to the other side)
- **Ignore** the packet (do not provide service for this packet: No bandwidth metering, transaction reporting etc. is done)

Block and **Ignore** actions affect only packets that are not part of an existing flow.

Note that **Block** and **Ignore** are mutually exclusive. However, blocked or ignored packets can also be counted.

It is possible for a single packet to match more that one rule (The simplest way to cause this is to configure two identical rules with different names). When this happens, the system operates as follows:

- Any counter counts a specific packet only once. This means that:
  - If two rules specify that the packet should be counted by the same counter, it is counted only once.
  - If two rules specify that the packet should be counted by different counters, it is counted twice, once by each counter.
- **Block** takes precedence over **Ignore**: If one rule specifies **Block**, and another rule specifies **Ignore**, the packet is blocked.

Traffic counters

Traffic counters count the traffic as specified by the traffic rules. The maximum number of counters is 32. Each counter is given a name when it is defined, which is then used when referring to the counter.

A traffic counter can be configured in one of two ways:

- **Count packets**: the counter is incremented by 1 for each packet it counts.

- **Count bytes**: the counter is incremented by the number of bytes in the packet for each packet it counts.

Configuring Traffic Counters

A traffic counter must be created before it can be referenced in a traffic rule. Use the following commands to create and delete traffic counters.

To create a traffic counter:

Step 1 From the ***SCE***`(config if)#` prompt, type **`traffic-counter name`** *`<name> (count-bytes|count-packets)`*

To delete a traffic counter:

Step 1 From the ***SCE***`(config if)#` prompt, type **`no traffic-counter name`** *`<name>`*

Note that a traffic counter cannot be deleted if it is used by any existing traffic rule.

To delete all existing traffic counters:

Step 1 From the ***SCE***`(config if)#` prompt, type **`no traffic-counter all`**

Configuring Traffic Rules

Use the following commands to create and delete traffic rules.

To create a traffic rule:

Step 1 From the ***SCE***`(config if)#` prompt, type **`traffic-rule name`** *`<name>`* **`IP-addresses`** *`(all|(subscriber-side <IP specification> network-side <IP specification>))`* **`protocol`** *`<protocol>`* **`ports`** *`(all|(subscriber-side <port specification> network-side <port specification>))`* **`flags`** *`<flags specification>`* **`direction`** *`<direction>`* **`traffic-counter`** *`<traffic-counter>`* [**`action`** *`<action>`*]

Where the command options are defined as follows:

IP specification:

`all|([all-but] (<ip-address>|<ip-range>))`

- *<ip-address>* is a single IP address in dotted-decimal notation, such as 10.1.2.3
- *<ip-range>* is an IP subnet range, in the dotted-decimal notation followed by the number of significant bits, such as 10.1.2.0/24.
- Use the **all-but** keyword to exclude the specified IP address or range of IP addresses

protocol:

Any one of the following protocols:

TCP/UCP/ICMP/IGRP/EIGRP/IS-IS/OSPF/Other

port specification (TCP/UDP only):

all|([all-but] (<port>|<port-range>))

- *<port>* is a single port number (0-65535)
- *<port-range>* is a port range in the following notation: <min-port>:<max-port>, such as 80:82.
- Use the **all-but** keyword to exclude the specified port or range of ports

<flags specification> (TCP only):

Defines criteria for matching packets based on the TCP flag values.

all | (SYN (0|1|all) [FIN (0|1|all) [RST (0|1|all) [ACK (0|1|all) [URG (0|1|all) [PSH (0|1|all)]]]]]

For each flag, a value of 0, 1, or 'all' can be selected. Default is "all".

Note that flags are always processed in order, so that it is not possible to define a specific value for one flag without defining criteria for the preceding flags. So, for example, to specify ACK = 0 as one of the criteria, the preceding flags, SYN, FIN, and RST, must be set to **all**. The URG and PSH flags can be ignored, as they come after the ACk flag.

direction:

Any of the following:

upstream/downstream/all

traffic-counter:

Either of the following:

- *name <name of an existing traffic counter>:* Packets meeting the criteria of the rule are to be counted in the specified counter. If a counter name is defined, the "count" action is also defined implicitly. The keyword **name** must appear as well as the actual name of the counter.
- *none:* If **none** is specified, then an action must be explicitly defined via the **action** option.

action: (not required if the action is count only)

Either of the following:

ignore/block

EXAMPLE 1

This example creates the following traffic rule:

Name = rule1

IP addresses: subscriber side = all IP addresses, network side = 10.10.10.10 only

Protocol = other

Direction = all

Traffic counter = counter1

Since it is not TCP/UDP, port and flags are not applicable.

The only action performed will be counting

```
SCE (config if)# traffic-rule rule1 IP-addresses subscriber-side all
network-side 10.10.10.10 protocol other direction all traffic-counter name
counter1
```

EXAMPLE 2

This example creates the following traffic rule:

Name = rule2

IP addresses: subscriber side = all IP addresses, network side = all IP addresses EXCEPT the subnet 10.10.10.0/24

Protocol = TCP

Ports: subscriber side = 100, network side = 100-150

Flags = FIN flag when value = 1 (preceding flag (SYN) must be set to all)

Direction = downstream

Traffic counter = counter2

Action = Block

The actions performed will be counting and blocking

```
SCE (config if)# traffic-rule rule2 IP-addresses subscriber-side all
network-side all-but 10.10.10.0/24 protocol TCP ports subscriber-side 100
network-side 100:150 flags SYN all FIN 1 direction downstream traffic-
counter name counter2 action block
```

EXAMPLE 3

This example creates the following traffic rule:

Name = rule3

IP addresses: all

Protocol = IS-IS

Direction = upstream

Traffic counter = none

Action = ignore (required since traffic-counter = none)

Since it is not TCP/UDP, port and flags are not applicable.

The only action performed will be **Ignore**.

```
SCE (config if)# traffic-rule rule3 IP-addresses all protocol IS-IS
direction upstream traffic-counter none action ignore
```

To delete a traffic rule:

Step 1 From the ***SCE***`(config if)#` prompt, type **`no traffic-rule name`** *<name>*

Note that a traffic counter cannot be deleted if it is used by any existing traffic rule.

To delete all existing traffic rules:

Step 1 From the ***SCE***`(config if)#` prompt, type **`no traffic-rule all`**

Managing Traffic Rules and Counters

Use these commands to display existing traffic rule configuration, as well as traffic counter configuration (packets/bytes and the name of the rule using the counter) and traffic counter value. You can also reset a specific counter or all counters.

To view a specified traffic rule:

Step 1 From the ***SCE***# prompt, type **`show interface linecard 0 traffic-rule name`** *<rule-name>*

To view all existing traffic rules:

Step 1 From the ***SCE***# prompt, type **`show interface linecard 0 traffic-rule all`**

To view a specified traffic counter:

Step 1 From the ***SCE***# prompt, type **`show interface linecard 0 traffic-counter name`** *<counter-name>*

EXAMPLE

The following example displays information for the traffic counter "cnt".

```
SCE# show interface linecard 0 traffic-counter name cnt
Counter 'cnt' value: 0 packets. Rules using it: None.
```

To view all existing traffic counters:

Step 1 From the ***SCE***# prompt, type **`show interface linecard 0 traffic-counter all`**

EXAMPLE

The following example displays information for all existing traffic counters.

```
SCE#show interface linecard 0 traffic-counter all

Counter 'cnt' value: 0 packets. Rules using it: None.
1 counters listed out of 32 available.
```

To reset a specified traffic counter:

Step 1 From the ***SCE***# prompt, type `clear interface linecard 0 traffic-counter name <counter-name>`

To reset all existing traffic counters:

Step 1 From the ***SCE***# prompt, type `clear interface linecard 0 traffic-counter all`

Configuring TOS Marking

The ***SCE*** platform TOS marking feature enables marking the TOS field in the IP header of each packet according to two applicative attributes of the packet: its Class (class of service) and its Color (reflects the packet's level of compliance to its relevant bandwidth limitations, where applicable). The actual TOS value set in the IP header is determined according to the configurable TOS table, based on the Class and Color. The default values in the TOS table are based on the Diffserv standard.

Note The first few TCP packets (connection establishment) are associated and marked with a default AF4 class that is mapped to the IQ2 queue and *are marked accordingly*. This occurs because the ***SCE*** platform transmits the first few packets before classifying the flow and identifying the application or service.

The following commands are relevant to TOS marking:

- `no tos-marking diffserv`
- `tos-marking mode`
- `tos-marking set-table-entry class`
- `tos-marking reset-table`
- `show interface LineCard tos-marking mode`
- `show interface LineCard tos-marking table`

Enabling and Disabling TOS Marking

To enable TOS marking:

Step 1 From the ***SCE*** `platform(config if)#` prompt, type **`tos-marking mode diffserv`** and press **Enter**.

To disable TOS marking:

Step 1 From the ***SCE***`(config if)#` prompt, type **`no tos-marking diffserv`** and press **Enter**.

Modifying the TOS Table

To modify the TOS table:

Step 1 From the ***SCE***(config if)# prompt, type **tos-marking set-table-entry** *class* class **color** *color* **value** *value* and press Enter.

class is the applicative class of the packet (BE, AF1, AF2, AF3, AF4, EF),, *color* is the applicative color (green, red or any) and *value* is the value to be assigned to the packet (value set to the IP TOS field). The *value* parameter must be in hexadecimal format in the range **0x0** to **0x3f**.

EXAMPLE:

The following example sets a TOS marking table entry.

```
SCE (config if)#tos-marking set-table-entry class AF3 color green value 0x24
```

Line Ethernet Interfaces

The Ethernet Interfaces are used to connect the ***SCE***platform to the network. See the description of network topologies in Topology.

The four Ethernet Interfaces may be used either to connect to the data link, or to connect to another ***SCE*** platform. Refer to Connecting the line ports to the network for cabling diagrams

To configure the Ethernet parameters, you must be in the Ethernet Configure Interface Mode

Entering Ethernet Line Interface Configuration Mode

Configuring GigabitEthernet Auto-Negotiation

To configure GBE auto-negotiation for a specified GBE port interface, complete the following steps:

Step 1 To enter the Global Configuration Mode, at the ***SCE***# prompt, type **configure,** and press **Enter**.

The ***SCE***(config)# prompt appears.

Step 2 To enter the desired GBE port interface, type **interface GigabitEthernet 0**/*portnumber*, and press **Enter**, where *portnumber* is the number of the selected port (1-4).

The ***SCE***(config if)# prompt appears.

Step 3 Type **auto-negotiate** and press **Enter**.

The ***SCE***(config if)# prompt appears.

Step 4 To return to Global Configuration Mode, type **exit** and press **Enter**.

The ***SCE***(config)# prompt appears.

Repeat this procedure to configure auto-negotiation for the other GBE port interfaces as needed.

Configuring the Connection

This chapter contains the following sections:

- Editing the Connection Mode 7-1
- Link Mode 7-2
- Forced Failure 7-4
- Failure Recovery Mode 7-4
- SCE Platform/SM Connection 7-5
- Enabling and Disabling Link Failure Reflection 7-5

Editing the Connection Mode

The connection mode command allows you to configure the topology of the system in one command. The connection mode is determined by the physical installation of the ***SCE*** platform.

There are four topology-related parameters included in the connection mode command:

- **Connection mode**: Can be any one of the following, depending on the physical installation of the ***SCE*** platform:
  - Inline: single ***SCE*** platform inline
  - Receive-only: single ***SCE*** platform receive-only
  - Inline-cascade: two cascaded ***SCE*** platforms inline
  - Receive-only-cascade: two cascaded ***SCE*** platforms receive-only

  Default: **inline**
- **Physically-connected-links**: In cascaded topologies, defines which link is connected to this ***SCE*** platform. Possible values are ‘link-0’ and ‘link-1’.

  Not applicable to single ***SCE*** platform topologies.
- **Priority**: This parameter defines which is the primary ***SCE*** platform. It is applicable only in a two ***SCE*** platform topology. Possible values are ‘primary’ and ‘secondary’

  Not applicable to single ***SCE*** platform topologies.

- **On-failure**: This parameter determines whether the system cuts the traffic or bypasses it when the ***SCE*** platform either has failed or is booting.

  Default: **bypass**

  Not applicable to receive-only topologies.

Note Do not change the connection mode unless the physical installation has been changed.

To define the system topology:

Step 1 From the ***SCE*** `(config if)#` prompt, type **`connection-mode`** *`inline`*`|`*`receive-only`*`|`*`inline-cascade`*`|`*`receive-only-cascade`* **`physically-connected-links [`***`link 0`*`|`*`link 1`***`] priority [`***`primary`*`|`*`secondary`***`] on-failure [`***`bypass`*`|`*`cutoff`***`]`** and press **Enter**.

EXAMPLE 1:

The following example defines the primary device in a two-***SCE*** platform redundant, inline topology.Link 0 is connected to this device, and the link mode on failure is bypass

```
SCE (config if)# connection-mode inline-cascade physically-connected-links
link-0  priority primary on-failure bypass
```

EXAMPLE 2:

The following example defines a single-***SCE*** platform, dual link, receive-only topology. Neither link mode on failure, nor physically-connected-links, nor priority is applicable.

```
SCE (config if)# connection-mode receive-only
```

Link Mode

The ***SCE*** platform has an internal hardware card used to maintain the links even when the ***SCE*** platform fails. This hardware card has four possible modes of operation:

- bypass
- forwarding
- cutoff
- sniffing

Normally, the link mode is selected by the ***SCE*** platform software according to the configured connection-mode. However, the **`link-mode`** command can be used to enforce a specific desired mode. This may be useful when debugging the network, or in cases where we would like the ***SCE*** platform just to forward the traffic. (Note that this is only relevant to inline topologies even though the configuration is available also when in receive-only mode.)

The following link mode options are available:

- **Forwarding:** forwards traffic on the specified link to the ***SCE*** platform for processing.
- **Bypass:** stops all forwarding of traffic on the specified link to the ***SCE*** platform. Traffic still flows on the link, but is not processed in any way by the ***SCE*** platform.

  This does not affect the redundancy states.
- **Sniffing:** allows the ***SCE*** platform to forward traffic on the specified link through the bypass mechanism while still analyzing the traffic passively.

  Sniffing is permitted to be configured for for all links, only (use the all-links option).
- **Cutoff:** completely cuts off flow of traffic through the specified link.

Note the following recommendations and restrictions:

- Since the SCE 1000 platform has only one link, the link is not specified.
- Since the SCE 2000 platforms have more than one link, it is required to specify the link. The link designations are different for the GBE and FE platforms, as follows:
  - SCE 2000 4xGBE: GBE1-GBE2/GBE3-GBE4
  - SCE 2000 4/8xFE: LINK1/LINK2
- Use the **`'all-links'`** option to configure the link mode for all links (SCE 2000 platforms only).
- It is recommended that both links be configured together. Use the all-links option.
- Link mode is relevant only to inline topologies.
- It is recommended that in cascaded topologies, both ***SCE*** platforms be configured for the same link mode, otherwise the service will be unpredictable.
- Sniffing can only be configured for all links, therefore, to configure sniffing, the all-links option is required, not just recommended.
- The default link mode is forwarding. When other link modes are selected, active service control is not available and any service control configuration will not be applicable.

To set the link mode:

Step 1 From the ***SCE*** `(config if)#` prompt, type **`link-mode [`**<*`link`*>`|`*`all-links`*`]`** **`[`***`forwarding`*`|`*`bypass`*`|`*`sniffing`*`|`*`cutoff`*`]`** and press **Enter**.

Forced Failure

Use the following commands to force a virtual failure condition, and to exit from the failure condition when performing an application upgrade. (See *Application Upgrade* (on page 10-12).)

To force a virtual failure condition:

Step 1 From the ***SCE*** `(config if)#` prompt, type **`force failure-condition`** and press **Enter**.

To exit the virtual failure condition:

Step 1 From the ***SCE*** `(config if)#` prompt, type **`no force failure-condition`** and press **Enter**.

Failure Recovery Mode

The **`failure-recovery operation-mode`** command defines the behavior of the system after boot resulting from failure. The system may return to operational mode, or remain not operational.

The default value is **operational**.

- [no|default] failure-recovery operation-mode

To edit the failure recovery operational mode:

Step 1 From the ***SCE*** `(config)#` prompt, type **`failure-recovery operation-mode operational|non-operational`** and press **Enter**.

Enter either the value **`operational`** or **`non-operational`**.

EXAMPLE 1:

The following example sets the system to boot as operational after a failure

```
SCE(config)#failure-recovery operation-mode operational
SCE(config)#
```

EXAMPLE 2:

The following example sets the system to the default failure recovery mode.

```
SCE(config)# default failure-recovery operation-mode
SCE(config)#
```

SCE Platform/SM Connection

The user can configure the behavior of the ***SCE*** platform in case of failure of the smartSUB Manager (SM):

- If SM functionality is critical to the operation of the system: configure forced failure of the ***SCE*** platform in the event of any loss of connection with the SM (may be due either to failure of the SM or failure of the connection itself).
- If SM functionality is not critical to the operation of the system: no action needs to be configured.

To configure forced failure of the ***SCE*** platform in case of failure of the SM:

Step 1 From the ***SCE***(config if)# prompt, type `subscriber sm-connection-failure action force-failure` and press **Enter**.

Enabling and Disabling Link Failure Reflection

In some topologies, link failure on one port must be reflected to the related port in order to allow the higher layer redundancy protocol in the network to detect the failure and function correctly. The **`link failure-reflection`** command determines the behavior of the system when there is a link problem.

The **`link failure-reflection`** command enables reflection of a link failure. Use the **[no]** form of this command to disable failure reflection on the link.

- `[no] link failure-reflection`

The default value is **disabled.**

To enable reflection of link failure:

Step 1 From the ***SCE***`(config)#` prompt, type **`interface Linecard 0`**, and press **Enter**.

The ***SCE***`(config if)#` prompt appears.

Step 2 Type **`link failure-reflection`** and press **Enter**.

Failure reflection on the link is enabled, and the ***SCE***`(config if)#` prompt appears.

Enabling and Disabling Link Failure Reflection on All Ports

The Link reflection on all ports feature extends the link failure reflection feature. allows the user to determine whether all ports should be taken down if a single port link fails.

In certain topologies, when a failure state occurs on one link, the link state must be reflected to all ports in order to signal any element using this SCE platform that the device is in a failure state, and therefore cannot be used.

Note The Link reflection on all ports feature cannot be used in a cascade mode, because in this mode one of the links is used to provide redundancy.

In link reflection on all ports mode, all ports of the SCE Platform are forced down and the link state of the first port is reflected on all the ports.

When recovering from the failure state, the forced down ports (the other link) are brought up only after the the first failed port (link) has recovered. In addition, the reflection algorithm will not try to reflect failure for this link again for the next 15 seconds, to avoid link stability problems on auto-negotiation.

The **`on-all-ports`** keyword enables reflection of a link failure to all ports. Use the **[no]** form of this command to disable failure reflection to all ports (the **`on-all-ports`** keyword is not used in the [no] form of the command).

- `[no] failure-reflection [on-all-ports]`

The default value is **disabled.**

To enable reflection of link failure to all ports:

Step 1 From the ***SCE***`(config)#` prompt, type **`interface Linecard 0`**, and press **Enter**.

The ***SCE***`(config if)#` prompt appears.

Step 2 Type **`link failure-reflection on-all-ports`** and press **Enter**.

Failure reflection to all ports is enabled, and the ***SCE***`(config if)#` prompt appears.

To disable reflection of link failure:

Step 1 From the ***SCE***`(config)#` prompt, type **`interface Linecard 0`**, and press **Enter**.

The ***SCE***`(config if)#` prompt appears.

Step 2 Type **`no link failure-reflection`** and press **Enter**.

Failure reflection is disabled, and the ***SCE***`(config if)#` prompt appears.

Configuring the RDR Formatter

This chapter contains the following sections:

- The RDR Formatter 8-1

The RDR Formatter

The RDR formatter is used to gather the streams of events passed from the application, format the data into Raw Data Records (RDRs), and send these RDRs to the appropriate destination(s).

There can be a maximum of four destinations for the RDRs. The system decides which destination to send the RDRs to on the basis of three factors:

- Categories: RDRs may be divided into four categories, with each category being assigned to a maximum of three of the defined destinations. A destination may be assigned to more than one categorie.
- Priority: The priority value assigned to the destination for a specific category
- Forwarding mode: the pattern in which the RDR traffic is divided between the various destinations

RDR Formatter Destinations

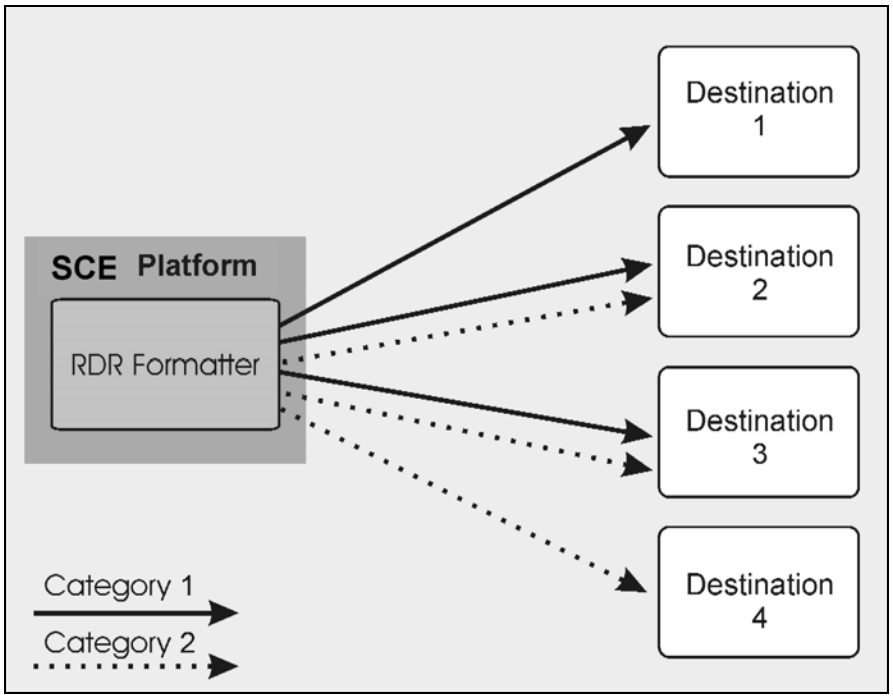
The ***SCE*** platform can be configured with a maximum of four RDR destinations, three destinations per category. Each destination is defined by its IP address and TCP port number, and is assigned a priority for each category to which it is assigned.

The following figure illustrates the simplest RDR formatter topology, with only one category and one destination.

Figure 8-1: Simple RDR Formatter Topology

The following figure illustrates a complex topology using both categories and the maximum number of destinations (four). Each category can send RDRs to three of the four destinations.

Figure 8-2: RDR Formatter Topology with Multiple Destinations



Categories

In certain installations, RDRs must be sent to different collector servers according to their type. For instance, in the pre-paid environment, some RDRs must be sent to the pre-paid collector to get a new quota, while others should be sent to the mediation system. In this case, the RDRs are divided into four groups, and each group, or category, is assigned to a particular destination or destinations.The categories are defined as follows:

- **Category 1**: Usage RDRs to Collection Manager\mediation system
- **Category 2**: Quota RDRs to Pre-Paid Server (e.g. Comverse) or Subscriber Controller OSS (e.g. Tazz)
- **Category 3**: External events RDR \ RT Signaling to various systems such as a Packet Cable Multi Media Policy Server
- **Category 4**: URL Query RDR to URL Filtering DB (e.g. surfControl)

(Assigning the RDRs to categories is defined by the application running on the ***SCE*** platform.)

The system supports four categories. Therefore, the RDR formatter destinations must be configured regarding each category. Each destination may be assigned to more than one category and may be assigned the same or different priorities for each category. If more than one destination is defined for a category, a load-balancing or multicast forwarding mode could be selected. (Obviously, these modes have no meaning of there is only one destination per category.)

It is also possible to remove a category from a destination, leaving only the desired category. If all categories are removed, the destination itself is deleted.

By default, the categories are referred to as Category 1 through Category 4. However, the user may define meaningful names for the categories. This generally reduces confusion and prevents errors.

Priority

The priority value is used to indicate whether the destination should be a destination for a given category. A high priority indicates that RDRs from a category should be sent a particular destination. No priority indicates that RDRs from a category should not be sent to a particular destination.

Priority also is related to the redundant forwarding mode, in that it indicates which is the primary active connection. Priority values have no effect in simple-load-balancing or multicast forwarding modes.

Each destination is assigned a priority value for each category. The first destination that is configured is automatically assigned a priority of 100 (highest priority) for all categories, unless explicitly defined otherwise.

Following are some important points to keep in mind regarding priority values:

- Two destinations may not have the same priority for one category. The priority values for destinations within a category must be unique in order to have any meaning.
- If only one category is defined by the application, the second priority value is ignored.
- If only one priority value is assigned to the destination, that priority is automatically assigned to all categories for that destination.
- If only one category is assigned a priority value for a destination, no RDRs from the other categories will be sent to the specified destination.
- Assign a high priority if RDRs from the specified category should be sent to this destination. Assign a low priority if RDRs from the specified category should less likely to be sent to this destination.
- Redundant forwarding mode: Assign a high priority to the primary destination for the system/category. Assign a lower priority to the secondary destination for the system/category.

Forwarding Modes

When more than one RDR destination is defined for a category, the system must decide which of these destinations is to receive the RDRs. This is determined by the forwarding mode. There are two forwarding modes:

- Redundancy: All RDRs are sent only to the primary (active) connection. If the primary connection fails, the RDRs will be sent to the connected destination with the next highest priority.
- Multicast: All RDRs are sent to all destinations. This feature may negatively affect performance in an installation with a high rate of RDRs.

Configuring the RDR Formatter

The following commands are relevant to the RDR-formatter:

- `RDR-formatter forwarding-mode`
- `service RDR-formatter`
- `no service RDR-formatter`
- `RDR-formatter destinations:`
  - RDR-formatter destination
  - no RDR-formatter destination
  - no RDR-formatter destination all
- `RDR-formatter categories:`
  - RDR-formatter category-number
  - no RDR-formatter category-number

To configure the RDR Formatter forwarding mode:

Step 1 From the ***SCE***`(config)#` prompt, type **`RDR-Formatter forwarding-mode <redundancy>|<multicast>`**, and press **Enter**.

The specified RDR Formatter forwarding mode is defined.

EXAMPLE:

The following example shows how to set the RDR Formatter forwarding-mode to multicast

```
SCE(config)# RDR-Formatter forwarding-mode multicast
```

Configuring the RDR Formatter Destinations

In order for the RDRs from the ***SCE*** platform to arrive at the correct location, the IP address of the destination and its TCP port number must be configured.

A priority value must be assigned. Priority is important in the redundancy forwarding mode, but not crucial in simple-load-balancing mode or multicast mode. Remember that in load-balancing and multicast modes, the existence of any priority value causes the destination to receive RDRs.

The relationship between priorities and categories is addressed in the next section.

To configure an RDR Formatter destination (all categories):

Step 1 From the ***SCE***`(config)#` prompt, type **`RDR-Formatter destination <IP address> port <port-number> [priority <priority(1-100)>]`**, and press **Enter**.

The RDR Formatter destination is defined. When no category is specified, as in the above example, the specified priority is assigned to all categories.

EXAMPLE:

The following example shows how to configure two RDR Formatter destinations in a system without using the categories.

The first destination will automatically be assigned a priority of 100, and therefore the priority does not need to be explicitly defined. For the second destination, the priority must be explicitly defined.

The same priority will automatically be assigned to both categories for each destination, but since the categories will be ignored, this is irrelevant.

```
SCE(config)# RDR-Formatter destination 10.1.1.205 port 33000
SCE(config)# RDR-Formatter destination 10.1.1.206 port 33000 priority 80
```

Configuring the RDR Formatter Categories

There are two steps in defining the RDR formatter destination categories:

Step 1 Define the category names (optional).

Step 2 Assign the destinations to both categories.

Configuring the destinations with the proper priorities for each category, as well as configuring all the other RDR formatter parameters, may be approached in several different ways, and may take some planning. Refer to the examples below for illustrations of some of the issues involved in configuring categories.

To configure an RDR Formatter category name:

Step 1 From the ***SCE***`(config)#` prompt, type **`RDR-Formatter category-number 1-4 name <category-name>`**, and press **Enter**.

The name for the specified category number is defined. This category name can then be used in any **RDR-formatter** command instead of the category number.

To configure a RDR Formatter destination and assign it to a category:

Step 1 From the ***SCE***`(config)#` prompt, type **`RDR-Formatter destination <IP address> port <port-number> category [name <category-name> |number [1-4]] [priority <priority(1-100)>] [category [name <category-name> |number [1-4]] [priority <priority(1-100)>]]`**, and press **Enter**.

The RDR Formatter destination is defined. A different priority may be assigned to each category. (This can be done in one command for a maximum of two categories.) If RDRs from the specified category should be sent to this destination, the priority for the category should be high. If the RDRs from the specified category should not be sent to this destination, the priority should be low.

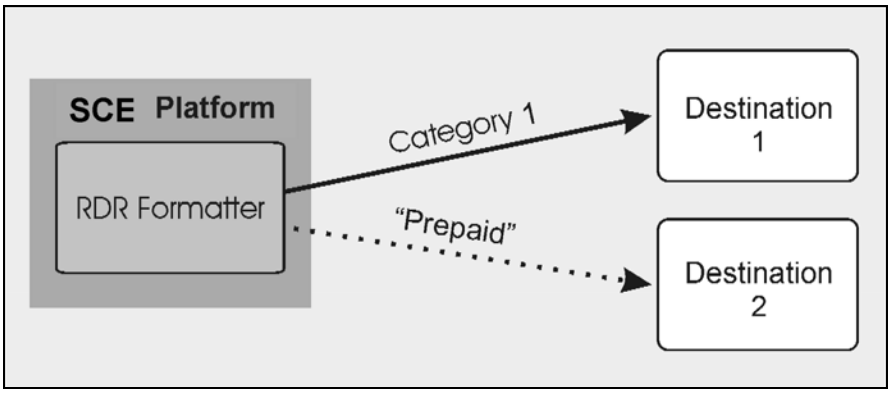
Note that within each category the priorities must be unique for each destination.

EXAMPLE 1:

The following example defines a name for one category, and then configures two RDR Formatter destinations, assigning each to a different category (see diagram).

The RDRs of category 1 are to go to the first destination, so a high priority was assigned to that category in the first destination, and no priority in the second.

Since all RDRs in category 2 (prepaid) are to go to the second destination, the priority assigned to category 2 is assigned only to the second destination and not to the first.

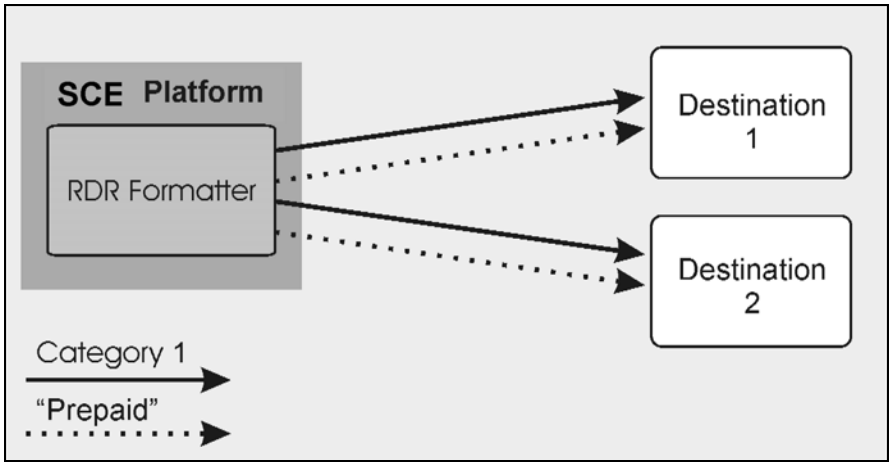


Note that if there is a loss of connection to either destination, transmission of RDRs of the relevant category is interrupted until the connection is re-established. There is no redundant connection defined for either category.

```
SCE(config)# RDR-Formatter category-number 2 name prepaid
SCE(config)# RDR-Formatter destination 10.1.1.205 port 33000 category number
1 priority 90
SCE(config)# RDR-Formatter destination 10.1.1.206 port 33000 category name
prepaid priority 80
```

EXAMPLE 2:

This example is similar to the above, but a low priority is assigned to the second category for each destination, rather than no priority. This allows each destination to function as a backup for the other in case of a problem with one of the connections (redundancy forwarding mode).




```
SCE(config)# RDR-Formatter category-number 2 name prepaid
SCE(config)# RDR-Formatter destination 10.1.1.205 port 33000 category name
prepaid priority 90 category number 1 priority 25
SCE(config)# RDR-Formatter destination 10.1.1.206 port 33000 category number
1 priority 80 category name prepaid priority 20
```

EXAMPLE 3:

This example demonstrates two methods for assigning one category to the first destination only, while the other category uses the second destination as the primary destination, and the first destination as a secondary destination.

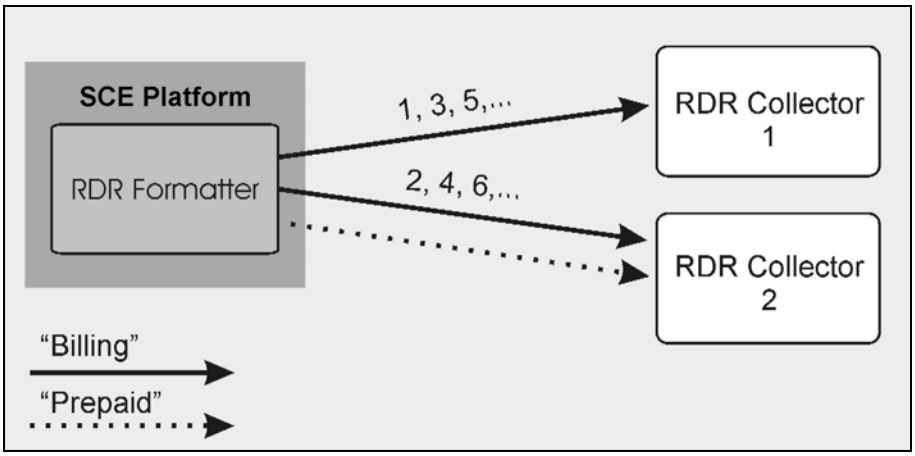
```
SCE(config)# RDR-Formatter category-number 2 name prepaid
SCE(config)# RDR-Formatter destination 10.1.1.205 port 33000 category name
prepaid priority 90 category number 1 priority 10
SCE(config)# RDR-Formatter destination 10.1.1.206 port 33000 category number
1 priority 95
```

In the following example, all priority values seem quite high. However, it is the relative values of priorities for a category that determine which destination is the primary destination.

```
SCE(config)# RDR-Formatter category-number 2 name prepaid
SCE(config)# RDR-Formatter destination 10.1.1.205 port 33000 priority 90
SCE(config)# RDR-Formatter destination 10.1.1.206 port 33000 priority 95
SCE(config)# no RDR-Formatter destination 10.1.1.206 port 33000 category
name prepaid
```

EXAMPLE 4:

Finally, the following illustrates a more complex configuration with one category (prepaid) assigned to one destination and the other (billing) being sent to both destinations, in multicast mode.



The forwarding mode is defined for the entire RDR formatter, not just one category. Since the category "prepaid" goes to only one destination, the forwarding mode is irrelevant. It is relevant, however to the "billing" category, since it goes to two different destinations.

```
SCE(config)# RDR-Formatter forwarding-mode multi-cast
SCE(config)# RDR-Formatter category-number 1 name billing
SCE(config)# RDR-Formatter category-number 2 name prepaid
SCE(config)# RDR-Formatter destination 10.1.1.205 port 33000 priority 40
SCE(config)# no RDR-Formatter destination 10.1.1.205 port 33000 category
name billing
SCE(config)# RDR-Formatter destination 10.10.10.96 port 33000 category name
billing priority 90
SCE(config)# RDR-Formatter destination 10.1.96.0 port 33000 category name
billing priority 80
```

Displaying RDR Formatter Configuration and Statistics

The system can display the complete RDR formatter configuration, or just specific parameters.

The following commands can be used to display the RDR formatter configuration and statistics:

- `show RDR-formatter`
- `show RDR-formatter connection-status`
- `show RDR-formatter counters`
- `show RDR-formatter destination`
- `show RDR-formatter enabled`
- `show RDR-formatter forwarding-mode`
- `Show RDR-formatter rdr-mapping`
- `show RDR-formatter statistics`

To display the current RDR formatter configuration:

Step 1 From the ***SCE#*** prompt, type **`show RDR formatter`**.

EXAMPLE:

The following example shows how to display the current RDR formatter configuration.

```
SCE#show RDR-formatter
Status: enabled
Connection is: up
Forwarding mode: redundancy
Connection table:
-------------------------------------------------------------------------|
Collector   | Port|Status|           Priority per Category:                |
IP Addres / |     |      |-------------------------------------------------|
Host-Name   |     |      |Category1  |Category2  |Category3  |Category4  |
-------------------------------------------------------------------------|
10.1.1.205  |33000|Up    |100 primary|100 primary|100 primary|100 primary|
10.1.1.206  |33000|Down  |60         |60         |60         |60         |
10.12.12.12 |33000|Up    |40         |40         |40         |40         |
-------------------------------------------------------------------------|

RDR:    queued:         0 ,sent:         0, thrown:         0
UM:     queued:         0 ,sent:         0, thrown:         0
Logger: queued:         0 ,sent:         0, thrown:         0
Errors: thrown:         0
Last time these counters were cleared: 14:05:57 UTC SUN February 23 2003
SCE#
```

Refer to CLI Command Reference for a complete description of the other **show RDR-formatter** commands.

Disabling the LineCard from Sending RDRs

The **silent** command disables the LineCard from issuing Raw Data Records (RDR). Use the **[no]** form of this command if you want the LineCard to send reports.

To disable the LineCard from sending Raw Data Records (RDRs):

Step 1 From the ***SCE***(config)# prompt, type **interface Linecard 0**, and press **Enter**.

The ***SCE***(config if)# prompt appears.

Step 2 Type **silent**, and press **Enter**.

The LineCard stops producing RDRs and the ***SCE***(config if)# prompt appears.

To enable the Line Card to produce RDRs:

Step 1 From the ***SCE***(config if)# prompt, type **no silent,** and press **Enter** .

The ***SCE***(config if)# prompt appears.

Managing Subscribers

The ***SCE*** patform is subscriber aware, that is, it can relate traffic and usage to specific customers. This ability to map between IP flows and a specific subscriber allows the system to do the following:

- Maintain the state of each subscriber transmitting traffic through the platform
- Provide usage information for specific subscribers
- Enforce the appropriate policy on subscriber traffic (each subscriber can have a different policy)

This chapter contains the following sections:

- Subscriber Overview 9-2
- Importing/Exporting Subscriber Information 9-6
- Removing Subscribers and Templates 9-8
- Importing/Exporting Anonymous Groups 9-10
- Monitoring Subscribers 9-10
- Subscriber Traffic Processor IP Ranges 9-18
- Subscriber Aging 9-26
- SCE Platform/SM Connection 9-28

Subscriber Overview

In the Service Control solution, a subscriber is defined as a managed entity on the subscriber side of the SCE Platform to which accounting and policy are applied individually.

The following table lists several examples of subscribers in Service Control solutions.

Table 9-1 Subscriber Examples

The Subscriber	**Subscriber Characteristics**	
	Managed Entity	**Subscriber (Entity) Identified By**
DSL residential subscriber	DSL residential user	IP address The list of IP addresses is allocated by a Radius server
Cable residential subscriber	Cable residential user	IP address The list of IP addresses of the CPEs is allocated dynamically by a DHCP server
Owner of a 3G-phone that is subscribed to data services	3G-phone owner	The MS-ISDN, which is dynamically allocated by a Radius server.
A corporate/enterprise customer of the service provider	The corporate/enterprise and the traffic it produces	The set of NAT-ed IP addresses, which are allocated statically
A CMTS	The CMTS and the broadband traffic of the Cable Modem users that connect to the Internet through the CMTS	• A range of IP addresses • A group of VLAN tags

Mapping IP traffic flows to subscribers enables the SCE Platform to enforce policies on these flows based on the subscriber who produced them.

The SCE Platform can also insert the information that identifies the subscriber into the RDR records that it produces for analyzed traffic, facilitating OSS systems that use these data records for billing and analysis purposes.

The SCE Platform includes dedicated infrastructure for per-subscriber BW shaping, IP traffic quota management, or any other per-subscriber long-term state management. This is implemented using a set of dedicated data structures that are dynamically managed in the SCE Platform per subscriber.

The SCE Platform examines each IP flow and maps it to the subscriber that produced the flow using one or more networking parameters of this flow. Examples of these could be:

- Source IP address
- Group of source IP addresses
- Range of source IP addresses
- VLAN tag

These parameters are sometimes referred to as *Network-ID*. In order to perform the mapping between the Network-ID and Subscriber-ID, the SCE Platform must be configured with this mapping information.

In some cases the subscriber's Network-ID is static and changes only rarely and at long intervals. In such cases, obtaining the mapping information is quite simple, and can be implemented by importing the content of a text file, or even by typing the information via the user interface. In other cases, the Network-ID has a dynamic nature, and tends to change every time the subscriber logs into the network. In this case the SCE Platform must obtain the mapping information from some element that stores this information.

The most common Network-IDs are IP addresses. Typically, obtaining the mappings between subscriber-IDs and IP addresses is done through integration with an AAA element or a subscriber repository.

Many times, the SCE Platform runs a Service Control Application that is policy-driven, so it should also be provisioned with the parameters of the policy that should be applied to each of the subscribers. In simple cases, there is only a small set of standard policy packages (Gold, Silver, Bronze…) so the per subscriber information includes only an index into the policies list. In other cases, a whole set of policy parameters should be configured per subscriber. Often the policy that should be applied per subscriber is managed using the same AAA infrastructure that is used for managing the Subscriber-ID to Network-ID mappings.

There are two methods of managing subscribers:

- smartSUB Manager (SM) component: usually necessary in topologies where full dynamic subscriber integration is required (see the *smartSUB Manager User Guide* for details).
- CLI commands: can be used to import and export subscriber information, as well as to monitor subscribers.

As is described in the following sections, subscriber-related information can be imported from external files. This provides an easy method for transferring large quantities of subscriber information to and from the SCE Platform.

Subscriber Modes in Service Control Solutions

Service Control solutions support several modes of handling subscribers:

- Subscriber-less mode
- Anonymous subscriber mode
- Static subscriber aware mode
- Dynamic subscriber aware mode

Note that not all the solutions support all modes.

The most basic mode is **Subscriber-less mode**. In this mode, there is no notion of subscriber in the system, and the entire link where the SCE Platform is deployed is treated as a single subscriber. Global Application level analysis (such as total p2p, browsing) can be conducted, as well as global control (such as limiting total p2p to a specified percentage). From a configuration stand point, this is a turnkey system and there is no need to integrate or configure the system from a subscriber perspective.

In **Anonymous subscriber mode**, analysis is performed on an incoming subscriber-IP address, as the SCE Platform creates an 'anonymous/on-the-fly' record for each subscriber. This permits analyzing traffic at an individual IP address level (for example, to identify/monitor what a particular 'subscriber' IP is currently doing) as well as control at this level (for example, to limit each subscriber's bandwidth to a specified amount, or block, or redirect). Anonymous-subscriber allows quick visibility into application and protocol usage without OSS integration, and permits the application of a uniform control scheme using predefined templates.

There are two possible **Subscriber Aware modes**. In these modes, subscriber IDs and currently used IP addresses are provisioned into the SCE Platform. The SCE Platform can then bind usage to a particular subscriber, and enforce per-subscriber policies on the traffic. Named reports are supported (such as top subscribers with the OSS IDs), quota-tracking (such as tracking a subscriber-quota over time even when IP addresses change) as well as dynamic binding of packages to subscribers. The two Subscriber Aware modes are:

- **Static subscriber aware:** The IP addresses are static. The system supports the definition of static-subscribers directly to the SCE Platform. This is achieved by using the SCE Platform CLI, and defining the list of subscribers, their IP addresses and policy information using interactive configuration or import/export operations.
- **Dynamic subscriber aware:** The IP addresses change dynamically for each subscriber login into the Service Provider’s network. In this case, subscriber awareness is achieved by integrating with AAA and provisioning systems for dynamically obtaining network-ID to subscriber ID mappings, and distributing them to the SCE Platforms.

Aging Subscribers

Subscribers can be aged automatically by the ***SCE*** platform. ‘Aging’ is the automatic removal of a subscriber, performed when no traffic sessions assigned to it have been detected for a certain amount of time. The most common usage for aging is for anonymous subscribers, since this is the easiest way to ensure that anonymous subscribers that have logged-out of the network are removed from the ***SCE*** platform and are no longer occupying resources. Aging time can be configured individually for introduced subscribers and for anonymous subscribers.

Anonymous Groups and Subscriber Templates

An anonymous group is a specified IP range, possibly assigned a subscriber template. When an anonymous group is configured, the SCE Platform generates anonymous subscribers for that group when it detects traffic with an IP address that is in the specified IP range. If a subscriber template has been assigned to the group, the anonymous subscribers generated have properties as defined by that template. If no subscriber template has been assigned, the default template is used.

Subscriber templates are identified by a number from 0-199. Subscriber templates 1-199 are defined in *csv* formatted subscriber template files. However, template #0 cannot change; it always contains the default values.

If an anonymous group is not explicitly assigned a template, the group uses template #0.

Subscriber Files

Individual subscribers, anonymous groups, and subscriber templates may all be defined in `csv` files. A `csv` file is a text file in a comma-separated-values format. Microsoft Excel™ can be used to view and create such files. The subscriber data is imported into the system using the appropriate CLI command. The ***SCE*** platform can also export the currently configured subscribers, subscriber templates and anonymous groups to csv-formatted files

Subscriber `csv` files and subscriber template `csv` files are application-specific. Refer to the relevant application documentation for the definition of the file format.

Each line in a `csv` file should contain either a comment (beginning with the character '#'), or a list of comma-separated fields.

Subscriber `csv` files are application-specific, but a default format is defined by the SCE, which is used when the application does not choose to over-ride it. The application might over-ride the format when additional data is desired for each subscriber or subscriber template. Refer to the relevant Service Control Application documentation to see if the application defines a different format.

Subscriber template `csv` files are application-specific. Refer to the relevant Service Control Application documentation of the file format.

Anonymous groups `csv` files are not application specific. Their format is described below.

Subscriber default csv file format

Each line has the following structure:

`name, mappings`

- **Name:** is the subscriber name
- **Mappings:** contains one of more mappings, specifying the Tunnel IDs or IP addresses mapped to this subscriber. Multiple mappings are separated by semi-colon. Tunnel IDs and IP address/range cannot be specified for the same subscriber. The following mapping formats are supported:
  - Tunnel ID: A number in the range 0-1023. Example: 4

Note Currently only VLAN IDs are supported.

- Tunnel ID range: A range of tunnel Ids. Example: 4-8
- IP address: in dotted decimal notation. Example: 10.3.4.5
- IP address range: dotted decimal, followed by the amount of significant bits. Note that the non-significant bits (As determined by the mask) must be set to zero. Example: 10.3.0.0/16. Example for a bad range: 10.1.1.1/24 (Should have been 10.1.1.0/24).

Here is an example for a subscriber `csv` file in the default format:

```
# A comment line
sub7, 10.1.7.0/24
sub8, 10.1.12.32
sub9, 5
sub10, 13-17
sub11, 39;41
sub12, 10.1.11.90; 10.3.0.0/16
```

Subscriber anonymous groups csv file format

Each line has the following structure:

`name, IP-range, template-index`

- **Name:** is the anonymous group name
- **IP-range:** dotted decimal, followed by the amount of significant bits. Example: 10.3.0.0/16
- **Template-index:** is the index of the subscriber template to be used by subscribers belonging to this anonymous group.

Here is an example for an anonymous groups `csv` file:

```
# Yet another comment line
anon1, 10.1.1.0/24, 1
anon2, 10.1.2.0/24, 2
anon3, 10.1.3.0/32, 3
anon4, 10.1.4.0/24, 3
anon5, 10.1.5.0/31, 2
anon6, 10.1.6.0/30, 1
anon7, 0.0.0.0/0, 1
```

Importing/Exporting Subscriber Information

Use the following commands to import subscriber data from *csv* files and to export subscriber data to these files:

- `subscriber import csv-file`
- `subscriber export csv-file`
- `subscriber anonymous-group import csv-file`
- `subscriber anonymous-group export csv-file`

- `subscriber template import csv-file`
- `subscriber template export csv-file`

These subscriber management commands are LineCard interface commands. Make sure that you are in LineCard Interface command mode, (see *Entering LineCard Interface Configuration Mode* (Entering LineCard Interface Configuration Mode "Entering LineCard Interface Configuration Mode" on page 2-8)).

Importing/Exporting Subscribers

To import subscribers from the csv subscriber file:

Step 1 From the ***SCE***`(config if)#` prompt, type **`subscriber import csv-f`***`ile file`*name and pr**ess E**nter.

The subscriber information is imported from the specified file and the ***SCE***`(config if)#` prompt appears.

Imported subscriber information is added to the existing subscriber information. It does not overwrite the existing data.

If the information in the imported file is not valid, the command will fail during the verification process before it is actually applied.

To export subscribers to a csv subscriber file:

Step 1 From the ***SCE***`(config if)#` prompt, type **`subscriber export csv-file`** *`filename`* and press **Enter**.

Subscriber information is exported to the specified file and the ***SCE***`(config if)#` prompt appears.

Importing/Exporting Subscriber Templates

To import a subscriber template from the csv file:

Step 1 From the ***SCE***`(config if)#` prompt, type **`subscriber template import`** *`csv-file`* filename a**nd pr**ess Enter.

The subscriber template is imported from the specified file and the ***SCE***`(config if)#` prompt appears.

To export a subscriber template to a csv file:

Step 1 From the ***SCE***`(config if)#` prompt, type **`subscriber template export csv-file`** *`filename`* and press **Enter**.

The subscriber template is exported to the specified file and the ***SCE***`(config if)#` prompt appears.

Removing Subscribers and Templates

Use the following commands to remove all subscribers, anonymous groups, or subscriber templates from the system.

- `no subscriber all`
- `no subscriber anonymous-group all`
- `clear subscriber anonymous`
- `default subscriber template all`

Use the following commands to remove a specific subscriber or anonymous group from the system.

- `no subscriber name`
- `no subscriber anonymous-group name`

These subscriber management commands are LineCard interface commands. Make sure that you are in LineCard Interface command mode, (see “Entering LineCard Interface Mode,” page and that the ***SCE***`(config if)#` prompt appears in the command line.

To remove a specific subscriber:

Step 1 From the ***SCE***`(config if)#` prompt, type **`no subscriber name`** *`subscriber-name`* and press **Enter**.

The specified subscriber is removed from the system, and the ***SCE***`(config)#` prompt appears.

To remove all introduced subscribers:

Step 1 From the ***SCE***`(config if)#` prompt, type **`no subscriber all`** and press **Enter**.

All introduced subscribers are removed from the system, and the ***SCE***`(config)#` prompt appears.

To remove a specific anonymous subscriber group:

Step 1 From the ***SCE***(config if)# prompt, type **no subscriber anonymous-group name** *group-name* and press **Enter**.

The specified anonymous group is removed from the system, and the ***SCE***(config)# prompt appears.

To remove all anonymous subscriber groups:

Step 1 From the ***SCE***(config if)# prompt, type **no subscriber anonymous-group all** and press **Enter**.

All anonymous groups are removed from the system, and the ***SCE***(config)# prompt appears.

To remove all anonymous subscribers:

Step 1 From the ***SCE***# prompt, type **clear interface linecard 0 subscriber anonymous all** and press **Enter**.

All anonymous subscribers are removed from the system, and the ***SCE***(config)# prompt appears.

Note The **clear subscriber anonymous** command is a Privileged Exec command.

To remove all subscriber templates:

Step 1 From the ***SCE***(config if)# prompt, type **default subscriber template all** and press **Enter**.

All subscriber templates are removed from the system, and the ***SCE***(config)# prompt appears. All anonymous subscribers will be assigned to the default subscriber template.

Importing/Exporting Anonymous Groups

To create anonymous groups by importing anonymous subscribers from the csv file:

Step 1 From the ***SCE***(config if)# prompt, type **subscriber anonymous-group import** *csv-file* filename **and p**ress Enter.

The anonymous subscriber information is imported from the specified file, creating anonymous groups and the ***SCE***(config if)# prompt appears.

Imported anonymous subscriber information is added to the existing anonymous subscriber information. It does not overwrite the existing data.

To export anonymous groups to a csv file:

Step 1 From the ***SCE***(config if)# prompt, type **subscriber anonymous-group export csv-file** *filename* and press **Enter**.

The anonymous groups are exported to the specified file and the ***SCE***(config if)# prompt appears.

Monitoring Subscribers

The CLI provides a number of commands that allow you to monitor subscribers. These commands can be used to display information regarding the following:

- Subscriber Database
- All subscriber meeting various criteria
- Individual subscriber information, such as properties and mappings
- Anonymous subscribers

Subscribers may be introduced to the SCE Platform via the SCE Platform CLI or via the smartSUB Manager. The monitoring commands may be used to monitor all subscribers and subscriber information, regardless of how the subscribers were introduced to the system.

Note that these commands are all in Privileged Exec mode. Make sure that you are in the proper mode and that the ***SCE#*** prompt appears in the command line. Note also that you must specify '**linecard 0**' in these commands.

Monitoring the Subscriber Database

Use the following commands to display statistics about the subscriber database, and to clear the "**total**" and "**maximum**" counters.

- `show interface linecard 0 subscriber db counters`
- `clear interface linecard 0 subscriber db counters`

To display statistics about the subscriber database:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber db counters`** and press **Enter**.

The following counters are displayed:

- Current number of subscribers
- Current number of introduced subscribers
- Current number of anonymous subscribers
- Current number of active subscribers (with active traffic sessions)
- Current number of subscribers with mappings
- Current number of IP mappings
- Current number of vlan mappings
- Max number of subscribers that can be introduced
- Max number of subscribers with mappings
- Max number of subscribers with mappings date / time
- Total aggregated number introduced
- Total number of aged subscribers
- Total number of pull events
- Number of traffic sessions currently assigned to the default subscriber

To clear subscriber database counters:

Step 1 From the ***SCE#*** prompt, type **`clear interface linecard 0 subscriber db counters`** and press **Enter**.

The "**total**" and "**maximum**" counters are cleared (see list above).

Displaying Subscribers

You can display specific subscriber name(s) that meet various criteria:

- A subscriber property is equal to, larger than, or smaller than a specified value
- Subscriber name matches a specific prefix or suffix
- Mapped to a specified IP address range
- Mapped to a specified VLAN ID

Use the following commands to display subscribers:

- `show interface linecard 0 subscriber [amount]`
- `[prefix ‘prefix’] [property ‘propertyname’ equals|greater-than|less-than ‘property-val’]`
- `show interface linecard 0 subscriber [amount] prefix ‘prefix’`
- `show interface linecard 0 subscriber [amount] suffix ‘suffix’`
- `show interface linecard 0 subscriber mapping IP ‘iprange’`
- `show interface linecard 0 subscriber [amount] mapping intersecting IP ‘iprange’`
- `show interface linecard 0 subscriber mapping VLANid ‘vlanid’`

Displaying Subscribers: By Subscriber Property or Prefix

You can search for all subscribers that match a specified value of one of the subscriber properties, or are greater than or less than the specified value. You can also search for all subscribers that match a specified prefix. You can also find out how many subscribers match any one of these criteria, rather than displaying all the actual subscriber names.

To display subscribers that match a specified value of a subscriber property:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber property ‘propertyname’ equals ‘property-val’`** and press Enter.

To display subscribers that are greater than or less than a specified value of a subscriber property:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber property ‘propertyname’ greater-than|less-than ‘property val’`** and press **Enter**.

To display subscribers that match a specified prefix:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber prefix 'prefix'`** and press **Enter**.

To display subscribers that match a specified suffix:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber suffix 'suffix'`** and press **Enter**.

To display the number of subscribers that match a specified value of a subscriber property:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber amount property 'propertyname' equals 'property val'`** and press **Enter**.

To display the number of subscribers that are greater than or less than a specified value of a subscriber property:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber amount property 'propertyname' greater-than|less-than 'property val'`** and press **Enter**.

To display the number of subscribers that match a specified prefix:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber amount prefix 'prefix'`** and press **Enter**.

To display the number of subscribers that match a specified prefix:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber amount suffix 'suffix'`** and press **Enter**.

Displaying Subscribers: By IP Address or VLAN ID

You can display the subscribers who are mapped to any of the following:

- A specified IP address, or range of IP addresses
- IP addresses intersecting a given IP address or IP range
- A specified VLAN ID
- no mapping

You can also display just the number of subscribers are mapped to IP addresses that intersect a given IP address or IP range.

To display subscribers that are mapped to a specified IP address, or range of IP addresses:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber mapping IP 'iprange' and`** press Enter.

To display subscribers that are mapped to IP addresses that intersect a given IP address or IP range:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber mapping intersecting IP 'iprange'`** and press **Enter**.

To display subscribers that are mapped to a specified IP address, or range of IP addresses:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber mapping IP 'iprange'`** and press **Enter**.

To display subscribers with no mapping:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber mapping none`** and press **Enter**.

To display the number of subscribers that are mapped to IP addresses that intersect a given IP address or IP range:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber amount mapping intersecting IP 'iprange'`** and press **Enter**.

To display the number of subscribers with no mapping:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber amount mapping none`** and press **Enter**.

Displaying Subscriber Information

You can display the following information about a specified subscriber:

- values of the various subscriber properties
- mappings
- OS counters:
  - current number of flows
  - bandwidth

Use the following commands to display subscriber information:

- `show interface linecard 0 subscriber properties`
- `show interface linecard 0 subscriber name 'name'`
- `show interface linecard 0 subscriber name 'name' mappings`
- `show interface linecard 0 subscriber name 'name' counters`
- `show interface linecard 0 subscriber name 'name' properties`

To display a listing of subscriber properties:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber properties`** **and** press Enter.

To display complete information for a specified subscriber - all values of subscriber properties and mappings:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber name 'name'`** and press **Enter**.

To display values of subscriber properties for a specified subscriber:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber name 'name' properties`** and press **Enter**.

To display mappings for a specified subscriber:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber name 'name' mappings`** and press **Enter**.

To display the OS counters for a specified subscriber:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber name 'name' counters`** and press **Enter**.

Displaying Anonymous Subscriber Information

You can display the following information regarding the anonymous subscriber groups:

- aging (see *Subscriber Aging* (on page 9-26))
- currently configured anonymous groups
- currently configured subscriber templates
- configuration of a specified anonymous group
- number of subscribers in a specified anonymous group, or in all anonymous groups

Use the following commands to display anonymous subscriber information:

- `show interface linecard 0 subscriber templates [index]`
- `show interface linecard 0 subscriber anonymous-group [all] [name 'groupname']`

- `show interface linecard 0 subscriber amount anonymous [name ‘groupname’]`
- `show interface linecard 0 subscriber anonymous [name ‘groupname’]`

To display the currently configured anonymous groups:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber anonymous-group all`** **and** press Enter.

To display the currently configured templates for anonymous groups:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber templates`** and press **Enter**.

To display the current configuration for a specified anonymous group:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber anonymous-group name ‘groupname’`** and press **Enter**.

To display the subscribers in a specified anonymous group:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber anonymous name ‘groupname’`** and press Enter.

To display all subscribers in anonymous groups:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber anonymous`** and press **Enter**.

To display the number of subscribers in a specified anonymous group:

Step 1 From the ***SCE***# prompt, type **`show interface linecard 0 subscriber amount anonymous name 'groupname'`** and press **Enter**.

To display the total number of subscribers in anonymous groups:

Step 1 From the ***SCE***# prompt, type **`show interface linecard 0 subscriber amount anonymous`** and press **Enter**.

Subscriber Traffic Processor IP Ranges

In a Cable environment, the ***SCE*** platform supports the capability of associating all CPE machines in a single home network (i.e. behind a single cable modem) to a single subscriber-context and applying a single policy to this subscriber context. This is also relevant for cases where each CPE uses multiple global IP addresses (as opposed to a residential gateway that performs NAT and allows all CPE machines to share an IP address).

The ***SCE*** platform places no limit on the number of subscribers that can have multiple IP addresses. In order to achieve this, all IP addresses used by each CPE must use a common pool of addresses (usually that assigned with their downstream CMTS device/blade), and the subscriber that uses all these CPEs should be configured to a single traffic processor (a single PPC in the ***SCE*** platform).

Assigning subscribers to a specific traffic processor can be implemented in either of the following scenarios:

- All the IP ranges of a given CMTS/BRAS are configured to be processed by the same traffic processor. This can only be done if one ***SCE*** platform is handling several CMTS/BRAS (otherwise there is a load-balancing issue).
- The service provider can control the IP range from which the subscriber IP address is allocated based on additional criteria such as the subscriber type. In this case, the range can be used by the ***SCE*** platform to assign subscribers to a particular traffic processor, independent of the definition of the subscriber network ID.

In such cases the ***SCE*** platform (based on management configuration) can ensure that the various IP addresses (either ranges or single IPs) of each subscriber will actually be handled by the same traffic processor. This is accomplished by assigning a subscriber IP range (or specific IP address) to a configured Traffic Processor IP Range (TIR). Since each TIR is assigned to a traffic processor, the relevant subscriber IP range is also assigned to the matching traffic processor. Note that all ranges and single IPs of a specific subscriber must be assigned to the same traffic processor at any given time.

It is assumed that editing TIR configuration (addition or removal) is done infrequently. Also, that it is generally done either before the relevant IP ranges are in use or after they are no longer in use.

Subscriber Mapping Modes

The introduction of the TIR functionality provides two possible modes of subscriber mapping:

- Legacy subscriber mapping: ensures that all mappings of a single subscriber reach the same traffic processor by internal means, using a hash on the subscriber IP and/or using specific subscriber rules on the IP/range when required
- TIR subscriber mapping: generally (regarding any relevant subscribers) configures all mappings in a specific range to reach the same traffic processor, reducing the need for internal specific rule resources per subscriber.

TIRs functionality is not necessarily applicable to all subscribers. Therefore, while the user may choose to assign relevant subscribers to traffic processors via TIRs (TIRs subscriber mapping), the remaining subscribers are processed as usual (legacy subscriber mapping).

Subscriber Mapping Conflicts

It is important to note that while both subscriber mapping modes can co-exist in one deployment, any one subscriber can be processed only in one mode or the other. The same subscriber cannot be processed partially using TIRs subscriber mappings and partially using legacy subscriber mappings. The resulting conflicting subscriber mappings will be rejected.

Another cause of conflicting subscriber mappings is when a subscriber is assigned a new range or single IP that is associated with a traffic processor, different from that with which the subscriber is already associated.

Conflicting mapping are rejected (any other subscriber mappings are accepted as is) in both cases below:

- Conflict between mappings of a single mapping request.
- Additive subscriber mappings that conflict with existing mappings.

Subscriber Rules for TIRs

The number of reserved rules for potential TIRs is configurable, and is at the expense of explicit subscriber rules. The total number of subscriber rules available is approximately 8000.

- The maximum number of allowed reserved rules is 4096. The remaining rules are reserved for explicit subscriber mappings usage (used by the ***SCE*** platform to enable the legacy internal OS allocation of subscribers to traffic processors).
- By default 0 (zero) rules are reserved for TIRs.
- Updating this configuration is a major system event and can only be performed when no subscriber mappings or TIRs are configured.

Reserving Rules for TIRs

Use these commands to reserve rules for TIRs and to restore default subscriber rule allocation.

These commands are LineCard interface commands. Make sure that you are in LineCard Interface command mode, (see Entering LineCard Interface Configuration Mode) and that the ***SCE*** (config if)# prompt appears in the command line.

To reserve a specified number of subscriber rules for TIRs:

Step 1 From the ***SCE*** (config if)# prompt, type **`subscriber TP-mappings max-TP-IP-ranges`** *`rules-number`* and press **Enter.**

The specified number of rules are allocated for TIRs, decreasing the number of explicit subscriber rules available, and the ***SCE*** (config)# prompt appears.

To restore the default rule number allocation:

Step 1 From the ***SCE*** (config if)# prompt, type **`default subscriber TP-mappings`** and press **Enter**.

The default rule number allocation is restored (all 8000 rules for explicit subscriber rules and no rules reserved for TIRs), and the ***SCE*** (config)# prompt appears.

Configuring TIRs

Use this command to create or update a TIR. This command specifies the following:

- TIR name: meaningful name assigned to this traffic processor IP range
- IP range: IP address and mask defining the IP range
- Target traffic processor: number of the traffic processor to which this TIR is to be assigned

Editing TIRs is permitted only if there are no subscriber mappings within the relevant IP ranges. Therefore, by default, if subscriber mappings already exist for the either an updated or an existing IP range, the command will fail. However, you can specify that any existing subscriber mappings in the IP range should be removed (*`remove-subscriber-mappings`* keyword). In this case the command will execute successfully even if subscriber mappings exist.

These commands are LineCard interface commands. Make sure that you are in LineCard Interface command mode, (see *Entering LineCard Interface Configuration Mode* (on page 2-8)) and that the ***SCE*** (config if)# prompt appears in the command line.

To create/update a TIR:

Step 1 From the ***SCE***(config if)# prompt, type **`subscriber TP-IP-range name`** *`range-name`* **`IP-range`** *`ip-address/mask-length`* **`target-TP`** *`TP-num`* and press **Enter**.

- Creating: A TIR with the specified name and IP range is created and assigned to the specified traffic processor, and the ***SCE***(config)# prompt appears.
- Updating: The IP range and/or assigned traffic processor is updated for the specified TIR, and the ***SCE***(config)# prompt appears.
  - Updating the IP range: If subscriber mappings exist for either the new or the old IP range, the command will fail.

To update a TIR even if subscriber mappings exist:

Step 1 From the ***SCE***(config if)# prompt, type **`subscriber TP-IP-range name`** *`range-name`* **`IP-range`** *`ip-address/mask-length`* **`target-TP`** *`TP-num`* **`remove-subscriber-mappings`** and press **Enter**.

If subscriber mappings exist for either the new or the old IP range, they will be removed and the command will execute successfully.

Removing TIRs and Subscriber Mappings

Use these commands to remove existing TIRs and subscriber mappings. You can perform the following operations:

- Remove a specified TIR
- Remove all TIRs
- Remove all subscriber mappings assigned to a specified TIR
- Remove all subscriber mappings assigned to a specified IP range

As with updating a TIR, by default, if subscriber mappings already exist for the specified IP range, the command will fail. However, you can specify that any existing subscriber mappings in the IP range should be removed (*remove-subscriber-mappings* keyword). In this case the command will execute even if subscriber mappings exist.

These commands are LineCard interface commands. Make sure that you are in LineCard Interface command mode, (see *Entering LineCard Interface Configuration Mode* (on page 2-8)) and that the ***SCE***(config if)# prompt appears in the command line.

To remove a specfied TIR:

Step 1 From the ***SCE***(config if)# prompt, type `no subscriber TP-IP-range name` *range-name* **`[remove-subscriber-mappings]`** and press **Enter**.

The specified TIR is removed, and the ***SCE***(config)# prompt appears.

If subscriber mappings exist for this IP range, the command will fail. Specify **`remove-subscriber-mappings`** to remove any existing subscriber mappings for this IP range, and the command will execute successfully.

To remove all TIRs:

Step 1 From the ***SCE***(config if)# prompt, type **`no subscriber TP-IP-range all [remove-subscriber-mappings]`** and press **Enter**.

All existing TIRs are removed, and the ***SCE***(config)# prompt appears.

If subscriber mappings exist for any IP range, those TIRs will not be removed. Specify **`remove-subscriber-mappings`** to remove existing subscriber mappings for any IP range, and the command will execute successfully.

To remove subscriber mappings for a specified TIR:

Step 1 From the ***SCE***(config if)# prompt, type **`no subscriber mappings included-in TP-IP-range name`** *`range-name`* and press **Enter**.

All existing subscriber mappings are removed for the specified TIR, and the ***SCE***(config)# prompt appears.

The **`remove-subscriber-mappings`** option is not applicable to this command.

To remove subscriber mappings for a specified IP range:

Step 1 From the ***SCE***(config if)# prompt, type **`no subscriber mappings included-in IP-range`** *`ip-address/mask-length`* and press **Enter**.

All existing subscriber mappings are removed for the specified IP range, and the ***SCE***(config)# prompt appears.

The **`remove-subscriber-mappings`** option is not applicable to this command.

Importing and Exporting TIRs

Use these commands to import TIR definitions from a csv file and to export TIR definitions to a *csv* file.

Following is the format of the *csv* file:

range name, ip-address/mask-length, target-TP

- **range name**: The name of the to which the IP addresses will be assigned
- **ip-address/mask-length**: individual IP address of range of IP addresses indicated by IP address/mask
- **target-TP**: traffic processor to which the specified range will be assigned

When importing TIR definitions, by default, if subscriber mappings already exist for any specified IP range, those IP ranges will not be updated by the import. However, you can specify that any existing subscriber mappings in the IP range should be removed (*remove-subscriber-mappings* keyword). In this case, the file import will be completely successful.

These commands are LineCard interface commands. Make sure that you are in LineCard Interface command mode, (see *Entering LineCard Interface Configuration Mode* (on page 2-8)) and that the ***SCE***(config if)# prompt appears in the command line.

To import TIRs from a csv file:

Step 1 From the ***SCE***(config if)# prompt, type **subscriber TP-IP-range import csv-file** *csv-file-name* **[remove-subscriber-mappings]** and press **Enter**.

The TIR definitions are imported from the specified *csv* file, and the ***SCE***(config)# prompt appears.

If the **remove-subscriber-mappings** keyword is specified, if subscriber mappings exist for any specified IP range, they will be removed and the command will execute successfully. Otherwise, if subscriber mappings exist for any IP range, those IP ranges will not be updated.

To export TIRs from a csv file:

Step 1 From the ***SCE***(config if)# prompt, type **subscriber TP-IP-range export csv-file** *csv-file-name* and press **Enter**.

The TIR definitions are exported to the specified *csv* file, and the***SCE***(config)# prompt appears.

The **remove-subscriber-mappings** option is not applicable to this command.

Monitoring TIRs

Use these commands to monitor TIRs and subscriber mappings. You can view the following:

- Traffic processor mappings state, including the partitioning between subscriber and TIR mappings, and the utilization of each.
- Configuration of a specified TIR
- Configuration of all TIRs
- All subscriber mappings related to a specified TIR
- Number of subscribers with mappings related to a specified TIR
- Information for a specified subscriber, including assigned TIR, where applicable
- All subscriber mappings in a specified IP range
- Number of subscribers with mappings in a specified IP range

These commands are Privileged Exec commands. Make sure that you are in Privileged Exec command mode by exiting any other modes, and that the ***SCE#*** prompt appears in the command line.

To display traffic processor mappings state:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber TP-mappings statistics`** and press **Enter**.

To display configuration of a specified TIR:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber TP-IP-range name`** *`range-name`* and press **Enter**

To display configuration of all existing TIRs:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber TP-IP-range all`** and press **Enter**.

To display all subscriber mappings related to a specified TIR:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber mapping included-in TP-IP-range name`** *`range-name`* and press **Enter**.

To display the number of subscribers with mappings related to a specified TIR:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber amount mapping included-in TP-IP-range name`** *`range-name`* and press **Enter**.

To display complete subscriber information, including which TIR the subscriber belongs to (if applicable):

Step 1 From the ***SCE***# prompt, type **`show interface linecard 0 subscriber name`** *`name`* and press **Enter**.

To display all subscribers mapped to a specified IP range:

Step 1 From the ***SCE***# prompt, type **`show interface linecard 0 subscriber mapping included-in IP`** *`IP-range`* and press **Enter**.

To display the number of subscribers mapped to a specified IP range:

Step 1 From the ***SCE***# prompt, type **`show interface linecard 0 subscriber amount mapping included-in IP`** *`IP-range`* and press **Enter**.

Subscriber Aging

As explained previously, aging is the automatic removal of a subscriber when no traffic sessions assigned to it have been detected for a certain amount of time. Aging may be enabled or disabled, and the aging timeout period (in minutes) can be specified.

Aging can be configured separately for introduced subscribers and for anonymous subscribers.

Use the following commands to configure and monitor aging.

- `[no] subscriber aging`
- `subscriber aging timeout`
- `show interface linecard 0 subscriber aging`

To enable aging for anonymous group subscribers:

Step 1 From the ***SCE***`(config if)#` prompt, **`subscriber aging anonymous`** and press **Enter**.

To enable aging for introduced subscribers:

Step 1 From the ***SCE***`(config if)#` prompt, **`subscriber aging introduced`** and press **Enter**.

To disable aging for anonymous group subscribers:

Step 1 From the ***SCE***`(config if)#` prompt, **`no subscriber aging anonymous`** and press **Enter**.

To disable aging for introduced subscribers:

Step 1 From the ***SCE***`(config if)#` prompt, **`no subscriber aging introduced`** and press **Enter**.

To set the aging timeout period (in minutes) for anonymous group subscribers:

Step 1 From the ***SCE***`(config if)#` prompt, **`subscriber aging anonymous timeout 'aging-time'`** and press **Enter**.

To set the aging timeout period (in minutes) for introduced subscribers:

Step 1 From the ***SCE***`(config if)#` prompt, **`subscriber aging introduced timeout 'aging-time'`** and press **Enter**.

To display aging for anonymous groups:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 subscriber aging anonymous`** and press **Enter**.

To display aging for anonymous groups:

Step 1 From the ***SCE***# prompt, type **`show interface linecard 0 subscriber aging introduced`** and press **Enter**.

SCE Platform/SM Connection

The user can configure the behavior of the ***SCE*** platform in case of failure of the smartSUB Manager (SM):

- If SM functionality is critical to the operation of the system: configure forced failure of the ***SCE*** platform in the event of any loss of connection with the SM (may be due either to failure of the SM or failure of the connection itself).
- If SM functionality is not critical to the operation of the system: no action needs to be configured.

To configure forced failure of the ***SCE*** platform in case of failure of the SM:

Step 1 From the ***SCE***(config if)# prompt, type `subscriber sm-connection-failure action force-failure` and press **Enter**.

CHAPTER 10

Redundancy and Fail-Over

This chapter presents the fail-over and redundancy capabilities of the ***SCE*** platform. It first defines relevant terminology, as well as pertinent theoretical aspects of the redundancy and fail-over solution. It then explains specific recovery procedures for both single and dual link topologies. It also explains specific update procedures to be used in a cascaded ***SCE*** platform deployments.

When fail over is required in a deployment, a topology with two cascaded ***SCE*** platforms is used. This cascaded solution provides both network link fail over, and fail over of the functionality of the ***SCE*** platform, including updated subscriber state.

Note The information in this chapter applies to the SCE 2000 4xGBE and SCE 2000 4/8xFE platforms only.

This chapter contains the following sections:

- Terminology and Definitions 10-2
- Redundant Topologies 10-2
- Failure Detection 10-3
- Hot Standby and Fail-over 10-4
- Recovery 10-7
- CLI Commands 10-9
- System Upgrades 10-11

Terminology and Definitions

Following is a list of definitions of terms used in the chapter as they apply to the Cisco fail-over solution, which is based on cascaded ***SCE*** platforms.

- **Fail-over**: A situation in which the ***SCE*** platform experiences a problem that makes it impossible for it to provide its normal functionality, and a second ***SCE*** platform device immediately takes over for the failed ***SCE*** platform.
- **Hot standby**: When two ***SCE*** platforms are deployed in a fail over topology, one ***SCE*** platform is active, while the second ***SCE*** platform is in standby, receiving from the active ***SCE*** platform all subscriber state updates and keep alive messages.
- **Primary/Secondary**: The terms *Primary* and *Secondary* refer to the default status of a particular ***SCE*** platform. The Primary SCE Platform is active by default, while the Secondary device is the default standby.

  Note that these defaults apply only when both devices are started together. However, if the primary ***SCE*** platform fails and then recovers, it will not revert to active status, but remains in standby status, while the secondary device remains active.
- **Subscriber state fail-over**: A fail over solution in which subscriber state is saved.

Redundant Topologies

All Cisco SCE platforms include an internal electrical bypass module, which provide the capability of preserving the network link in case the SCE platform fails. The ***SCE*** platform, which can handle two Ethernet links, includes two such bypass modules. However, in some cases, the service provider wishes to preserve the ***SCE*** platform functionality in case of a failure, in addition to preserving the network link.

Cisco provides a unique solution for this scenario, through deploying two cascaded ***SCE*** platforms on these two Ethernet links.

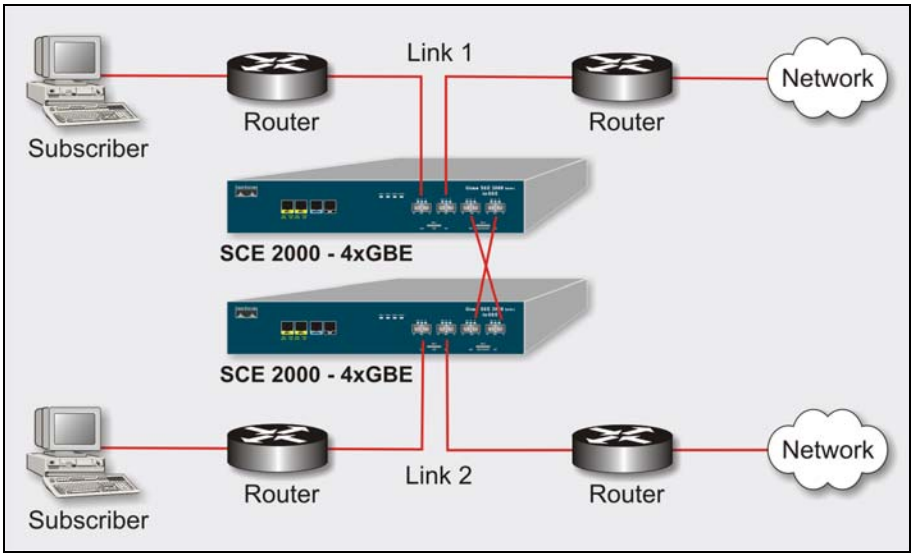
The cascading is implemented by connecting the two ***SCE*** platforms using two of the Ethernet links. This fail over solution applies to both inline and receive-only topologies.

In each ***SCE*** platform, two of the four Ethernet interfaces are connected to each of the network links, while the other two Ethernet interfaces are used for cascading between the ***SCE*** platforms. (See Connecting the Line Ports to the Network for specific cabling procedures for redundant topologies.) The cascade ports are used for transferring network traffic, keep-alive messages and subscriber state updates.

In-line Dual Link Redundant Topology

This topology serves inline deployments where the ***SCE*** platform functionality should be preserved in case of a failure, in addition to preserving the network link.

Figure 10-1: Two SCE Platforms: Dual Link Inline Topology



Failure Detection

The ***SCE*** platform has several types of mechanisms for detecting failures:

- Internal failure detection: The ***SCE*** platform monitors for hardware and software conditions such as overheating and fatal software errors.
- Inter-device failure detection: The ***SCE*** platform sends periodic keep-alive messages via the cascade ports
- ***SCE*** platform-smartSUB Manager (SM) communication failure detection: A failure to communicate with the SM may be regarded as a cause for fail over. However, this communication failure is not necessarily a problem in the ***SCE*** platform. If the connection to the SM of the active ***SCE*** platform has failed, while the connection to the SM of the standby ***SCE*** platform is alive, a fail over process will be initiated to allow the ***SCE*** platform proper exchange of information between the ***SCE*** platforms and the SM.
- Link failure: The system monitors all three types of links for failures:
  - Traffic port link failure: Traffic cannot flow through the ***SCE*** platform.
  - Cascade port link failure: Traffic cannot flow between the ***SCE*** platforms through the cascade ports.
  - Management port link failure: This is not a failure that interrupts traffic on the link in and of itself. However, when SM is used, management port link failure will cause an SM connection failure and this, in turn, will be declared as a failure of the ***SCE*** platform.

This type of failure, in most cases, does not require reboot of the ***SCE*** platform. When the connection with the SM is re-established the ***SCE*** platform is again ready for hot standby. If both ***SCE*** platforms lose their connections with the SM, it is assumed that it is the SM which has failed, thus, no action will be taken in the ***SCE*** platform.

Link Failure Reflection

The ***SCE*** platforms are transparent at Layers 2 and 3. The ***SCE*** platform operates in promiscuous mode, and the network elements on both sides of the ***SCE*** platform, are using the MAC address of the other network element when forwarding traffic.

In order to assist the network elements on both sides of the ***SCE*** platform to identify the link failures as quickly as possible, the ***SCE*** platform supports a functionality of reflecting to the other side of the ***SCE*** platforms events of link failure. When the link on one side of the ***SCE*** platform fails, the corresponding link on the other side is forced down, to reflect the failure.

Link failure reflection is done on the traffic ports. When operating in deployments of single ***SCE*** platform with two Ethernet links, link failure is reflected between the two ports of each link.

When working with two cascaded ***SCE*** platforms, link failure is reflected in two cases:

- Reflection between the traffic ports of each ***SCE*** platform.
- If there is a failure in the cascade port link, the two ***SCE*** platforms can no longer support proper processing of the two links, since the traffic flowing on the standby ***SCE*** platform's link must be forwarded to the active ***SCE*** platform for processing. In this case the link failure is reflected from the cascade ports to the traffic ports of the standby ***SCE*** platform, in order to force the network to switch all the traffic only through the link of the active ***SCE*** platform.

Link failure reflection is supported both when the ***SCE*** platform is operational and when it is in failure/boot status.

Link reflection, like fail-over, is dependent on the bypass mechanism of the ***SCE*** platform.

Hot Standby and Fail-over

The fail over solution requires two ***SCE*** platforms connected in a cascade manner.

Hot Standby

In fail over solution, one of the ***SCE*** platforms is used as the active ***SCE*** platform and the other is used as the standby. Although traffic enters both the active and the standby ***SCE*** platforms, all traffic processing takes place in the ***SCE*** platform which is currently the active one. The active ***SCE*** platform processes the traffic coming on both links, its own link and the link connected to the standby ***SCE*** platform, as follows

- All traffic entering the active ***SCE*** platform through its traffic ports is processed in that ***SCE*** platform and then forwarded to the line.
- All traffic entering the standby ***SCE*** platform through its traffic ports is forwarded through the cascade ports to the active ***SCE*** platform where it is processed, and then returned to the standby ***SCE*** platform through the cascade ports to be forwarded to the original line from which it came.

Since only one ***SCE*** platform processes all traffic at any given time, split flows, which are caused by asymmetrical routing, that exist in the two Ethernet links are handled correctly.

In order to support subscriber-state fail-over, both ***SCE*** platforms hold subscriber states for all parties, and subscriber state updates are exchanged between the active ***SCE*** platform and the standby. This way, if the active ***SCE*** platform fails, the standby ***SCE*** platform is able to start serving the line immediately with a minimum loss of subscriber-state.

The two ***SCE*** platforms also use the cascade channel for exchanging periodic keep-alive messages.

Fail-over

In fail over solution, the two ***SCE*** platforms exchange keep alive messages via the cascade ports. This keep alive mechanism enables fast detection of failures between the ***SCE*** platforms and fast fail over to the standby ***SCE*** platform when required.

If the active ***SCE*** platform fails, the standby ***SCE*** platform then assumes the role of the active ***SCE*** platform.

The failed ***SCE*** platform uses its electrical bypass mechanism, which is a hardware entity that is separate from the main board and processors, to forward traffic to the other ***SCE*** platform, and to forward processed traffic back to the link. The previously standby ***SCE*** platform now processes all the traffic of this other link that is forwarded to it by the previously active ***SCE*** platform in addition to the traffic of its own link.

When the failed ***SCE*** platform recovers, it will remain in standby, while the previously standby ***SCE*** platform remains active. Switching the ***SCE*** platforms back to their original roles may be performed manually, if required, after the failed ***SCE*** platform has either recovered or been replaced.

If the failure is in the standby ***SCE*** platform, it will continue to forward traffic to the active ***SCE*** platform and back to its link, while the active ***SCE*** platform continues to provide its normal processing functionality to the traffic of the two links.

There are two user-configurable options that are relevant in a situation when an ***SCE*** platform fails:

- **Bypass**: Maintain the link in bypass mode (continue sending traffic to the other ***SCE*** platform, and then continue forwarding the processed traffic back to the link). The incoming traffic in the failed ***SCE*** platform is forwarded to the working ***SCE*** platform, where it is processed and then sent back to the original ***SCE*** platform and back to the link.
  - **Effect on the network link**: negligible.
  - **Effect on the *SCE* platform functionality**: The effect on the ***SCE*** platform functionality is dependent on the failed ***SCE*** platform.
    - If the failure is in the standby ***SCE*** platform: the active ***SCE*** platform continues providing its normal functionality, processing the traffic of the two links.
    - If the failure is in the active ***SCE*** platform: the standby ***SCE*** platform takes over processing the traffic, and becomes the active ***SCE*** platform.

    This is the default configuration, and is also the recommended option for most deployments.

- **Cutoff**: Change the link of the failed ***SCE*** platform to cutoff (layer 1) forcing the network to switch all traffic through the line of the working ***SCE*** platform. This will, of course, decrease the network capacity by 50%, but may be useful in some cases.
  - **Effect on the network**: The network looses 50% of its capacity (until the failed ***SCE*** platform has recovered).
  - **Effect on the *SCE* platform functionality**: The effect on the ***SCE*** platform functionality is dependent on the failed ***SCE*** platform:
    - If the failure is in the standby ***SCE*** platform: ***SCE*** platform continues providing its normal functionality, processing the traffic of its own link.
    - If the failure is in the active ***SCE*** platform: the standby ***SCE*** platform takes over processing the traffic, and becomes the active ***SCE*** platform. This option is available for use in special cases, and requires specific configuration.

Failure in the Cascade Connection

The effect of a failure in the cascade connection between the two SCE platforms depends on whether one or both connections fail:

- Only one cascade connection is down: In this case, both SCE platforms can still communicate, so each still knows the status of the peer.

  As long as one cascade connection remains up, the standby will cut off its traffic links so that all traffic is routed via the active SCE platform. Therefore, split flow is avoided, but at the expense of half line capacity.
- Both cascade links are down: In this case, neither SCE platform knows anything about the status of the peer. Each platform then works in standalone mode, which means that each SCE platform processes on its own traffic, only. This results in split flows.

Installing a Cascaded System

This section outlines the installation procedures for a redundant solution with two cascaded ***SCE*** platforms. Refer to the *Cisco Service Control Engine (SCE) CLI Command Reference* for details of the CLI commands.

Warning When working with two ***SCE*** platforms with split-flow and redundancy, it is extremely important to follow this installation procedure.

To install a cascaded system, complete the following steps:

Step 1 Install both ***SCE*** platforms, power them up, and perform the initial system configuration. (See Installation and Maintenance and Connecting the Management Interfaces and Performing Initial System Configuration.)

Step 2 Connect both ***SCE*** platforms to the management station.(See Connecting the Management Interface)

Step 3 Connect the cascade ports. (See Dual Link: Two ***SCE*** platforms Topology.)

Step 4 Set topology configurations for each ***SCE*** platform via the connection-mode options. (See *Topology-Related Parameters for Redundant Topologies* (on page 10-9).)

Step 5 Make sure that the ***SCE*** platforms have synchronized and active ***SCE*** platform was selected.

Use the **`show interface linecard 0 connection-mode`** command.

Step 6 If you want to start with bypass/sniffing, change the link mode to your required mode in both ***SCE*** platforms on both links. The bypass mode will be applied only to the active ***SCE*** platform. (See *Link Mode* (on page 7-2).)

Step 7 Make sure that the link mode is as you required. (See *Monitoring the System* (on page 10-10).)

Use the **`show interface linecard 0 link mode`** command.

Step 8 Connect the traffic port of ***SCE*** platform #1. This will cause a momentary down time until the network elements from both sides of the ***SCE*** platform auto-negotiate with it and start working (when working inline). (See Dual Link: Two ***SCE*** platforms Topology.)

Step 9 Connect the traffic port of ***SCE*** platform #2, this will cause a momentary down time until the network elements from both sides of the ***SCE*** platform auto-negotiate with it and start working (when working inline). (See Dual Link: Two ***SCE*** platforms Topology.)

Step 10 When full control is needed, change the link mode on both ***SCE*** platforms on both links to 'forwarding'. It is recommended to first configure the active ***SCE*** platform and then the standby. (See *Link Mode* (on page 7-2).)

Step 11 You can now start working with the Subscriber Manager.

Recovery

This section specifies the procedure for recovery after a failure. The purpose of the recovery procedure is to restore the system to fully functional status. After the recovery procedure, the behavior of the system is the same as after installation.

A failed ***SCE*** platform may either recover automatically or be replaced (manual recovery). Whether recovery is automatic or manual depends on the original cause of the failure:

- Power failure: manual or automatic recovery can be implemented.
- Any failure resulting in a reboot: manual or automatic recovery can be implemented (this is configurable).
- 3-consecutive reboots within half an hour: manual recovery only
- Cascade ports link-failure: automatic recovery when link revives.
- Traffic link failure: automatic recovery when link revives.
- Failure in the communications with the SM: automatic by SM decisions after connection is re-established.
- Hardware malfunction: manual recovery, after replacing the malfunctioning ***SCE*** platform.

Replacing the SCE platform (manual recovery)

This is done in two stages, first manual installation steps performed by the technician, and then automatic configuration steps performed by the system.

Manual steps:

Step 1 Disconnect the failed ***SCE*** platform from the network.

Step 2 Connect a new ***SCE*** platform to the management link and the cascade links (leave network ports disconnected.)

Step 3 Configure the ***SCE*** platform.

Step 4 Basic network configurations done manually (first time).

Step 5 Load application software (***Service Control Application Suite for Broadband***) to the ***SCE*** platform. Provide application configuration.

Step 6 Connect the traffic ports to the network links.

Automatic steps (in parallel with the manual steps, requires no user intervention):

Step 1 Establishment of inter-***SCE*** platform communication.

Step 2 Synchronization with the SM

Step 3 Copying updated subscriber states from the active***SCE*** platform to the standby.

Reboot only (fully automatic recovery)

Step 1 Reboot of the ***SCE*** platform.

Step 2 Basic network configurations.

Step 3 Establishment of inter-***SCE*** platform communication.

Step 4 Selection of the active ***SCE*** platform.

Step 5 Synchronization of the recovered ***SCE*** platform with the SM.

Step 6 Copying updated subscriber states from the active ***SCE*** platform to the standby.

CLI Commands

This section presents CLI commands relevant to the configuration and monitoring of a redundant system.

Use the following commands to configure and monitor a redundant system:

- `connection-mode`
- `[no] force failure-condition`
- `Show interface linecard ‘number’ connection-mode`
- `Show interface linecard ‘number’ physically-connected-links`

Topology-Related Parameters for Redundant Topologies

All four of the topology-related parameters are required when configuring a redundant topology.

- **Connection mode**: Redundancy is achieved by cascading two ***SCE*** platforms. Therefore the connection mode for both ***SCE*** platforms may be either:
  - Inline-cascade
  - Receive-only-cascade
- **Physically-connected-links**: For each of the cascaded ***SCE*** platforms, this parameter defines the number of the link (Link 0 or Link 1) connected to this ***SCE*** platform.
- **Priority**: For each of the cascaded ***SCE*** platforms, this parameter defines whether it is the primary or secondary device.
- **On-failure**: For each of the cascaded ***SCE*** platforms, this parameter determines whether the system cuts the traffic or bypasses it when the ***SCE*** platform either has failed or is booting.

Configuring the Connection Mode

Use the following command to configure the connection mode, including the following parameters:

- inline/receive only
- physically connected links
- behavior upon failure of the ***SCE*** platform
- primary/secondary

To configure the connection mode:

Step 1 From the ***SCE***`(config if)#` prompt, type **`connection-mode`** *`inline-cascade/receive-only-cascade`* **`[p`**`hysically-connected-links` *`{link-0/link-1}`* **`]`** `[`**`priority`** *`{primary/secondary}`* **`]`** `[`**`on-failure`** *`{bypass/cutoff}`*`]` and press **Enter**.

EXAMPLE 1

Use the following command to configure the primary ***SCE*** platform in a two-***SCE*** platform inline topology. Link 1 is connected to this ***SCE*** platform and the behavior of the ***SCE*** platform if a failure occurs is *bypass*.

```
SCE(config if)# connection-mode inline-cascade physically-connected-links
link-1 priority primary on-failure bypass
```

EXAMPLE 2

Use the following command to configure the ***SCE*** platform that might be cascaded with the ***SCE*** platform in Example 1. This ***SCE*** platform would have to be the secondary ***SCE*** platform, and Link 0 would be connected to this ***SCE*** platform, since Link 1 was connected to the primary. The connection mode would be the same as the first, and the behavior of the ***SCE*** platform if a failure occurs is also bypass.

```
SCE(config if)# connection-mode inline-cascade physically-connected-links
link-0 priority secondary on-failure bypass
```

Forced Failure

Use the following commands to force a virtual failure condition, and to exit from the failure condition when performing an application upgrade. (See *Application Upgrade* (on page 10-12).)

To force a virtual failure condition:

Step 1 From the ***SCE*** `(config if)#` prompt, type **`force failure-condition`** and press **Enter**.

To exit the virtual failure condition:

Step 1 From the ***SCE*** `(config if)#` prompt, type **`no force failure-condition`** and press **Enter**.

Monitoring the System

Use the following commands to view the current connection mode and link mode parameters.

To view the current connection mode:

Step 1 From the ***SCE***`#` prompt, type **`show interface linecard 0 connection-mode`** and press `Enter`.

To view the current link mode:

Step 1 From the ***SCE***# prompt, type **`show interface linecard 0 link mode`** and press `Enter`.

To view the current link mappings:

Step 1 From the ***SCE***# prompt, type **`show interface linecard 0 physically-connected-links`** and press `Enter`.

System Upgrades

In a redundant solution, it is important that firmware and/or application upgrades be performed in such a way that line and service are preserved.

Refer to the following sections for instructions on how to perform these procedures on two cascaded SCE platforms:

- Upgrade the firmware only
- Upgrade the application only
- Upgrade both the firmware and the application at the same time

Firmware Upgrade (package installation)

Step 1 Install package on both ***SCE*** platforms (open the package and copy configuration).

Step 2 Reload the standby ***SCE*** platform.

Step 3 Wait until the standby finishes synchronizing and is ready to work.

Step 4 Make sure that the connection mode configurations are correct.

Step 5 Reload the active ***SCE*** platform.

Step 6 After the former active ***SCE*** platform reboots and is ready to work manually, it may be left as standby or we can manually switch the ***SCE*** platforms to their original state.

Application Upgrade

Step 1 Unload the application in the standby ***SCE*** platform.

Step 2 Load new application to the standby ***SCE*** platform.

Step 3 Make sure that the connection mode configurations are correct.

Step 4 Wait until the standby ***SCE*** platform finishes synchronizing and is ready to work.

Step 5 Force failure condition in the active ***SCE*** platform.

Step 6 Upgrade the application in the former active ***SCE*** platform.

Step 7 Remove the force failure condition in that platform.

Step 8 After the former active ***SCE*** platform recovers and is ready to work, it may remain the standby or can be manually switched back to active.

Simultaneous Upgrade of Firmware and Application

Step 1 In the standby SCE platform:

- Uninstall the application.
- Upgrade the firmware (this includes a reboot).
- Install the new application.

Step 2 Force-failure in the active SCE platform.

This makes the updated SCE platform the active one, and it begins to give the NEW service.

Step 3 Repeat step 1 for the (now) standby SCE platform.

(Since this includes a reboot, it is not necessary to undo the force failure command.)

CHAPTER 11

Identifying And Preventing Distributed-Denial-Of-Service Attacks

This chapter describes the ability of the ***SCE*** platform to identify and prevent DoS and DDoS attacks, and the various procedures for configuring and monitoring the Attack Filter Module.

This chapter contains the following sections:

- Attack Filtering 11-1
- Attack Detection 11-2
- Attack Detection Thresholds 11-3
- Attack Handling 11-3
- Configuring Attack Detectors 11-5
- Configuring Subscriber Notifications 11-11
- Managing Attack Filtering 11-12
- Monitoring Attack Filtering 11-14

Attack Filtering

The ***SCE*** platform includes enhanced capabilities of identifying DoS and DDoS attacks, and protecting against them. Previous versions of the SEos provided a means to monitor the entire link and identify a global increase in flow-open rate, indicative of a DoS attack.

The new SEos that runs on the ***SCE*** platform extends this concept by improving the detection mechanism, adding individual IP address granularity, and providing a set of actions to report (to the operator), block, and notify (the subscriber) of the attack.

The system tracks the following two metrics in an attempt to identify abnormal flow/ connection increase:

- **open-flows**: Total number of flows (TCP, UDP, ICMP, other) that are concurrently open
- **ddos-suspected-flows**: Total number of flows that are possible suspects of being part of a denial- of- service attack because they are un- established (in TCP the 3-way handshake is incomplete, in UDP/ ICMP/ OTHER, less than 3 packets have been transmitted on a flow).

The above two metrics are maintained for each IP address, and the system tracks the values against pre- defined (and user- configurable) thresholds (an attack is defined when the threshold is breached for a certain IP address).

Note that the system makes a distinction between an Attack- Source & Attack-Destination. As each attack is associated with an IP address, the IP- address is classified as either the attack source (i. e. it is generating the attack traffic) or its destination (i.e. it is being attacked). This parameter is later reported, and can also be used in creating filtering and action rules for the DoS mechanism.

Once an attack is identified, the system can be instructed to perform any of the following actions:

- **Report**: The system will generate an SNMP trap each time an attack ‘starts’ and ‘stops’. The SNMP trap contains the following information fields:
  - A specific IP address
  - **Protocol** (TCP, UDP, ICMP or Other)
  - **Interface** (User/Network) behind which the detected IP address is found. This is referred to below as the attack ‘side’
  - **Attack direction** (whether the IP address is the attack source or the attack destination).
  - **Type of threshold breached** (open- flows / ddos- suspected- flows) [‘ attack- start’ traps only]
  - **Threshold value breached** [‘ attack- start’ traps only]
  - **Action taken** (report, block) indicating what was the action taken by the ***SCE*** platform in response to the detection
  - **Amount of attack flows blocked/ reported** providing the total number of flows blocked by the protection mechanism during the attack [‘ attack- stop’ traps only]
- **Block**: The system will block all suspected traffic from / to the attack IP address (depending on whether the IP address is an Attack- Source or Attack-Destination)
- **Subscriber notification**: When the IP address identified is mapped to a particular subscriber context, the system can be configured to notify the subscriber of the fact that he is under an attack (or a machine in his network is generating such an attack), using HTTP Redirect.

Attack Detection

The attack interface, protocol and specific IP address are detected. When one specific IP address is attacking a different specific IP address, two separate attack detections should be identified, one for the attacking host and one for the attacked host. The system can identify a maximum of 1000 independent, simultaneous attacks.

Attack detections are identified using the following parameters:

- A specific IP address
- Protocol (TCP, UDP, ICMP or Other)
- Interface (User / Network) behind which the detected IP address is found.
- This is referred to below as the attack side.
- Attack direction (whether the IP address is the attack source or the attack destination address).

Attack detection and handling are user-configurable. The remainder of this chapter explains how to configure and monitor attack detection.

Attack Detection Thresholds

There are two counters that are used for attack detection. These counters are maintained by the SCE Platform for each IP address, protocol, interface and attack-direction.

- **Concurrently open flows**: The number of flows that have been opened and have not yet been closed by TCP FIN or by aging.
- **DDoS-suspected open flows**: The definition of a DDoS-suspected open flow varies according to the protocol:
  - TCP flows: A flow for which the first payload packet has not been detected. (Also called un-established.)
  - All other flows: A flow for which less than three packets have been detected.

Note that every flow begins life in the ***SCE*** platformas a DDoS-suspected flow, and stops being DDoS-suspected when the system determines that it is carrying a real TCP connection due or that its length identifies it as a normal flow. When observing traffic related to a specific IP address, it is expected that under normal conditions there will be not many DDoS-suspected flows, even though there might be a lot of concurrently open flows.

The system has a separate default threshold for the number of concurrently open flows and DDoS-suspected open flows. If either threshold is crossed for a particular IP address/interface combination, an attack is declared for that IP address. When the number of flows decreases and the threshold is crossed in the opposite direction for more than three seconds, the system declares that the attack has ended.

The user may define values for these thresholds that override the preset defaults. It is also possible to configure specific thresholds for certain conditions (per IP range, protocol, interface and attack direction). This enables the user to set different detection criteria for different types of network entities, such as a server farm, DNS server, or large enterprise customer.

Attack Handling

Attack handling can be configured as follows:

- **Configuring the action**:
  - Report: Attack packets are processed as usual, and the occurrence of the attack is reported.
  - Block: Attack packets are dropped by the SE200, and therefore do not reach their destination.

  Regardless of which action is configured, two reports are generated for every attack: one when the start of an attack is detected, and one when the end of an attack is detected.

  Attack start and end are defined as follows:

  - Attack start: Reported as soon as the threshold value for concurrent open-flows or DDoS-suspected flows is exceeded.

- Attack end: Reported when both the number of concurrent open-flows and the number of DDoS-suspected flows are below the threshold value for at least 3 seconds

- **Configuring subscriber-notification**:
  - Enabled: If the subscriber IP address is detected to be attacked or attacking, the subscriber is notified about the attack.
  - Disabled: The subscriber is not notified about the attack.

Subscriber Notification

When an attack is identified, if the IP address is detected on the subscriber side and is mapped to a subscriber, the system notifies the application about the attack. This enables the application to notify the subscriber about the attack on-line by redirecting HTTP requests of this subscriber to a server that will notify it of the attack.

In addition, when blocking TCP traffic, the system can be configured to not block certain ports in order to make this redirection possible. A list of up to three port numbers can be configured to be *un-blockable*.

Note that subscriber-notification can only function if supported by the Service Control Application currently loaded to the SCE Platform, and the application is configured to activate this capability. To verify whether the application you are using supports attack subscriber notification, and for details about enabling attack subscriber notification in the application, please refer to the documentation of the relevant Service Control Application**.**

Configuring Attack Detectors

The Cisco attack detection mechanism is controlled by defining and configuring special entities called Attack Detectors.

There is one attack detector called ‘default’, which is always enabled, and 99 attack detectors (numbered 1-99), which are disabled by default. Each detector (both the default and detectors 1-99) can be configured with a separate action and threshold values for all possible combinations of protocol, direction and side.

When detectors 1-99 are disabled, the default attack detector configuration determines the thresholds used for detecting an attack, and the action taken by the SCE Platform when an attack is taken. For each combination of protocol (TCP/UDP/ICMP/Other), attack-direction (source/destination) and side (Network/Subscriber), a different set of thresholds and action can be set. In addition, subscriber-notification can be enabled or disabled in the same granularity.

The default attack detector should be configured with values that reflect the desired SCE platform behavior for the majority of the traffic flows flowing through it. However, it is not feasible to use the same set of values for all the traffic that traverses through the ***SCE*** platform, since there might be some network entities for which the characteristics of their normal traffic should be considered as an attack when coming from most other network elements. Here are two common examples:

- A DNS server is expected to be the target of many short DNS queries. These queries are typically UDP flows, each flow consisting of two packets: The request and the response. Normally, the SCE considers all UDP flows that are opened to the DNS server as DDoS-suspected flows, since these flows include less than 3 packets. A DNS server might serve hundreds of DNS requests at peak times, and so the system should be configured with a suitable threshold for DDos-suspected flows for *protocol = UDP* and *direction = attack-destination*. A threshold value of 1000 would probably be suitable for the DNS server. However, this threshold would be unsuitable for almost all other network elements, since, for them, being the destination of such large number of UDP flows would be considered an attack. Therefore setting a threshold of 1000 for all traffic is not a good solution.
- The subscriber side of the ***SCE*** platform might contain many residential subscribers, each having several computers connected through an Internet connection, and each computer having a different IP address. In addition, there might be a few business subscribers, each using a NAT that hides hundreds of computers behind a single IP address. Clearly, the traffic seen for an IP address of a business subscriber contains significantly more flows than the traffic of an IP address belonging to a residential subscriber. The same threshold cannot be adequate in both cases.

To let the SCE Platform treat such special cases differently, the user can configure non-default attack detectors in the range of 1-99. Like the default attack detector, non-default attack detectors can be configured with different sets of values of action and thresholds for every combination of Protocol, attack direction and side. However, in order to be effective, a non-default attack detector must be enabled and must be assigned an ACL (access control list). The action and thresholds configured for such attack detector are effective only for IP addresses permitted by the ACL. Non-default attack-detectors can be assigned a label for describing their purpose, such as 'DNS servers' or 'Server farm'.

Non-default attack detectors are effective only for combinations of protocol, attack direction and sides that have been specifically configured. This eliminates the need to duplicate the default attack detector configuration into the configuration non-default attack detectors, and is best illustrated with an example: Suppose an HTTP server on the subscriber side of the ***SCE*** platform is getting many requests, which requires the use of a non-default attack detector for configuring high threshold values for incoming TCP flows. Assume attack detector number 4 is used for this purpose; hence it is enabled, and assigned an ACL which permits the IP address of the HTTP server. Also suppose that it is desirable to protect subscribers from UDP attacks, hence the default attack detector is configured to block UDP attacks coming from the network (The default configuration is only to report attacks, not block them). If the HTTP server is attacked by a UDP attack from the network , the configuration of the default attack detector will hold for this HTTP server as well, since attack detector number 4 was not configured for UDP attacks.

For each possible combination of protocol, attack direction, and side, the set of enabled attack detectors, together with the default attack detector, forms a database used to determine the threshold and action to take when an attack is detected. When the platform detects a possible attack, it uses the following algorithm to determine the thresholds for attack detection.

- Enabled attack detectors are scanned from low to high numbers.
- If the IP address is permitted by the ACL specified by the attack detector, and a threshold is configured for this combination of protocol, direction and side, then the threshold value specified by this attack detector are used. If not, the scan continues to the next attack detector.
- If no attack detector matches the IP address/protocol combination, then the values of the default attack detector are used.

The same logic is applied when deciding what action the platform should take in handling the attack. The action that is used, is the one specified by the lowest-numbered enabled attack detector that has a specific action setting for the attack protocol, direction and side is used. If none exists, the configuration of the default attack detector is used.

Use the following commands to configure and enable attack detection:

- `[no] attack-filter`
- `attack-detector (default|<number>) protocol <protocol> attack-direction <direction> side <side> action <action> [open-flows <number> ddos-suspected-flows <number>]`
- `attack-detector (default|<number>) protocol <protocol> attack-direction <direction> side <side> (notify-subscriber|dont-notify-subscriber)`
- `default attack-detector (default|<number>) protocol <protocol> attack-direction <direction> side <side>`
- `attack-detector <number> access-list comment`

- `[no] attack-filter subscriber-notification ports`
- `no attack-detector <number>`

Note All the above CLI commands are line interface configuration commands. You must enter line interface configuration mode and see the ***SCE***`(config if)#` prompt displayed.

Enabling Specific-IP Detection

By default, specific-IP detection is disabled, however the user may enable it.

To disable Specific-IP Detection:

Step 1 From the ***SCE***`(config if)#` prompt, type **`no attack-filter`** and press **Enter**.

To enable Specific-IP Detection:

Step 1 From the ***SCE***`(config if)#` prompt, type **`attack-filter`** and press **Enter**.

Default Attack Detector

Use these commands to define default thresholds and attack handling action. If a specific attack detector is defined for a particular situation (protocol/attack direction/side), it will override these defaults. The default values configured for the default attack detector are:

- Default action: Report
- Default TCP thresholds:
  - Concurrently open flows: 10000
  - DDoS-suspected flows: 2000
- Default UDP thresholds:
  - Concurrently open flows: 10000
  - DDoS-suspected flows: 5000
- Default ICMP/Other flows:
  - Concurrently open flows: 1000
  - DDoS-suspected flows: 500
- Subscriber notification: Disabled

To define action and optionally the default thresholds:

Step 1 From the ***SCE***(config if)# prompt, type **attack-detector default protocol (***TCP*|*UDP*|*ICMP*|*other***) attack-direction (***attack-source*|*attack-destination*|*both***) side (***subscriber*|*network*|*both***) action (***report*|*block***) [open-flows <***number***> ddos-suspected-flows <***number***>]** and press **Enter**.

Use the following command to set the default values for the subscriber notification mechanism.

Step 2 From the ***SCE***(config if)# prompt, type **attack-detector default protocol** (*TCP*|*UDP*|*ICMP*|*other*) a**ttack-direction** (*attack-source*|*attack-destination*|*both*) **side** *(subscriber*|*network*|*both) (notify-subscriber*|*dont-notify-subscriber)* and press **Enter**.

Use the following command delete user-defined default values for action, thresholds and subscriber notification for a given combination of protocol, direction and side, and reinstate the system defaults.

To delete user-defined defaults for a specific situation:

Step 1 From the ***SCE***(config if)# prompt, type **no attack-detector default protocol (***TCP*|*UDP*|*ICMP*|*other***) attack-direction (***attack-source*|*attack-destination*|*both***) side (***subscriber*|*network*|*both***)** and press **Enter**.

Specific Attack Detectors

A specific attack detector may be configured for each possible combination of protocol direction, and side. The ***SCE*** platform supports a maximum of 100 attack detectors. Each attack detector is identified by a number (1-99). Each detector can be either disabled (default) or enabled. An enabled attack detector must be configured with the following parameters:

- Access-Control List (ACL) number: Identifies the IP addresses selected by this detector. (See *Access Control Lists* ("Configuring Access Control Lists (ACLs)" on page 5-1).)
- Comment: For documentation purposes

In addition, an enabled attack detector may contain the following settings:

- Threshold values for number of concurrently open flows and for number of DDoS-suspected flows
- Action to take when an attack is detected (Report or Block)
- Subscriber notification setting (Enabled or Disabled)

Use these commands to define thresholds, actions, and subscriber notification setting for a specific attack detector for a particular situation (protocol/attack direction/side).

To enable a specific attack detector and assign and it an ACL:

Step 1 From the ***SCE***(config if)# prompt, type **attack-detector** <*number*> **access-list** <*number*> **comment** <*comment*> and press **Enter**.

To disable a specific attack detector:

Step 1 From the ***SCE***(config if)# prompt, type **no attack-detector** <*number*> and press **Enter**.

To disable all non-default attack detectors:

Step 1 From the ***SCE***(config if)# prompt, type **no attack-detector** *all-numbered* and press **Enter**.

To define action and optionally thresholds for a specific attack detector:

Step 1 From the ***SCE***(config if)# prompt, type **attack-detector** <*number*> **protocol (***TCP*|*UDP*|*ICMP*|*other***) attack-direction (***attack-source*|*attack-destination*|*both***) side (***subscriber*|*network*|*both***) action (***report*|*block***) [open-flows** <*number*> **ddos-suspected-flows** <*number*>**]** and press **Enter**.

Use the following command to set the subscriber notification setting for a given attack detector and a given combination of protocol, direction and side.

To define the subscriber notification setting for a specific attack detector:

Step 1 From the ***SCE***(config if)# prompt, type **attack-detector** <*number*> **protocol (***TCP*|*UDP*|*ICMP*|*other***) attack-direction (***attack-source*|*attack-destination*|*both***) side (***subscriber*|*network*|*both***) (***notify-subscriber*|*dont-notify-subscriber***)** and press **Enter**.

Use the following command to remove settings of action, thresholds and subscriber notification for a specific attack detector and combination of protocol, direction and side.

Use the following command to remove the specific user-defined default values for this attack detector and reinstate the default values.

To delete user-defined values for a specific situation:

Step 1 From the ***SCE***(config if)# prompt, type **default attack-detector** *<number>* **protocol (***TCP*|*UDP*|*ICMP*|*other***) attack-direction (***attack-source*|*attack-destination*|*both***) side (***subscriber*|*network*|*both***) (***notify-subscriber*|*dont-notify-subscriber***)** and press **Enter**.

Sample Attack Detector Configuration

The following configuration changes the default user threshold values used for detecting ICMP attacks, and configures an attack-detector with high thresholds for UDP attacks, preventing false detections of two DNS servers (10.1.1.10 and 10.1.1.13) as being attacked.

```
   (First enter the linecard interface configuration mode)
SCE(config)# interface linecard 0

   (Configure the default ICMP threshold and action.)
SCE(config if)# attack-detector default protocol ICMP attack-direction
attack-source action report open-flows 100 ddos-suspected-flows 100

   (Enable attack detector #1 and assign ACL #3 to it.)
SCE(config if)# attack-detector 1 access-list 3 comment "DNS servers"

   (Define the thresholds and action for attack detector #1)
SCE(config if)# attack-detector 1 protocol UDP attack-direction attack-
destination action report open-flows 1000000 ddos-suspected-flows 1000000

   (Enable subscriber notification for attack detector #1)
SCE(config if)# attack-detector 1 protocol UDP attack-direction attack-
destination side subscriber notify-subscriber

   (Exit the linecard interface configuration mode)
SCE(config if)# exit

   (Define the ACL)
SCE(config)# access-list 3 permit 10.1.1.10
SCE(config)# access-list 3 permit 10.1.1.13
```

Configuring Subscriber Notifications

Subscriber notification is a capability used- for notifying a subscriber in real-time about current attacks involving IP addresses mapped to that subscriber. Subscriber notification is configured on a per-attack-detector level, as explained above, and must also be enabled and configured by the application loaded to the ***SCE*** platform, as explained in the appropriate Service Control Application user guide**.**

In the current solutions, the SCE Platform notifies the subscriber about the attack by redirecting HTTP flows originating from the subscriber to the service provider's server, that should notify the subscriber that he is under attack. This raises a question regarding TCP attacks originating from the subscriber that are configured with *block* action. Such attacks cannot normally be notified to the subscriber using HTTP redirection, since all HTTP flows originating from the subscriber are TCP flows, and they are therefore blocked along with all other attack flows. In order to enable effective use of HTTP redirect, there is a CLI command that prevents blocking of TCP flows originating from the subscriber to specified TCP ports, even when the above scenario occurs.

Subscriber Notification Ports

Up to three ports can be specified as subscriber notification ports. The attack filter will, never block TCP Traffic from the subscriber side of the ***SCE*** platform to these ports, leaving them always available for subscriber notification.

To add ports to the list of subscriber notification ports:

Step 1 From the ***SCE*** `(config if)#` prompt, type **`attack-filter subscriber-notification ports`** *`<port1> [<port2> [<port3>]]`* and press **Enter**.

To remove all ports from the list of subscriber notification ports:

Step 1 From the ***SCE*** `(config if)#` prompt, type **`no attack-filter subscriber-notification ports`** and press **Enter**.

Managing Attack Filtering

After configuring the attack detectors, the SCE Platform automatically detects attacks and handles them according to the configuration. However, there are scenarios in which a manual intervention is desired, either for debug purposes, or because it is not trivial to reconfigure the SCE attack-detectors properly. For example:

- The SCE Platform has detected an attack, but the user knows this to be a false alarm. The proper action that should be taken by the user is to configure the system with higher thresholds (for the whole IP range, or maybe for specific IP addresses). However, this might take time, and, if attack handling is specified as 'Block', the user may wish to stop the block action for this specific attack quickly, leaving the configuration changes for a future time when there is time to plan the needed changes properly.

  Use the `dont-filter` command described below for this type of case.

- An ISP is informed that one of his subscribers is being attacked by a UDP attack from the network side. The ISP wants to protect the subscriber from this attack by blocking all UDP traffic to the subscriber, but unfortunately the SCE Platform did not recognize the attack. (Alternatively, it could be that the attack was recognized, but the configured action was 'report' and not 'block').

  Use the `force-filter` command described below for this type of case.

The user can use the CLI attack filtering commands to do the following:

- Prevent/stop filtering of an attack related to a specified IP address
- Force filtering of an attack related to a specified IP address

Use the following commands to either force or prevent attack filtering:

- `attack-filter slot 0 dont-filter`
- `attack-filter slot 0 force-filter`
- `no attack-filter slot 0 dont-filter all`
- `no attack-filter slot 0 force-filter all`

Note All the above CLI commands are privileged exec commands. If in line interface configuration mode, you must exit to the privileged exec mode and see the ***SCE#*** prompt displayed

Preventing Attack Filtering

Attack filtering can be prevented for a specified IP address/protocol by executing a **dont-filter** CLI command. If filtering is already in process, it will be stopped. When attack filtering has been stopped, it remains stopped until explicitly restored by another CLI command (either **force-filter** or **no dont-filter**).

To prevent attack filtering for the specified situation:

Step 1 From the ***SCE#*** prompt, type **attack-filter slot 0 dont-filter ip <***IP-address***> protocol (***TCP|UDP|ICMP|other***) attack-direction (***attack-source|attack-destination|both***) side (***subscriber|network|both***)** and press **Enter**.

To restore automatic attack filtering for the specified situation:

Step 1 From the ***SCE#*** prompt, type **no attack-filter slot 0 dont-filter ip <***IP-address***> protocol (***TCP|UDP|ICMP|other***) attack-direction (***attack-source|attack-destination|both***) side (***subscriber|network|both***)** and press **Enter**.

To restore all stopped attack filtering:

Step 1 From the ***SCE#*** prompt, type **no attack-filter slot 0 dont-filter all** and press **Enter**.

Forcing Attack Filtering

Attack filtering can be forced for a specified IP address/protocol. If filtering is already in process, it will be stopped. Forced attack filtering will continue until undone by an explicit CLI command (either **no force-filter** or **dont-filter**).

To force attack filtering for the specified situation:

Step 1 From the ***SCE#*** prompt, type **attack-filter slot 0 force-filter action (report|block) ip <**IP-address**> protocol (**TCP|UDP|ICMP|other**) attack-direction (**attack-source|attack-destination|both**) side (**subscriber|network|both**)** [notify-subscriber] and press **Enter**.

To undo forced attack filtering for the specified situation:

Step 1 From the ***SCE#*** prompt, type **`no attack-filter slot 0 force-filter ip <`**`IP-address`**`> protocol (`**`TCP|UDP|ICMP|other`**`) attack-direction (`**`attack-source|attack-destination|both`**`) side (`**`subscriber|network|both`**`)`** and press **Enter**.

To undo all forced attack filtering:

Step 1 From the ***SCE#*** prompt, type **`no attack-filter slot 0 force-filter all`** and press **Enter**.

Monitoring Attack Filtering

Use these commands to monitor attack detection and filtering:

- `show interface linecard 0 attack-detector`
- `show interface linecard 0 attack-filter`
- `show interface linecard 0 attack-filter query`
- `show interface linecard 0 attack-filter current-attacks`
- `show interface linecard 0 attack-filter dont-filter`
- `show interface linecard 0 attack-filter force-filter`
- `show interface linecard 0 attack-filter subscriber-notification ports`

Note All the above CLI commands are privikeged exec commands. If in line interface configuration mode, you must exit to the privileged exec mode and see the ***SCE#*** prompt displayed

To display a specified attack detector configuration:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 attack-detector <number>`** and press **Enter**.

To display the default attack detector configuration:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 attack-detector default`** and press **Enter**.

To display all attack detector configurations:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 attack-detector all`** and press **Enter**.

To display the configured threshold values and action for the attack detector for a specified IP address:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 attack-filter query IP-address <IP-address> configured`** and press **Enter**.

To display the current counters for the attack detector for all protocols, attack directions, and sides for a specified IP address:

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 attack-filter query IP-address <IP-address> counters`** and press **Enter**.

To display all currently handled attacks

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 attack-filter current-attacks`** and press **Enter.**

To display all existing forced attack filters

Step 1 From the ***SCE#*** prompt, type **`show interface linecard 0 attack-filter force-filter`** and press **Enter.**

To display all existing stopped attack filters

Step 1 From the ***SCE***# prompt, type **`show interface linecard 0 attack-filter dont-filter`** and press **Enter.**

To display the list of ports selected for subscriber notification

Step 1 From the ***SCE***# prompt, type **`show interface linecard 0 attack-filter subscriber-notification ports`** and press **Enter**.

APPENDIX A

Proprietary MIB Reference

This appendix describes the SCE proprietary MIB supported by the ***SCE*** platform. A MIB (Management Information Base) is a database of objects that can be monitored by a network management system (NMS). The Service Control Platform supports both the standard MIB-II and a proprietary Service Control Enterprise MIB. This proprietary **pcube** MIB enables the external management system to perform configuration, performance, troubleshooting and alerting operations specific to the SCE Platform, and therefore not provided by the standard MIB.

Service Control Enterprise MIB

The Service Control Enterprise MIB splits into four main groups: Products, Modules, Management, and Workgroup. The Service Control enterprise tree structure is defined in a MIB file named `Pcube.mib`.

- The `pcubeProducts` sub-tree contains the sysObjectIDs of the Service Control products.

  Service Control product sysObjectIDs are defined in a MIB file named `Pcube-Products-MIB`

- The `pcubeModules` sub-tree provides a root object identifier from which MIB modules can be defined.
- The `pcubeMgmt` sub-tree contains the configuration copy MIB. (See *pcubeMgmt: pcubeConfigCopyMIB* (on page A-2).)
- The `pcubeWorkgroup` sub-tree contains the SCE MIB, which is the main MIB for the Service Control OS products. (See *pcubeWorkgroup* ("pcubeWorkgroup: pcubeSeMIB" on page A-4).)

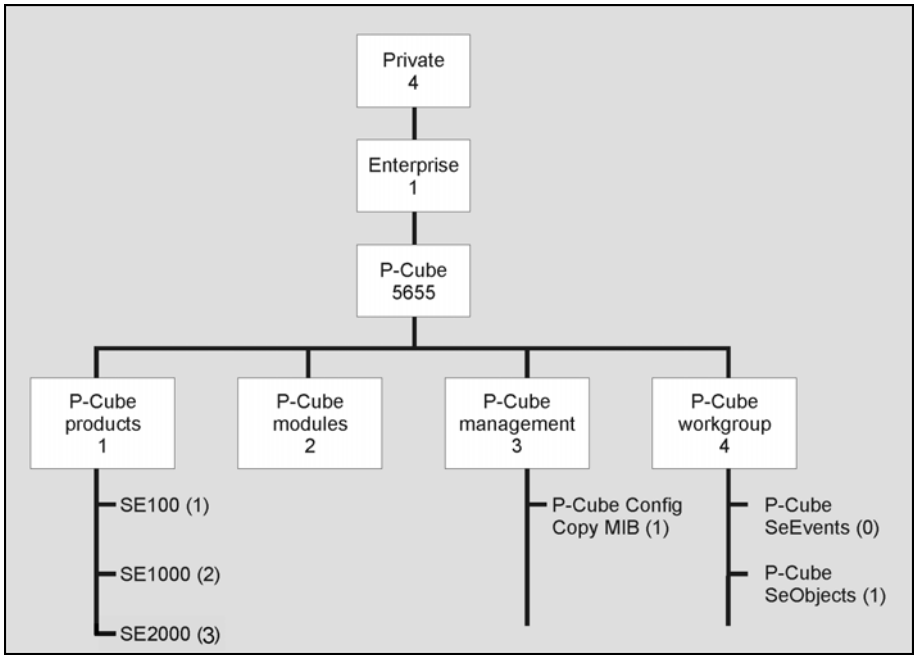
  The SCE MIB is divided into two main groups:

  - **pcubeSeEvents**
  - **pcubeSEObjs**

Note The following object identifier represents the Service Control Enterprise MIB: `1.3.6.1.4.1.5655`, or `iso.org.dod.internet.private.enterprise.pcube`.

The figure below, illustrates the Service Control Enterprise MIB structure.

Figure A-1: Service Control MIB Structure



Currently, the proprietary **pcube** MIB consists of two main sub-trees:

- The *pcubeMgmt* sub-tree: the *pcubeConfigCopyMib* enables saving the running configuration of Cisco products.
- The *pcubeWorkgroup* sub-tree: the *pcubeSeMib* provides a wide variety of configuration and runtime statistics.

Using this Reference

This reference is divided into sections according to the MIB object groups. For each object, the following information is presented:

DESCRIPTION Description of the object, including format and legal values, if applicable.

ACCESS Access control associated with the object:

- Read only (**RO**)
- Read/Write (**RW**)

SYNTAX The general format of the object.

pcubeMgmt: pcubeConfigCopyMIB

The configuration copy MIB supports only local copying of the running configuration to the startup configuration in order to save the current running configuration.

Cisco configuration copy is defined in a file called *Pcube-Config-Copy-mib.mib*.

Config-Copy MIB Objects

Following is a list of the Config-Copy MIB objects.

PcubeCopyIndex {pcubeCopyEntry 1}

PcubeCopyEntryRowStatus {pcubeCopyEntry 2}

pcubeCopySourceFileType {pcubeCopyEntry 3}

pcubeCopyDestFileType {pcubeCopyEntry 4}

pcubeCopyIndex (pcubeCopyEntry 1)

Table index for multi asynchronous copy commands.

As the MIB does not support multiple commands in this release, the value of this index must be "1".

Access RW

SYNTAX

INTEGER: *(1)*

pcubeCopyEntryRowStatus (pcubeCopyEntry 2)

Triggers the actual copy operation.

The value must be *"createAndGo".*

Access RW

SYNTAX

DisplayString*: (createAndGo)*

pcubeCopySourceFileType (pcubeCopyEntry 3)

The source file type.

The value must be "*runningConfig".*

Access RW

SYNTAX

ConfigFileType: *(runningConfig(2))*

pcubeCopyDestFileType (pcubeCopyEntry 4)

The destination file type.

The value must be *"startupConfig".*

Access RW

SYNTAX

ConfigFileType*: (startupConfig(1))*

pcubeWorkgroup: pcubeSeMIB

The pcubeSeMIB is the main MIB for the Cisco Service Control products such as the SCE 2000 and the SCE 1000. This MIB provides configuration and runtime status for chassis, control modules, and line modules on the Cisco Service Control systems.

pcubeSeMIB is defined in a file called *Pcube-Se-mib.mib*.

The **pcubeSeMIB** is divided into two main objects:

- **pcubeSeEvents** (**pcubeWorkgroup 0**)
- **pcubeSEObjs** (**pcubeWorkgroup 1**)

pcubeSeEvents (pcubeWorkgroup 0)

The SCE events are traps for critical asynchronous events.

SCE Events

Following is a list of the SCE events:

operationalStatusOperationalTrap	*{pcubeSeEvents 1}*
operationalStatusWarningTrap	*{pcubeSeEvents 2}*
operationalStatusFailureTrap	*{pcubeSeEvents 3}*
systemResetTrap	*{pcubeSeEvents 4}*
chassisTempAlarmOnTrap	*{pcubeSeEvents 5}*
chassisTempAlarmOffTrap	*{pcubeSeEvents 6}*
chassisVoltageAlarmOnTrap	*{pcubeSeEvents 7}*
chassisFansAlarmOnTrap	*{pcubeSeEvents 8}*
chassisPowerSupplyAlarmOnTrap	*{pcubeSeEvents 9}*
rdrActiveConnectionTrap	*{pcubeSeEvents 10}*
rdrNoActiveConnectionTrap	*{pcubeSeEvents 11}*
rdrConnectionUpTrap	*{pcubeSeEvents 12}*
rdrConnectionDownTrap	*{pcubeSeEvents 13}*
loggerUserLogIsFullTrap	*{pcubeSeEvents 18}*
sntpClockDriftWarnTrap	*{pcubeSeEvents 19}*
linkModeBypassTrap	*{pcubeSeEvents 20}*
linkModeForwardingTrap	*{pcubeSeEvents 21}*
linkModeCutoffTrap	*{pcubeSeEvents 22}*
pcubeSeEventGenericString1	*{cubeSeEvents 23}*
pcubeSeEventGenericString2	*{pcubeSeEvents 24}*
moduleAttackFilterActivatedTrap	*{pcubeSeEvents 25}*
moduleAttackFilterDeactivatedTrap	*{pcubeSeEvents 26}*

moduleEmAgentGenericTrap	*{pcubeSeEvents 27}*
linkModeSniffingTrap	*{pcubeSeEvents 28}*
moduleRedundancyReadyTrap	*{pcubeSeEvents 29}*
moduleRedundantConfigurationMismatchTrap	*{pcubeSeEvents 30}*
moduleLostRedundancyTrap	*{pcubeSeEvents 31}*
moduleSmConnectionDownTrap	*{pcubeSeEvents 32}*
moduleSmConnectionUpTrap	*{pcubeSeEvents 33}*
moduleOperStatusChangeTrap	*{pcubeSeEvents 34}*
portOperStatusChangeTrap	*{pcubeSeEvents 35}*
chassisLineFeedAlarmOnTrap	*{pcubeSeEvents 36}*
rdrFormatterCategoryDiscardingReportsTrap	*{pcubeSeEvents 37}*
rdrFormatterCategoryStoppedDiscardingReports Trap	*{pcubeSeEvents 38}*
sessionStartedTrap	*{pcubeSeEvents 39}*
sessionEndedTrap	*{pcubeSeEvents 40}*
sessionDeniedAccessTrap	*{pcubeSeEvents 41}*
sessionBadLoginTrap	*{pcubeSeEvents 42}*

pcubeSEObjs (pcubeWorkgroup 1)

The SCE objects provide configuration and runtime status for the SCE Platform.

SCE-MIB Objects

Following is a list of the SCE-MIB objects. Each object consists of a number of subordinate, object types, as summarized in the next section.

systemGrp	*{pcubeSEObjs 1}*
chassisGrp	*{pcubeSEObjs 2}*
moduleGrp	*{pcubeSEObjs 3}*
linkGrp	*{pcubeSEObjs 4}*
diskGrp	*{pcubeSEObjs 5}*
rdrFormatterGrp	*{pcubeSEObjs 6}*
loggerGrp	*{pcubeSEObjs 7}*
subscribersGrp	*{pcubeSEObjs 8}*
trafficProcessorGrp	*{pcubeSEObjs 9}*
portGrp	*{pcubeSEObjs 10}*
txQueuesGrp	*{pcubeSEObjs 11}*
globalControllersGrp	*{pcubeSEObjs 12}*
applicationGrp	*{pcubeSEObjs 13}*
trafficCountersGrp	*{pcubeSEObjs 14}*
attackGrp	*{pcubeSEObjs 15}*

SCE-MIB Structure

Following is a summary of the structure of the SCE-MIB. Note the table structure for objects that may have multiple entries, such as the RDR destination, or traffic processors.

`systemGrp`

sysOperationalStatus

sysFailureRecovery

sysVersion

`chassisGrp`

chassisSysType

chassisPowerSupplyAlarm

chassisFansAlarm

chassisTempAlarm

chassisVoltageAlarm

chassisNumSlots

chassisSlotConfig

chassisPsuType

chassisLineFeedAlarm

`moduleGrp`

moduleTable

moduleEntry

moduleIndex

moduleType

moduleNumTrafficProcessors

moduleSlotNum

moduleHwVersion

moduleNumPorts

moduleNumLinks

moduleConnectionMode

moduleSerialNumber

moduleUpStreamAttackFilteringTime

moduleUpStreamLastAttackFilteringTime

moduleDownStreamAttackFilteringTime

moduleDownStreamLastAttackFilteringTime

moduleAttackObjectsClearTime

moduleAdminStatus

moduleOperStatus

`linkGrp`

linkTable

linkEntry

linkModuleIndex

linkIndex

linkAdminModeOnActive

linkAdminModeOnFailure

linkOperMode

linkStatusReflectionEnable

linkSubscriberSidePortIndex

linkNetworkSidePortIndex

`diskGrp`

diskNumUsedBytes

diskNumFreeBytes

`rdrFormatterGrp`

rdrFormatterEnable

rdrFormatterDestTable

rdrFormatterDestEntry

rdrFormatterDestIPAddr

rdrFormatterDestPort

rdrFormatterDestPriority

rdrFormatterDestStatus

rdrFormatterDestConnectionStatus

rdrFormatterDestNumReportsSent

rdrFormatterDestNumReportsDiscarded

rdrFormatterDestReportRate

rdrFormatterDestReportRatePeak

rdrFormatterDestReportRatePeakTime

rdrFormatterNumReportsSent

rdrFormatterNumReportsDiscarded

rdrFormatterClearCountersTime

rdrFormatterReportRate

rdrFormatterReportRatePeak

rdrFormatterReportRatePeakTime

rdrFormatterProtocol

rdrFormatterForwardingMode

rdrFormatterCategoryTable

rdrFormatterCategoryEntry

rdrFormatterCategoryIndex

rdrFormatterCategoryName

rdrFormatterCategoryNumReportsSent

rdrFormatterCategoryNumReportsDiscarded

rdrFormatterCategoryReportRate

rdrFormatterCategoryReportRatePeak

rdrFormatterCategoryReportRatePeakTime

rdrFormatterCategoryNumReportsQueued

rdrFormatterCategoryDestTable

rdrFormatterCategoryDestEntry

rdrFormatterCategoryDestPriority

rdrFormatterCategoryDestStatus

`loggerGrp`

loggerUserLogEnable

loggerUserLogNumInfo

loggerUserLogNumWarning

loggerUserLogNumError

loggerUserLogNumFatal

loggerUserLogClearCountersTime

`subscribersGrp`

subscribersInfoTable

subscribersInfoEntry

subscribersNumIntroduced

subscribersNumFree

subscribersNumIpAddrMappings

subscribersNumIpAddrMappingsFree

subscribersNumIpRangeMappings

subscribersNumIpRangeMappingsFree

subscribersNumVlanMappings

subscribersNumVlanMappingsFree

subscribersNumActive

subscribersNumActivePeak

subscribersNumActivePeakTime

subscribersNumUpdates

subscribersCountersClearTime

subscribersNumTpIpRanges

subscribersNumTpIpRangesFree

subscribersNumAnonymous

subscribersNumWithSessions

subscribersPropertiesTable

subscribersPropertiesEntry

spIndex

spName

spType

subscribersPropertiesValueTable

subscribersPropertiesValueEntry

spvIndex

spvSubName

spvPropertyName

spvRowStatus

spvPropertyStringValue

spvPropertyUintValue

spvPropertyCounter

`trafficProcessorGrp`

tpInfoTable

tpInfoEntry

tpModuleIndex

tpIndex

tpTotalNumHandledPackets

tpTotalNumHandledFlows

tpNumActiveFlows

tpNumActiveFlowsPeak

tpNumActiveFlowsPeakTime

tpNumTcpActiveFlows

tpNumTcpActiveFlowsPeak

tpNumTcpActiveFlowsPeakTime

tpNumUdpActiveFlows

tpNumUdpActiveFlowsPeak

tpNumUdpActiveFlowsPeakTime

tpNumNonTcpUdpActiveFlows

tpNumNonTcpUdpActiveFlowsPeak

tpNumNonTcpUdpActiveFlowsPeakTime

tpTotalNumBlockedPackets

tpTotalNumBlockedFlows

tpTotalNumDiscardedPacketsDueToBwLimit

tpTotalNumWredDiscardedPackets

tpTotalNumFragments

tpTotalNumNonIpPackets

tpTotalNumIpCrcErrPackets

tpTotalNumIpLengthErrPackets

tpTotalNumIpBroadcastPackets

tpTotalNumTtlErrPackets

tpTotalNumTcpUdpCrcErrPackets

tpClearCountersTime

tpHandledPacketsRate

tpHandledPacketsRatePeak

tpHandledPacketsRatePeakTime

tpHandledFlowsRate

tpHandledFlowsRatePeak

tpHandledFlowsRatePeakTime

tpCpuUtilization

tpCpuUtilizationPeak

tpCpuUtilizationPeakTime

tpFlowsCapacityUtilization

tpFlowsCapacityUtilizationPeak

tpFlowsCapacityUtilizationPeakTime

tpServiceLoss

`portGrp`

portTable

portEntry

portModuleIndex

portIndex

portType

portNumTxQueues

portIfIndex

portAdminSpeed

portAdminDuplex

portOperDuplex

portLinkIndex

portOperStatus

`txQueuesGrp`

txQueuesTable

txQueuesEntry

txQueuesModuleIndex

txQueuesPortIndex

txQueuesQueueIndex

txQueuesDescription

txQueuesBandwidth

txQueuesUtilization

txQueuesUtilizationPeak

txQueuesUtilizationPeakTime

txQueuesClearCountersTime

txQueuesDroppedBytes

`globalControllersGrp`

globalControllersTable

globalControllersEntry

globalControllersModuleIndex

globalControllersPortIndex

globalControllersIndex

globalControllersDescription

globalControllersBandwidth

globalControllersUtilization

globalControllersUtilizationPeak

globalControllersUtilizationPeakTime

globalControllersClearCountersTime

globalControllersDroppedBytes

`applicationGrp`

appInfoTable

appInfoEntry

appName

appDescription

appVersion

appPropertiesTable

appPropertiesEntry

apIndex

apName

apType

appPropertiesValueTable

appPropertiesValueEntry

apvIndex

apvPropertyName

apvRowStatus

apvPropertyStringValue

apvPropertyUintValue

apvPropertyCounter

`trafficCountersGrp`

trafficCountersTable

trafficCountersEntry

trafficCounterIndex

trafficCounterValue

trafficCounterName

trafficCounterType

`attackGrp`

attackTypeTable

attackTypeEntry

attackTypeIndex

attackTypeName

attackTypeCurrentNumAttacks

attackTypeTotalNumAttacks

attackTypeTotalNumFlows

attackTypeTotalNumSeconds

attackTypeTableClearTime

SCE Events: pcubeSeEvents

operationalStatusOperationalTrap (pcubeSeEvents 1)

The system operational state of the SCE Platform has changed to *Operational* (3).

operationalStatusWarningTrap (pcubeSeEvents 2)

The system operational state of the SCE Platform has changed to *Warning* (4).

operationalStatusFailureTrap (pcubeSeEvents 3)

The system operational state of the SCE Platform has changed to *Failure* (5)."

systemResetTrap (pcubeSeEvents 4)

The agent entity is about to reset itself either per user request or due to a fatal event.

chassisTempAlarmOnTrap (pcubeSeEvents 5)

The **chassisTempAlarm** object in this MIB has transitioned to the *On* (3) state, indicating that the temperature is too high.

chassisTempAlarmOffTrap (pcubeSeEvents 6)

The **chassisTempAlarm** object in this MIB has transitioned to the *Off* (2) state, indicating that the temperature level is back to normal.

chassisVoltageAlarmOnTrap (pcubeSeEvents 7)

The **chassisVoltageAlarm** object in this MIB has transitioned to the *On* (3) state, indicating that the voltage level is is out of safe bounds.

chassisFansAlarmOnTrap (pcubeSeEvents 8)

The **chassisFansAlarm** object in this MIB has transitioned to the *On* (3) state, indicating fan malfunction.

chassisPowerSupplyAlarmOnTrap (pcubeSeEvents 9)

The *chassisPowerSupplyAlarm* object in this MIB has transitioned to the *On* (3) state, indicating power supply malfunction.

rdrActiveConnectionTrap (pcubeSeEvents 10)

One of the RDR-formatter connections has become the active connection.

rdrNoActiveConnectionTrap (pcubeSeEvents 11)

There is no active connection between the RDR-formatter and any Collection Manager.

rdrConnectionUpTrap (pcubeSeEvents 12)

The **rdrFormatterDestConnectionStatus** object in this MIB has transitioned to *Up* (2), indicating that one of the RDR-formatter connections was established.

rdrConnectionDownTrap (pcubeSeEvents 13)

The **rdrFormatterDestConnectionStatus** object in this MIB has transitioned to *Down* (3), indicating that one of the RDR-formatter connections was disconnected.

loggerUserLogIsFullTrap (pcubeSeEvents 18)

The User log file is full. The agent entity then rolls to the next file.

sntpClockDriftWarnTrap (pcubeSeEvents 19)

The SNTP agent has not received an SNTP time update for a long period, which may result in a time drift of the agent entity's clock.

linkModeBypassTrap (pcubeSeEvents 20)

The link mode has changed to bypass.

linkModeForwardingTrap (pcubeSeEvents 21)

The link mode has changed to forwarding.

linkModeCutoffTrap (pcubeSeEvents 22)

The link mode has changed to cutoff.

pcubeSeEventGenericString1 (pcubeSeEvents 23)

Temporary string used for traps.

pcubeSeEventGenericString2 (pcubeSeEvents 24)

Temporary string used for traps.

moduleAttackFilterActivatedTrap (pcubeSeEvents 25)

The attack filter module has detected an attack and activated a filter. The type of attack-filter that was activated is returned in pcubeSeEventGenericString1.

Following are several examples of pcubeSeEventGenericString1 for various scenarios:

- **Attack detected automatically** (the number of open flows or ddos-suspected flows has exceeded the maximum configured for the attack detector):
  - **Source of the attack is detected** (at the subscriber side, IP address = 10.1.4.134, attacking the network side using UDP., number of open flows = 10000, configured action is 'report'):

    ```
    Attack detected: Attack from IP address 10.1.4.134, from
    subscriber side, protocol UDP. 10000 concurrent open flows
    detected, 57 concurrent Ddos-suspected flows detected.
    Action is: Report.
    ```

  - **Target of the attack is detected** (at the network side, IP address = 10.1.4.135, being attacked from the subscriber side using ICMP, number of ddos-suspected flows = 500, configured action is 'block'):

    ```
    Attack detected: Attack on IP address 10.1.4.135, from
    subscriber side, protocol ICMP. 745 concurrent open flows
    detected, 500 concurrent Ddos-suspected flows detected.
    Action is: Block.
    ```

- **Forced filtering** using the 'force-filter' command:
  - Action is 'block', attack-direction is attack-source, side is subscriber, IP address = 10.1.1.1, and protocol is TCP:

    ```
    Attack filter: Forced block of flows from IP address
    10.1.1.1, from subscriber side, protocol TCP. Attack forced
    using a force-filter command.
    ```

  - When the action is 'report', attack-direction is attack-destination, side is subscriber, IP address = 10.1.1.1, and protocol is Other:

    ```
    Attack filter: Forced report to IP address 10.1.1.1, from
    network side, protocol Other. Attack forced using a force-
    filter command.
    ```

moduleAttackFilterDeactivatedTrap (pcubeSeEvents 26)

The attack filter module has removed a filter that was previously activated.

- Attack filter type: in pcubeSeEventGenericString1 (refer to corresponding moduleAttackFilterActivatedTrap)
- Reason for deactivating the filter: in pcubeSeEventGenericString2

Following are several examples of pcubeSeEventGenericString1 for various scenarios:

- **Attack end detected automatically** (the number of open flows or ddos-suspected flows drops below the minimum value configured for the attack detector):

  End-of-attack detected: Attack on IP address 10.1.4.135, from subscriber side, protocol UDP. Action is: Report. Duration 20 seconds, attack comprised of 11736 flows.

  End-of-attack detected: Attack from IP address 10.1.4.134, from subscriber side, protocol ICMP. Action is: Block. Duration 10 seconds, attack comprised of 2093 flows.

- **Attack end forced** by a ‘dont-filter’, or a previous ‘force-filter’ command is removed:

  Attack filter: Forced to end block of flows from IP address 10.1.1.1, from subscriber side, protocol TCP. Attack end forced using a 'no force-filter' or a 'dont-filter' command. Duration 6 seconds, 1 flows blocked.

  Attack filter: Forced to end report to IP address 10.1.1.1, from network side, protocol Other. Attack end forced using a 'no force-filter' or a 'dont-filter' command. Duration 13 seconds, attack comprised of 1 flows.

moduleEmAgentGenericTrap (pcubeSeEvents 27)

A generic trap used by the Cisco EM agent.

- Trap name: in pcubeSeEventGenericString1 (refer to corresponding moduleAttackFilterActivatedTrap)
- Relevant parameter: in pcubeSeEventGenericString2

linkModeSniffingTrap (pcubeSeEvents 28)

The agent entity has detected that the **linkOperMode** object in this MIB has changed to sniffing(5).

moduleRedundancyReadyTrap (pcubeSeEvents 29)

The module was able to connect and synch with a redundant entity, and is now ready to handle fail-over if needed.

moduleRedundantConfigurationMismatchTrap (pcubeSeEvents 30)

The module was not able to synch with a redundant entity, due to an incompatibility in essential configuration parameters between the module and the redundant entity.

moduleLostRedundancyTrap (pcubeSeEvents 31)

The module has lost the ability to perform the fail-over procedure.

moduleSmConnectionDownTrap (pcubeSeEvents 32)

The virtual connection to the SM (smartSub Manager) is broken.

moduleSmConnectionUpTrap (pcubeSeEvents 33)

The virtual connection to the SM is up and working.

moduleOperStatusChangeTrap (pcubeSeEvents 34)

The value of **moduleOperStatus** has changed.

portOperStatusChangeTrap (pcubeSeEvents 35)

The value of the **portOperStatus** object of the **portIndex** has changed, indicating that the link was either forced down or the force down was released.

chassisLineFeedAlarmOnTrap (pcubeSeEvents 36)

The agent entity has detected that the **chassisLineFeed** object in this MIB has changed to the on(3) state.

rdrFormatterCategoryDiscardingReportsTrap (pcubeSeEvents 37)

The agent entity has detected that reports sent to this category are being discarded.

The rdrFormatterCategoryNumReportsDiscarded object in this MIB counts the number of discarded reports.

rdrFormatterCategoryStoppedDiscardingReportsTrap (pcubeSeEvents 38)

The agent entity has detected that reports sent to this category are no longer being discarded.

The rdrFormatterCategoryNumReportsDiscarded object in this MIB counts the number of discarded reports.

sessionStartedTrap (pcubeSeEvents 39)

The agent entity has accepted a new session. The pcubeSeEventGenericString1 contains the session type (telnet/SSH) and client IP address.

sessionEndedTrap (pcubeSeEvents 40)

The agent entity has detected the end of a session. The pcubeSeEventGenericString1 contains the session type (telnet/SSH) and client IP address.

sessionDeniedAccessTrap (pcubeSeEvents 41)

The agent entity has refused a session from unauthorized source. The pcubeSeEventGenericString1 contains the session type (telnet/SSH) and client IP address.

sessionBadLoginTrap (pcubeSeEvents 42)

The agent entity has detected attempt to login with a wrong password. The pcubeSeEventGenericString1 contains the session type (telnet/SSH) and client IP address.

System Group: systemGrp (pcubeSEObjs 1)

The System group provides data on the system-wide functionality of the SCE Platform.

sysOperationalStatus (systemGrp 1)

Indicates the operational status of the system.

Access RO

SYNTAX

INTEGER *{*

1 *(other):* none of the following

2 *(boot):* the system is in boot process

3 *(operational):* the system is operational

4 *(warning):* the system is in Warning status

5 *(failure):* the system is in Failure status

}

sysFailureRecovery (systemGrp 2)

Indicates the behavior of the system after abnormal boot.

Access RO

SYNTAX

INTEGER *{*

1 *(other):* none of the following

2 *(operational):* the system should enter Operational mode after abnormal boot

3 *(non-operational):* the system should enter Failure mode after abnormal boot

}

sysVersion (systemGrp 3)

The system version.

Access RO

SYNTAX

DisplayString

Chassis Group: chassisGrp (pcubeSEObjs 2)

The Chassis group defines and identifies the chassis, as well as environmental alarms related to the chassis.

ChassisSysType (chassisGrp 1)

The chassis system type.

Access RO

SYNTAX

INTEGER *{*

1 *(other):* none of the following

2 *(SE1000):* SE1000 platform

3 *(SE100):* SE100 platform

4 *(SE2000):* SE2000 platform

}

chassisPowerSupplyAlarm (chassisGrp 2)

Indicates whether the power supply to the chassis is normal. If the alarm is 'on', it means that one or more of the power supplies is not functional

Access RO

SYNTAX

INTEGER *{*

1 *(other):* none of the following

2 *(off):* the power supply to the chassis is normal

3 *(on):* the power supply to the chassis is not normal, and probably one or more of the power supplies is not functional.

}

chassisFansAlarm (chassisGrp 3)

Indicates whether all the fans on the chassis are functional.

Access RO

SYNTAX

INTEGER *{*

1 *(other):* none of the following

2 *(off):* all fans are functional

3 *(on):* one or more fans is not functional.

}

chassisTempAlarm (chassisGrp 4)

Indicates the chassis temperature alarm status.

Access RO

SYNTAX

INTEGER *{*

1 *(other):* none of the following

2 *(off):* temperature is within acceptable range

3 *(on):* temperature is too high.

}

chassisVoltageAlarm (chassisGrp 5)

Indicates the chassis internal voltage alarm status. If the alarm is ‘on’, it indicates that the voltage level of one or more unit in the chassis is not in the normal range.

Access RO

SYNTAX

INTEGER *{*

1 *(other):* none of the following

2 *(off):* voltage level is within normal range

3 *(on):* voltage level is out of the acceptable bounds.

}

chassisNumSlots (chassisGrp 6)

Indicates the number of slots in the chassis available for plug-in modules, including both currently occupied and empty slots.

Access RO

SYNTAX

INTEGER *(0..255)*

chassisSlotConfig (chassisGrp 7)

An indication of which slots in the chassis are occupied.

This is an integer value with bits set to indicate configured modules. It is expressed as the function:

Sum of f(x) as x goes from 1 to the number of slots, where:

- no module inserted: f(x) = 0
- module inserted: f(x) = exp(2, x-1)

Access RO

SYNTAX

INTEGER *(0..65535)*

chassisPsuType (chassisGrp 8)

Indicates the type of the power supplies.

Access RO

SYNTAX

INTEGER *{*

1 *(other):* none of the following

2 *(AC):* AC power supply

3 *(DC):* DC power supply

}

chassisLineFeedAlarm (chassisGrp 9)

Indicates whether the line feed to the chassis is connected and whether it is supplying power to the power supply unit.

Access RO

SYNTAX

INTEGER *{*

1 *(other):* none of the following

2 *(OFF):* The line feed to the chassis is connected and has power

3 *(ON):* The line feed to the chassis is not normal. One or both of the line feeds may not be connected properly or have no power.

}

Module Group: moduleGrp (pcubeSEObjs 3)

The Module group identifies and defines the modules, or cards, in the SCE Platform.

moduleTable (moduleGrp 1)

A list of module entries containing information defining the modules in the chassis.

The number of entries is the number of modules in the chassis.

Access not-accessible

SYNTAX

Sequence of moduleEntry

moduleEntry (moduleTable 1)

Entry containing a number of parameters defining the physical characteristics of one module in the chassis.

Access not-accessible

INDEX

{moduleIndex}

SYNTAX

SEQUENCE *{*

moduleIndex

moduleType

moduleNumTrafficProcessors

moduleSlotNum

moduleHwVersion

moduleNumPorts

moduleNumLinks

moduleConnectionMode

moduleSerialNumber

moduleUpStreamAttackFilteringTime

moduleUpStreamLastAttackFilteringTime

moduleDownStreamAttackFilteringTime

moduleDownStreamLastAttackFilteringTime

moduleAttackObjectsClearTime

moduleAdminStatus

moduleOperStatus}

moduleIndex (moduleEntry 1)

An ID number identifying the module. A unique value for each module within the chassis.

Access RO

SYNTAX

INTEGER *(1..255)*

moduleType (moduleEntry 2)

The type of module.

Access RO

SYNTAX

INTEGER *{*

1 *(other):* none of the following

2 *(gbe2Module):* 2 port Gigabit Ethernet line interface, 2 Fast Ethernet 10/100 management interfaces

3 *(fe2Module):* 2 port Fast Ethernet line interface, 1 Fast Ethernet 10/100 management interface

4 *(gbe4Module):* 4 port Gigabit Ethernet line interface, 2 Fast Ethernet 10/100 management interfaces

5 *(fe4Module):* 4 port Fast Ethernet line interface, 2 Fast Ethernet 10/100 management interfaces

6 *(oc12-4Module):* 4 port OC12 line interface, 2 Fast Ethernet 10/100 management interfaces

7 *(fe8Module):* 8 port Fast Ethernet line interface, 2 Fast Ethernet 10/100 management interfaces

}

moduleNumTrafficProcessors (moduleEntry 3)

The number of traffic processors supported by the module.

Access RO

SYNTAX

INTEGER *(0..255)*

moduleSlotNum (moduleEntry 4)

The number of the slot in the chassis in which the module is installed.

Valid entries are from 1 to the value of chassisNumSlots.

Access RO

SYNTAX

INTEGER *(1..255)*

moduleHwVersion (moduleEntry 5)

The hardware version of the module.

Access RO

SYNTAX

DisplayString

moduleNumPorts (moduleEntry 6)

The number of ports supported by the module.

Access RO

SYNTAX

INTEGER *(0..255)*

moduleNumLinks (moduleEntry 7)

The number of links carrying inband traffic that are supported by the module. The link is uniquely defined by the two ports that are at its endpoints.

Access RO

SYNTAX

INTEGER *(0..255)*

moduleConnectionMode (moduleEntry 8)

Indicates the connection mode of the module.

Access RO

SYNTAX

INTEGER *{*

1 *(other):* none of the following

2 *(inline):* SCE is both receiving and transmitting traffic on the line ports.

3 *(receive-only):* SCE can only receive packets from the line ports. This mode is suitable for external splitting topology.

4 *(inline-cascade):* SCE is both receiving and transmitting traffic on the line ports and the cascade ports.

5 *(receive-only-cascade):* SCE can only receive packets from the line and the cascade ports. This mode is suitable for external splitting topology

moduleSerialNumber (moduleEntry 9)

The serial number of the module.

Access RO

SYNTAX

DisplayString

moduleUpStreamAttackFilteringTime (moduleEntry 10)

The accumulated time (in hundredths of a second) during which attack up-stream traffic was filtered.

Access RO

SYNTAX

TimeTicks

moduleUpStreamLastAttackFilteringTime (moduleEntry 11)

The time (in hundredths of a second) since the previous attack filtered in the up-stream traffic.

Access RO

SYNTAX

TimeTicks

moduleDownStreamAttackFilteringTime (moduleEntry 12)

The accumulated time (in hundredths of a second) during which attack down-stream traffic was filtered.

Access RO

SYNTAX

TimeTicks

moduleDownStreamLastAttackFilteringTime (moduleEntry 13)

The time (in hundredths of a second) since the previous attack filtered in the down-stream traffic.

Access RO

SYNTAX

TimeTicks

moduleAttackObjectsClearTime (moduleEntry 14)

The time (in hundredths of a second) since the attack objects were cleared. Writing a 0 to this object causes the counters to be cleared.

Access RO

SYNTAX

TimeTicks

moduleAdminStatus (moduleEntry 15)

Indicates whether the module is configured to handle traffic on startup or reboot (active), to be the hot standby.

Access RO

SYNTAX

INTEGER *{*

1 *(other):* none of the following

2 *(primary):* Handle traffic on startup.

3 *(secondary):* Fail-over module on startup.

}

moduleOperStatus (moduleEntry 16)

Indicates whether the module is currently handling (active), or is on standby.

Access RO

SYNTAX

INTEGER *{*

1 *(other):* none of the following

2 *(active):* Currently is handling traffic.

3 *(standby):* Currently is the fail-over module.

}

Link Group: linkGrp (pcubeSEObjs 4)

The Link group defines and identifies the link. It provides information regarding the mode of operation of the link defined for each status of the platform.

linkTable (linkGrp 1)

A list of link entries containing information regarding the configuration and status of the links that pass through the SCE and carry in-band traffic.

The number of entries is determined by the number of modules in the chassis and the number of links on each module.

Access not-accessible

SYNTAX

Sequence of linkEntry

linkEntry (linkTable 1)

Entry containing information about the Link.

Access not-accessible

INDEX

{linkModuleIndex, linkIndex}

SYNTAX

SEQUENCE *{*

linkModuleIndex

linkIndex

linkAdminModeOnActive

linkAdminModeOnFailure

linkOperMode

linkStatusReflectionEnable

linkSubscriberSidePortIndex

linkNetworkSidePortIndex

}

linkModuleIndex (linkEntry 1)

An index value (**moduleIndex**) that uniquely identifies the module where this link is located.

Access RO

SYNTAX

INTEGER *(1..255)*

linkIndex (linkEntry 2)

An index value that uniquely identifies the link within the specified module.

Valid entries are 1 to the value of **moduleNumLinks** for this module.

Access RO

SYNTAX

INTEGER *(1..255)*

linkAdminModeOnActive (linkEntry 3)

The desired mode of the link when the operating status of the module is active and it is not in boot or failure.

Possible values (*LinkModeType*):

- *Bypass*: the traffic is forwarded from one port to the other using an internal splitter.
- *Forwarding*: the traffic is forwarded by the internal hardware and software modules of the ***SCE*** platform.

Access RO

SYNTAX

LinkModeType

linkAdminModeOnFailure (linkEntry 4)

The desired mode of the link when the system status is failure.

Possible values (*LinkModeType*):

- *Bypass*: the traffic is forwarded from one port to the other using an internal splitter.
- *Cutoff*: all traffic is dropped by the SCE.

Access RO

SYNTAX

LinkModeType

linkOperMode (linkEntry 5)

The current operational mode of the link.

Possible values (*LinkModeType*):

- *Bypass*: the traffic is forwarded from one port to the other using an internal splitter with no processing taking place.
- *Forwarding*: the traffic is forwarded by the internal hardware and software modules of the SCE.
- *Sniffing*: the traffic is forwarded in the same manner as in Bypass mode, however it passes through and is analysed by the internal software and hardware modules of the SCE Platform.

Access RO

SYNTAX

LinkModeType

linkStatusReflectionEnable (linkEntry 6)

Indicates whether failure of the physical link on one interface should trigger the failure of the link on the other interface on the module.

Access RO

SYNTAX

INTEGER *{*

1 *(enabled)*

2 *(disabled)*

}

linkSubscriberSidePortIndex (linkEntry 7)

An index value that uniquely identifies this link with the related port that is connected to the subscriber side.

Access RO

SYNTAX

INTEGER *(0..255)*

linkSubscriberSidePortIndex (linkEntry 8)

An index value that uniquely identifies this link with the related port that is connected to the network side.

Access RO

SYNTAX

INTEGER *(0..255)*

Disk Group: diskGrp (pcubeSEObjs 5)

The Disk group provides data regarding the space utilization on the disk.

diskNumUsedBytes (diskGrp 1)

The number of used bytes on the disk.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

diskNumFreeBytes (diskGrp 2)

The number of free bytes on the disk.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

RDR Formatter Group: rdrFormatterGrp (pcubeSEObjs 6)

The RDR Formatter provides information regarding RDR Formatter destinations (Collection Managers), as well as RDR statistics.

rdrFormatterEnable (rdrFormatterGrp 1)

Indicates whether the RDR-formatter is enabled or disabled.

When the RDR-formatter is enabled, it sends the reports it gets from the traffic processors to the Collection Manager as defined in the rdrFormatterDestTable.

Access RO

SYNTAX

```
INTEGER {
1 (enabled)
2 (disabled)
}
```

rdrFormatterDestTable (rdrFormatterGrp 2)

This table lists the addresses of Collection Managers.

If the RDR-formatter is enabled, the destination with the highest priority to which a TCP connection can be established is designated as the active connection, and would receive the reports generated by the traffic processors.

The table may contain a maximum of three entries.

Access not-accessible

SYNTAX

```
Sequence of rdrFormatterDestEntry
```

rdrFormatterDestEntry (rdrFormatterDestTable 1)

Entry defining one RDR destination.

Access not-accessible

INDEX

{ rdrFormatterDestIPAddr, rdrFormatterDestPort }

SYNTAX

SEQUENCE *{*

rdrFormatterDestIPAddr

rdrFormatterDestPort

rdrFormatterDestPriority

rdrFormatterDestStatus

rdrFormatterDestConnectionStatus

rdrFormatterDestNumReportsSent

rdrFormatterDestNumReportsDiscarded

rdrFormatterDestReportRate

rdrFormatterDestReportRatePeak

rdrFormatterDestReportRatePeakTime

}

rdrFormatterDestIPAddr (rdrFormatterDestEntry 1)

The IP address of a Collection Manager.

Access RO

SYNTAX

IP Address

rdrFormatterDestPort (rdrFormatterDestEntry 2)

The TCP port on which the Collection Manager listens and the to which the RDR-Formatter should connect.

Access RO

SYNTAX

INTEGER *(1...65535)*

rdrFormatterDestPriority (rdrFormatterDestEntry 3)

The priority given to the Collection Manager. The active Collection Manager is the Collection Manager with the highest priority whose TCP connection is up.

Access RO

SYNTAX

INTEGER *(1...100)*

rdrFormatterDestStatus (rdrFormatterDestEntry 4)

Indicates whether this destination is the active one.

In redundancy and simple-load-balancing modes there can be only one 'active' destination, which is the one to which the reports are sent. In multicast mode all destinations receive the active mode.

Access RO

SYNTAX

INTEGER *{*

1 *(other):* none of the following

2 *(active):* this destination is where the reports are sent

3 *(standby):* this destination is a backup

}

rdrFormatterDestConnectionStatus (rdrFormatterDestEntry 5)

The status of TCP connection to this destination.

Access RO

SYNTAX

INTEGER *{*

1 *(other):* none of the following

2 *(up):* the TCP connection to this destination is up

3 *(down):* the TCP connection to this destination is down

}

rdrFormatterDestNumReportsSent (rdrFormatterDestEntry 6)

The number of reports sent by the RDR-formatter to this destination.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

rdrFormatterDestNumReportsDiscarded (rdrFormatterDestEntry 7)

The number of reports dropped by the RDR-formatter at this destination.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

rdrFormatterDestReportRate (rdrFormatterDestEntry 8)

The current rate (in reports per second) of sending reports to this destination.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

rdrFormatterDestReportRatePeak (rdrFormatterDestEntry 9)

The maximum rate of sending reports to this destination.

ACCESS RO

SYNTAX

Unsigned32 *(0...4294967295)*

rdrFormatterDestReportRatePeakTime (rdrFormatterDestEntry 10)

The time (in hundredths of a second) since the **rdrFormatterDestReportRatePeak** value occurred.

Access RO

SYNTAX

TimeTicks

rdrFormatterNumReportsSent (rdrFormatterGrp 3)

The number of reports sent by the RDR-formatter.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

rdrFormatterNumReportsDiscarded (rdrFormatterGrp 4)

The number of reports dropped by the RDR-formatter.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

rdrFormatterClearCountersTime (rdrFormatterGrp 5)

The time (in hundredths of a second) since the RDR-formatter counters were last cleared. Writing a 0 to this object causes the RDR-formatter counters to be cleared.

Access RW

SYNTAX

TimeTicks

rdrFormatterReportRate (rdrFormatterGrp 6)

The current rate (in reports per second) of sending reports to all destinations.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

rdrFormatterReportRatePeak (rdrFormatterGrp 7)

The maximum rate of sending reports to all destinations.

ACCESS RO

SYNTAX

Unsigned32 *(0...4294967295)*

rdrFormatterReportRatePeakTime (rdrFormatterGrp 8)

The time (in hundredths of a second) since the **rdrFormatterReportRatePeak** value occurred.

Access RO

SYNTAX

TimeTicks

rdrFormatterProtocol (rdrFormatterGrp 9)

The RDR protocol currently in use.

Access RO

SYNTAX

INTEGER *{*

1 *(other):* none of the following

2 *(RDRv1):* RDR protocol version 1

3 *(RDRv2):* RDR protocol version 2

}

rdrFormatterForwardingMode (rdrFormatterGrp 10)

The manner in which the RDR formatter sends the reports to the destinations.

Access RO

SYNTAX

INTEGER *{*

1 *(other):* none of the following

2 *(redundancy):* all RDRs are sent to the primary (active) destination, and all other destinations are in standby

3 *(simpleLoadBalancing):* each successive RDR is sent to a different destination, one destination after the other, in a round robin manner

4 *(multicast):* all RDRs are sent to all destinations

}

rdrFormatterCategoryTable (rdrFormatterGrp 11)

This table describes the different categories of RDRs and supplies some statistical information about the RDRs sent to these categories

Access not-accessible

SYNTAX

Sequence of rdrFormatterCategoryEntry

rdrFormatterCategoryEntry (rdrFormatterCategoryTable 1)

Entry containing information about the RDR formatter categories.

Access not-accessible

INDEX

{rdrFormatterCategoryIndex}

SYNTAX

SEQUENCE *{*

rdrFormatterCategoryIndex

rdrFormatterCategoryName

rdrFormatterCategoryNumReportsSent

rdrFormatterCategoryNumReportsDiscarded

rdrFormatterCategoryReportRate

rdrFormatterCategoryReportRatePeak

rdrFormatterCategoryReportRatePeakTime

rdrFormatterCategoryNumReportsQueued

}

rdrFormatterCategoryIndex (rdrFormatterCategoryEntry 1)

The RDR formatter category number.

Access RO

SYNTAX

INTEGER *(1..4)*

rdrFormatterCategoryName (rdrFormatterCategoryEntry 2)

The name of the category.

Access RO

SYNTAX

DisplayString

rdrFormatterCategoryNumReportsSent (rdrFormatterCategoryEntry 3)

The number of reports sent by the RDR-formatter to this category.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

rdrFormatterCategoryNumReportsDiscarded (rdrFormatterCategoryEntry 4)

The number of reports dropped by the RDR formatter for this category.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

rdrFormatterCategoryReportRate (rdrFormatterCategoryEntry 5)

The rate of the reports (in reports per second) currently sent to this category.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

rdrFormatterCategoryReportRatePeak (rdrFormatterCategoryEntry 6)

The maximum report rate sent to this category.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

rdrFormatterCategoryReportRatePeakTime (rdrFormatterCategoryEntry 7)

The time (in hundredths of a second) since the **rdrFormatterCategoryReportRatePeak** value occurred.

Access RO

SYNTAX

TimeTicks

rdrFormatterCategoryNumReportsQueued (rdrFormatterCategoryEntry 8)

The number of pending reports in this category.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

rdrFormatterCategoryDestTable (rdrFormatterGrp 12)

This table describes the partition of the RDR destinations between the different categories and the priority and status of each destination in each category

Access not-accessible

SYNTAX

Sequence of rdrFormatterCategoryDestEntry

rdrFormatterCategoryDestEntry (rdrFormatterCategoryDestTable 1)

A destination table entry.

Access not-accessible

INDEX

{rdrFormatterCategoryIndex, rdrFormatterDestIPAddr, rdrFormatterDestPort}

SYNTAX

SEQUENCE *{*

rdrFormatterCategoryDestPriority

rdrFormatterCategoryDestStatus

}

rdrFormatterCategoryDestPriority (rdrFormatterCategoryDestEntry 1)

The priority assigned to the Collection Manager for this category.

The active Collection Manager is the Collection Manager with the highest priority and a TCP connection that is up.

Access RO

SYNTAX

INTEGER *(1...100)*

rdrFormatterCategoryDestStatus (rFormatterCategoryDestEntry 2)

Indicates whether the destination is currently active or standby.

In redundancy and in simple Load Balancing `rdrFormatterForwardingMode` there can be only one active destination, which is where the reports are currently being sent. In multicast mode, all destinations will be assigned the active(2) status

Access RO

SYNTAX

INTEGER {

`1 (other)`: none of the following

`2 (active)`: this is the destination to which reports are currently being sent

`3 (standby)`: this destination is a backup

}

Logger Group: loggerGrp (pcubeSEObjs 7)

The Logger group is responsible for logging the system synchronous and asynchronous events.

loggerUserLogEnable (loggerGrp 1)

Indicates whether the logging of user information is enabled or disabled.

Access RO

SYNTAX

INTEGER *{*

1 *(enabled)*

2 *(disabled)*

}

loggerUserLogNumInfo (loggerGrp 2)

The number of Info messages logged into the user log file since last reboot or last time the counter was cleared

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

loggerUserLogNumWarning (loggerGrp 3)

The number of **Warning** messages logged into the user log file since last reboot or last time the counter was cleared.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

loggerUserLogNumError (loggerGrp 4)

The number of **Error** messages logged into the user log file since last reboot or last time the counter was cleared.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

loggerUserLogNumFatal (loggerGrp 5)

The number of **Fatal** messages logged into the user log file since last reboot or last time the counter was cleared

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

loggerUserLogClearCountersTime (loggerGrp 6)

The time (in hundredths of a second) since user log counters were last cleared.

Writing a 0 to this object causes the user log counters to be cleared.

Access RW

SYNTAX

TimeTicks

Subscribers Group: subscribersGrp (pcubeSEObjs 8)

The Subscribers group provides statistics concerning the number of subscribers and subscriber mappings. It also provides data on the subscriber properties and the value of those properties for a specified subscriber.

subscribersInfoTable (subscribersGrp 2)

Data regarding subscriber management operations performed.

Access not-accessible

SYNTAX

Sequence of subscribersInfoEntry

subscribersInfoEntry (subscribersInfoTable 1)

Entry describing the subscriber management operations performed on a certain module.

Access not-accessible

INDEX

{moduleIndex}

SYNTAX

SEQUENCE {

subscribersNumIntroduced

subscribersNumFree

subscribersNumIpAddrMappings

subscribersNumIpAddrMappingsFree

subscribersNumIpRangeMappings

subscribersNumIpRangeMappingsFree

subscribersNumVlanMappings

subscribersNumVlanMappingsFree

subscribersNumActive

subscribersNumActivePeak

subscribersNumActivePeakTime

subscribersNumUpdates

subscribersCountersClearTime

subscriberssubscribersNumTpIpRangeMappings

subscribersNumTpIpRangeMappingsFreeCountersClearTime

subscribersNumAnonymous

subscribersNumWithSessions

}

subscribersNumIntroduced (subscribersInfoEntry 1)

The current number of subscribers introduced to the SCE. These subscribers may or may not have IP address or VLAN mappings. Subscribers who do not have mappings of any kind cannot be associated with traffic, and will be served by the SCE according to the default settings.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

subscribersNumFree (subscribersInfoEntry 2)

The number of subscribers that may be introduced in addition to the currently introduced subscribers.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

subscribersNumIpAddrMappings (subscribersInfoEntry 3)

The current number of IP address to subscriber mappings.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

subscribersNumIpAddrMappingsFree (subscribersInfoEntry 4)

The number of free IP address to subscriber mappings that are available for defining new mappings.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

subscribersNumIpRangeMappings (subscribersInfoEntry 5)

The current number of IP-range to subscriber mappings.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

subscribersNumIpRangeMappingsFree (subscribersInfoEntry 6)

The number of free IP range to subscriber mappings that are available for defining new mappings.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

subscribersNumVlanMappings (subscribersInfoEntry 7)

The current number of VLAN to subscriber mappings

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

subscribersNumVlanMappingsFree (subscribersInfoEntry 8)

The number of free VLAN to subscriber mappings that are available for defining new mappings.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

subscribersNumActive (subscribersInfoEntry 9)

The current number of active subscribers. These subscribers necessarily have IP address or VLAN mappings that define the traffic to be served according to the subscriber service agreement.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

subscribersNumActivePeak (subscribersInfoEntry 10)

The peak value of **subscribersNumActive** since the last time it was cleared or the system started.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

subscribersNumActivePeakTime (subscribersInfoEntry 11)

The time (in hundredths of a second) since the s**ubscribersNumActivePeak** value occurred.

Access RO

SYNTAX

TimeTicks

subscribersNumUpdates (subscribersInfoEntry 12)

The accumulated number of subscribers database updates received by the SCE.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

subscribersCountersClearTime (subscribersInfoEntry 13)

The time (in hundredths of a second) since the subscribers counters were cleared.

Writing a 0 to this object causes the counters to be cleared.

Access RW

SYNTAX

TimeTicks

subscribersNumTpIpRangeMappings (subscribersInfoEntry 14)

The current number of IP range to Traffic Processor mappings.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

subscribersNumTpIpRangeMappingsFree (subscribersInfoEntry 15)

The current number of IP range to Traffic Processor mappings that are available for defining new mappings.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

subscribersNumAnonymous (subscribersInfoEntry 16)

The current number of anonymous subscribers.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

subscribersNumWithSessions (subscribersInfoEntry 17)

The current number of subscribers with open sessions.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

subscribersPropertiesTable (subscribersGrp 2)

List of all subscriber properties. This table is updated each time an application is loaded on the SCE Platform.

Access not-accessible

SYNTAX

Sequence of subscribersPropertiesEntry

subscribersPropertiesEntry (subscribersPropertiesTable 1)

Entry describing subscriber properties of the application relevant for a certain module.

Access not-accessible

INDEX

{moduleIndex, spIndex}

SYNTAX

SEQUENCE *{*

spIndex

spName

spType

}

spIndex (subscribersPropertiesEntry 1)

An index value that uniquely identifies the subscriber property.

Access RO

SYNTAX

INTEGER *(1..255)*

spName (subscribersPropertiesEntry 2)

Name of the subscriber property.

Access RO

SYNTAX

DisplayString

spType (subscribersPropertiesEntry 3)

Property type in respect to: variable type (integer, boolean, string etc), number of elements (scalar or array), and restrictions, if any.

Access RO

SYNTAX

`DisplayString`

subscriberPropertiesValuesTable (subscribersGrp 3)

The subscriber properties value table is used to provide values for the subscriber properties for a specific subscriber introduced into the SCE Platform.

An entry must be created by setting the entry spvRowStatus object with CreateAndGo (4) before setting the name of the subscriber and the property requested. The property requested must be one of the properties from the subscribersPropertiesTable. To remove an entry set the spvRowStatus object with Destroy (6).

To poll the subscriber property, either of these objects should be polled:

- spvPropertyStringValue
- spvPropertyUnitValue

The table is cleared when the application is unloaded.

Access not-accessible

SYNTAX

`Sequence of subscribersPropertiesValueEntry`

subscriberPropertiesValueEntry (subscriberPropertiesValueTable 1)

Entry providing information on the value of one of the specified subscriber properties.

Access not-accessible

INDEX

{moduleIndex, spvIndex}

SYNTAX

SEQUENCE *{*

SpvIndex

spvSubName

spvPropertyName

spvRowStatus

spvPropertyStringValue

spvPropertyUintValue

spvPropertyCounter64Value

}

spvIndex (subscriberPropertiesValueEntry 1)

An index value that uniquely identifies the entry.

Access RO

SYNTAX

INTEGER *(1.. 1024)*

spvSubName (subscriberPropertiesValueEntry 2)

A name that uniquely identifies the subscriber.

Access RC

SYNTAX

DisplayString *(Size 1...40)*

spvPropertyName (subscriberPropertiesValueEntry 3)

A name that uniquely identifies the subscriber property.

Array-type properties may be accessed one element at a time in C-like format. (For example: x[1], or y[1][2])

Access RC

SYNTAX

DisplayString *(Size 1...128)*

spvRowStatus (subscriberPropertiesValueEntry 4)

Controls creation of a table entry. Only setting CreateAndGo (4) and Destroy (6) will change the status of the entry.

Access RC

SYNTAX

RowStatus

spvPropertyStringValue (subscriberPropertiesValueEntry 5)

The value of the subscriber property in display string format.

Access RO

SYNTAX

DisplayString *(SIZE 0...128)*

spvPropertyUintValue (subscriberPropertiesValueEntry 6)

The value of the subscriber property in Uint format.

If the property cannot be cast to Uint format, getting this object returns zero.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

spvPropertyCounter64Value (subscriberPropertiesValueEntry 7)

The value of the subscriber property in Counter64 format.

If the property cannot be cast to Counter64 format, getting this object returns zero.

Access RO

SYNTAX

Counter64

Traffic Processor Group: trafficProcessorGrp (pcubeSEObjs 9)

The Traffic Processor group provides statistics regarding the traffic flow handled by each traffic processor.

tpInfoTable (trafficProcessorGrp 1)

The Traffic Processor Info table consists of data regarding traffic handled by the traffic processors, classified by packets and flows.

Access not-accessible

SYNTAX

Sequence of TpInfoEntry

tpInfoEntry (tpInfoTable)

Entry containing information from the traffic processors.

Access not-accessible

INDEX

{ tpModuleIndex, tpIndex }

SYNTAX

SEQUENCE *{*

tpModuleIndex

tpIndex

tpTotalNumHandledPackets

tpTotalNumHandledFlows

tpNumActiveFlows

tpNumActiveFlowsPeak

tpNumActiveFlowsPeakTime

tpNumTcpActiveFlows

tpNumTcpActiveFlowsPeak

tpNumTcpActiveFlowsPeakTime

tpNumUdpActiveFlows

tpNumUdpActiveFlowsPeak

tpNumUdpActiveFlowsPeakTime

tpNumNonTcpUdpActiveFlows

tpNumNonTcpUdpActiveFlowsPeak

tpNumNonTcpUdpActiveFlowsPeakTime

tpTotalNumBlockedPackets

tpTotalNumBlockedFlows

tpTotalNumDiscardedPacketsDueToBwLimit

tpTotalNumWredDiscardedPackets

tpTotalNumFragments

tpTotalNumNonIpPackets

tpTotalNumIpCrcErrPackets

tpTotalNumIpLengthErrPackets

tpTotalNumIpBroadcastPackets

tpTotalNumTtlErrPackets

tpTotalNumTcpUdpCrcErrPackets

tpClearCountersTime

tpHandledPacketsRate

tpHandledPacketsRatePeak

tpHandledPacketsRatePeakTime

tpHandledFlowsRate

tpHandledFlowsRatePeak

tpHandledFlowsRatePeakTime

tpCpuUtilization

tpCpuUtilizationPeak

tpCpuUtilizationPeakTime

tpFlowsCapacityUtilization

tpFlowsCapacityUtilizationPeak

tpFlowsCapacityUtilizationPeakTime

tpServiceLoss

}

tpModuleIndex (tpInfoEntry 1)

An index value (**moduleIndex**) that uniquely identifies the module in which this traffic processor is located.

Access RO

SYNTAX

INTEGER *(1...255)*

tpIndex (tpInfoEntry 2)

An index value that uniquely identifies the traffic processor within the specified module. The value is determined by the location of the traffic processor on the module.

Valid entries are 1 to the value of **moduleNumTrafficProcessors** for the specified module.

Access RO

SYNTAX

INTEGER *(1...255)*

tpTotalNumHandledPackets (tpInfoEntry 3)

The accumulated number of packets handled by this traffic processor since last reboot or last time this counter was cleared.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

tpTotalNumHandledFlows (tpInfoEntry 4)

The accumulated number of flows handled by this traffic processor since last reboot or last time this counter was cleared.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

tpNumActiveFlows (tpInfoEntry 5)

The number of flows currently being handled by this traffic processor.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

tpNumActiveFlowsPeak (tpInfoEntry 6)

The peak value of **tpNumActiveFlows** since the last time it was cleared or the system started.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

tpNumActiveFlowsPeakTime (tpInfoEntry 7)

The time (in hundredths of a second) since the **tpNumActiveFlowsPeak** value occurred.

Access RO

SYNTAX

TimeTicks

tpNumTcpActiveFlows (tpInfoEntry 8)

The number of TCP flows currently being handled by this traffic processor

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

TpNumTcpActiveFlowsPeak (tpInfoEntry 9)

The peak value of **tpNumTcpActiveFlows** since the last time it was cleared or the system started.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

tpNumTcpActiveFlowsPeakTime (tpInfoEntry 10)

The time (in hundredths of a second) since the **tpNumTcpActiveFlowsPeak** value occurred.

Access RO

SYNTAX

TimeTicks

tpNumUdpActiveFlows (tpInfoEntry 11)

The number of UDP flows currently being handled by the traffic processor.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

tpNumUdpActiveFlowsPeak (tpInfoEntry 12)

The peak value of **tpNumUdpActiveFlows** since the last time it was cleared or the system started.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

tpNumUdpActiveFlowsPeakTime (tpInfoEntry 13)

The time (in hundredths of a second) since the **tpNumUdpActiveFlowsPeak** value occurred.

Access RO

SYNTAX

TimeTicks

tpNumNonTcpUdpActiveFlows (tpInfoEntry 14)

The number of non TCP/UDP flows currently being handled by the traffic processor.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

tpNumNonTcpUdpActiveFlowsPeak (tpInfoEntry 15)

The peak value of **tpNumNonTcpUdpActiveFlows** since the last time it was cleared or the system started.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

tpNumNonTcpUdpActiveFlowsPeakTime (tpInfoEntry 16)

The time (in hundredths of a second) since the **tpNumNonTcpUdpActiveFlowsPeak** value occurred.

Access RO

SYNTAX

TimeTicks

tpTotalNumBlockedPackets (tpInfoEntry 17)

The accumulated number of packets discarded by the traffic processor according to application blocking rules.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

tpTotalNumBlockedFlows (tpInfoEntry 18)

The accumulated number of flows discarded by the traffic processor according to application blocking rules.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

tpTotalNumDiscardedPacketsDueToBwLimit (tpInfoEntry 19)

The accumulated number of packets discarded by the traffic processor due to subscriber bandwidth limitations.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

tpTotalNumWredDiscardedPackets (tpInfoEntry 20)

The accumulated number of packets discarded by the traffic processor due to congestion in the queues.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

tpTotalNumFragments (tpInfoEntry 21)

The accumulated number of fragmented packets handled by the traffic processor.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

tpTotalNumNonIpPackets (tpInfoEntry 22)

The accumulated number of non IP packets handled by the traffic processor.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

tpTotalNumIpCrcErrPackets (tpInfoEntry 23)

The accumulated number of packets with IP CRC error handled by the traffic processor.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

tpTotalNumIpLengthErrPackets (tpInfoEntry 24)

The accumulated number of packets with IP length error handled by the traffic processor.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

tpTotalNumIpBroadcastPackets (tpInfoEntry 25)

The accumulated number of IP broadcast packets handled by the traffic processor.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

tpTotalNumTtlErrPackets (tpInfoEntry 26)

The accumulated number of packets with TTL error handled by the traffic processor.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

tpTotalNumTcpUdpCrcErrPackets (tpInfoEntry 27)

The accumulated number of TCP/UDP packets with CRC error handled by the traffic processor.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

tpClearCountersTime (tpInfoEntry 28)

The time (in hundredths of a second) since the traffic processor statistics counters were last cleared. Writing a 0 to this object causes the RDR-formatter counters to be cleared.

Access RW

SYNTAX

TimeTicks

tpHandledPacketsRate (tpInfoEntry 29)

The rate in packets per second of the packets handled by this traffic processor..

Access RO

SYNTAX

Unsigned32 *(0... 4294967295)*

tpHandledPacketsRatePeak (tpInfoEntry 30)

The peak value of **tpHandledPacketsRate** since the last time it was cleared or the system started.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

tpHandledPacketsRatePeakTime (tpInfoEntry 31)

the time (in hundredths of a second) since the **tpHandledPacketsRatePeak** value occurred.

Access RO

SYNTAX

TimeTicks

tpHandledFlowsRate (tpInfoEntry 32)

The rate in flows start per second of the flows handled by this traffic processor.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

tpHandledFlowsRatePeak (tpInfoEntry 33)

The peak value of **tpHandledFlowsRate** since the last time it was cleared or the system started.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

tpHandledFlowsRatePeakTime (tpInfoEntry 34)

the time (in hundredths of a second) since the **tpHandledFlowsRatePeak** value occurred.

Access RO

SYNTAX

TimeTicks

tpCpuUtilization (tpInfoEntry 35)

The current percentage of CPU utilization

Access RO

SYNTAX

INTEGER *(1..100)*

tpCpuUtilizationPeak (tpInfoEntry 36)

The peak value of t**pCpuUtilization** since the last time it was cleared or the system started.

Access RO

SYNTAX

INTEGER *(1..100)*

tpCpuUtilizationPeakTime (tpInfoEntry 37)

The time (in hundredths of a second) since the **pCpuUtilizationPeak** value occurred.

Access RO

SYNTAX

TimeTicks

tpFlowsCapacityUtilization (tpInfoEntry 38)

The percentage of flows capacity utilization.

Access RO

SYNTAX

INTEGER *(1..100)*

tpFlowsCapacityUtilizationPeak (tpInfoEntry 39)

The peak value of tpFlowsCapacityUtilization since the last time it was cleared or the system started.

Access RO

SYNTAX

INTEGER *(1..100)*

tpFlowsCapacityUtilizationPeakTime (tpInfoEntry 40)

The time (in hundredths of a second) since the t**pFlowsCapacityUtilizationPeak** value occurred.

Access RO

SYNTAX

`TimeTicks`

tpServiceLoss (tpInfoEntry 41)

The relative amount of service loss in this traffic processor, in units of 0.001%, since last reboot or last time this counter was cleared.

Access RO

SYNTAX

`INTEGER` *`(1..100000)`*

Port Group: portGrp (pcubeSEObjs 10)

The Port group provides data regarding the port, such as its type and speed.

portTable (portGrp 1)

A list of port entries.

The number of entries is determined by the number of modules in the chassis and the number of ports on each module.

Access not-accessible

SYNTAX

Sequence of portEntry

portEntry (portTable 1)

Entry containing information for a specified port on a module.

Access not-accessible

INDEX

{portModuleIndex, portIndex}

SYNTAX

SEQUENCE *{*

portModuleIndex

portIndex

portType

ortNumTxQueues

portIfIndex

portAdminSpeed

portAdminDuplex

portOperDuplex

portLinkIndex

portOperStatus

}

portModuleIndex (portEntry 1)

An index value (**moduleIndex**) that uniquely identifies the module where the port is located.

Access RO

SYNTAX

INTEGER *(1..255)*

portIndex (portEntry 2)

An index value that uniquely identifies the port within the specified module. The value is determined by the location of the port on the module.

Valid entries are 1 to the value of **moduleNumPorts** for this module.

Access RO

SYNTAX

INTEGER *(1..255)*

portType (portEntry 3)

The type of physical layer medium dependent interface on the port.

Access RO

SYNTAX

INTEGER *{*

1 *(other):* none of the following

11 *(e100BaseTX):* UTP Fast Ethernet (Cat 5)

28 *(e1000BaseSX):* Short Wave fiber Giga Ethernet

}

portNumTxQueues (portEntry 4)

The number of transmit queues supported by this port.

Access RO

SYNTAX

INTEGER *(1..255)*

portIfIndex (portEntry 5)

The value of the instance of the ifIndex object, defined in MIB-II, for this port.

Access RO

SYNTAX

INTEGER *(1..255)*

portAdminSpeed (portEntry 6)

The desired speed of the port. The current operational speed of the port can be determined from ifSpeed.

Access RO

SYNTAX

INTEGER *{*

1 *(autoNegotiation):*

10000000 *(s10000000):* 10 Mbps

100000000 *(s100000000):* 100 Mbps

1000000000 *(s1000000000):* 1 Gbps

}

portAdminDuplex (portEntry 7)

The desired duplex of the port.

Access RO

SYNTAX

INTEGER *{*

1 *(half)*

2 *(full)*

4 *(auto)*

}

portOperDuplex (portEntry 8)

Indicates whether the port is operating in half-duplex or full-duplex.

Access RO

SYNTAX

INTEGER *{*

1 *(half)*

2 *(full)*

}

portLinkIndex (portEntry 9)

The **linkIndex** of the link to which this port belongs.

Value of 0 indicates that this port is not associated with any link.

Value of -1 indicates that this port is associated with multiple links.

Access RO

SYNTAX

INTEGER *(-1..255)*

portOperStatus (portEntry 10)

The status of the port. If the port is down, the reason is indicated.

Access RO

SYNTAX

INTEGER *{*

1 *(other):* none of the following

2 *(up):* the port is up

3 *(reflectionForcingDown):* the port is currently forced down due to the link reflection mechanism

4 *(redundancyForcingDown):* the port is currently forced down due to redundancy reasons

5 *(otherDown):* the port is down due to other reasons

}

Transmit Queues Group: txQueuesGrp (pcubeSEObjs 11)

The Transmit Queues group provides data regarding the transmit queue counters.

txQueuesTable (txQueuesGrp 1)

A list of information for each SCE transmit queue.

Access not-accessible

SYNTAX

Sequence of txQueuesEntry

txQueuesEntry (txQueuesTable 1)

Entry containing information for a specified SCE transmit queue.

Access not-accessible

INDEX

{txQueuesModuleIndex, txQueuesPortIndex, txQueuesQueueIndex}

SYNTAX

SEQUENCE *{*

txQueuesModuleIndex

txQueuesPortIndex

txQueuesQueueIndex

txQueuesDescription

txQueuesBandwidth

txQueuesUtilization

txQueuesUtilizationPeak

txQueuesUtilizationPeakTime

txQueuesClearCountersTime

}

txQueuesModuleIndex (txQueuesEntry 1)

An index value (**moduleIndex**) that uniquely identifies the module where the queue is located.

Access RO

SYNTAX

INTEGER *(1..255)*

txQueuesPortIndex (txQueuesEntry 2)

An index value that uniquely identifies the port on which the queue is located.

Access RO

SYNTAX

INTEGER *(1..255)*

txQueuesQueueIndex (txQueuesEntry 3)

An index value that uniquely identifies the queue within the specified port. The value is determined by the location of the queue on the port.

Valid entries are 1 to the value of **portNumTxQueues** for the specified port.

Access RO

SYNTAX

INTEGER *(1..255)*

txQueuesDescription (txQueuesEntry 4)

Description of the transmit queue.

Access RO

SYNTAX

DisplayString

txQueuesBandwidth (txQueuesEntry 5)

The bandwidth in kbps configured for this queue.

Access RO

SYNTAX

INTEGER *(1...1000000)*

txQueuesUtilization (txQueuesEntry 6)

The percentage of bandwidth utilization relative to the to the configured rate.

Access RO

SYNTAX

INTEGER *(0...100)*

txQueuesUtilizationPeak (txQueuesEntry 7)

The peak value of **txQueuesUtilization** since the last time it was cleared or the system started.

Access RO

SYNTAX

INTEGER *(0...100)*

txQueuesUtilizationPeakTime (txQueuesEntry 8)

The time (in hundredths of a second) since the **txQueuesUtilizationPeak** value occurred.

Access RO

SYNTAX

TimeTicks

txQueuesClearCountersTime (txQueuesEntry 9)

The time (in hundredths of a second) since the transmit queues statistics counters were last cleared.

Writing a 0 to this object causes the transmit queues counters to be cleared.

Access RW

SYNTAX

TimeTicks

txQueuesDroppedBytes (txQueuesEntry 10)

Number of dropped bytes. Valid only if the system is configured to count dropped bytes per TX queue.

Access RO

SYNTAX

Counter64

Global Controllers Group: globalControllersGrp (pcubeSEObjs 12)

The Global Controllers group provides data regarding the Global Controllers configuration and counters.

globalControllersTable (globalControllersGrp 1)

A list of information for each global controller.

Access not-accessible

SYNTAX

Sequence of globalControllersEntry

globalControllersEntry (globalControllersTable 1)

Entry containing information for a specified global controller.

Access not-accessible

INDEX

{globalControllersModuleIndex, globalControllersPortIndex, globalControllersIndex}

SYNTAX

SEQUENCE *{*

globalControllersModuleIndex

globalControllersPortIndex

globalControllersIndex

globalControllersDescription

globalControllersBandwidth

globalControllersUtilization

globalControllersUtilizationPeak

globalControllersUtilizationPeakTime

globalControllersClearCountersTime

globalControllersDroppedBytes

}

globalControllersModuleIndex (globalControllersEntry 1)

An index value (**moduleIndex**) that uniquely identifies the module where the Global Controller is located.

Access RO

SYNTAX

INTEGER *(1..255)*

globalControllersPortIndex (globalControllersEntry 2)

An index value that uniquely identifies the port on which the Global Controller is located.

Access RO

SYNTAX

INTEGER *(1..255)*

globalControllersIndex (globalControllersEntry 3)

An index value that uniquely identifies this Global Controller within the specified port.

Access RO

SYNTAX

INTEGER *(1..255)*

globalControllersDescription (globalControllersEntry 4)

Description of the Global Controller.

Access RO

SYNTAX

DisplayString

globalControllersBandwidth (globalControllersEntry 5)

The bandwidth in kbps configured for this Global Controller.

Access RO

SYNTAX

INTEGER *(1...1000000)*

globalControllersUtilization (globalControllersEntry 6)

The percentage of bandwidth utilization relative to the to the configured rate (**globalControllersBandwidth**).

Access RO

SYNTAX

INTEGER *(0...100)*

globalControllersUtilizationPeak (globalControllersEntry 7)

The peak value of **bwLimitersUtilization** since the last time it was cleared or the system started.

Access RO

SYNTAX

INTEGER *(0...100)*

globalControllersUtilizationPeakTime (globalControllersEntry 8)

The time (in hundredths of a second) since the **globalControllersUtilizationPeak** value occurred.

Access RO

SYNTAX

TimeTicks

globalControllersClearCountersTime (globalControllersEntry 9)

The time (in hundredths of a second) since the Global Controller statistics counters were last cleared.

Writing a 0 to this object causes the Global Controller counters to be cleared.

Access RW

SYNTAX

TimeTicks

globalControllersDroppedBytes (globalControllersEntry 10)

Number of dropped bytes. Valid only if the system is configured to count dropped bytes per global controller.

Access RO

SYNTAX

Counter64

Application Group: applicationGrp (pcubeSEObjs 13)

The Application group indicates which application is installed in the SCE Platform, and what the properties of the application and values of those properties are.

appInfoTable (applicationGrp 1)

Information identifying the application that is currently installed in the SCE Platform.

Access not-accessible

SYNTAX

Sequence of appInfoEntry

appInfoEntry (appInfoTable 1)

Entry containing identifying information for the application that is currently installed in the SCE Platform.

Access not-accessible

INDEX

{moduleIndex}

SYNTAX

SEQUENCE *{*

appName

appDescription

appVersion

}

appName (appInfoEntry 1)

Name of the application currently installed in the SCE Platform. This object returns an empty string if no application is currently installed.

Access RO

SYNTAX

DisplayString

appDescription (appInfoEntry 2)

Description of the application currently installed in the SCE Platform.

Access RO

SYNTAX

DisplayString

appVersion (appInfoEntry 3)

Version information for the application currently installed in the SCE Platform.

Access RO

SYNTAX

DisplayString

appPropertiesTable (applicationGrp 2)

List of all properties available for the application. The table is cleared when the application is unloaded.

Access not-accessible

SYNTAX

Sequence of appPropertiesEntry

appPropertiesEntry (appPropertiesTable 1)

Entry describing one of the properties available for the application.

Access not-accessible

INDEX

{moduleIndex, apIndex}

SYNTAX

SEQUENCE *{*

apIndex

apName

apType

}

apIndex (appPropertiesEntry 1)

An index value that uniquely identifies the property.

Access RO

SYNTAX

INTEGER *(1..255)*

apName (appPropertiesEntry 2)

Name of the property.

Access RO

SYNTAX

DisplayString

apType (appPropertiesEntry 3)

Property type in respect to: variable type (integer, boolean, string etc), number of elements (scalar or array), and restrictions, if any.

Access RO

SYNTAX

DisplayString

appPropertiesValuesTable (applicationGrp 3)

The applications properties value table is used to provide specific values for the applications properties.

An entry must be created by setting the entry apvRowStatus object with CreateAndGo (4) before setting the name of the property requested. The property requested must be one of the properties from the appPropertiesTable. To remove an entry set the apvRowStatus object with Destroy (6).

To poll the application property, any of these objects should be polled:

- apvPropertyValue
- apvPropertyUnitValue
- apvPropertyCounter64 object.

The table is cleared when the application is unloaded.

Access not-accessible

SYNTAX

Sequence of appPropertiesValueEntry

appPropertiesValueEntry (appPropertiesValueTable 1)

Entry providing information on the value of one of the specified application properties.

Access not-accessible

INDEX

{moduleIndex, apvIndex}

SYNTAX

SEQUENCE *{*

apvIndex

apvPropertyName

apvRowStatus

apvPropertyStringValue

apvPropertyUintValue

apvPropertyCounter64Value

}

apvIndex (appPropertiesValueEntry 1)

An index value that uniquely identifies the property.

Access RO

SYNTAX

INTEGER *(1.. 1024)*

apvPropertyName (appPropertiesValueEntry 2)

A name that uniquely identifies the application property.

Array-type properties may be accessed one element at a time in C-like format. (For example: x[1], or y[1][2])

Access RC

SYNTAX

DisplayString

apvRowStatus (appPropertiesValueEntry 3)

Controls creation of a table entry.

Access RC

SYNTAX

RowStatus

apvPropertyStringValue (appPropertiesValueEntry 4)

The value of the application property in display string format.

Access RO

SYNTAX

DisplayString *(SIZE 0...128)*

apvPropertyUintValue (appPropertiesValueEntry 5)

The value of the application property in Uint format.

If the property cannot be cast to Uint format, getting this object returns zero.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

apvPropertyCounter64Value (appPropertiesValueEntry 6)

The value of the application property in Counter64 format.

If the property cannot be cast to Counter64 format, getting this object returns zero.

Access RO

SYNTAX

Counter64

Traffic Counters Group: trafficCountersGrp (pcubeSEObjs 14)

The Traffic Counters group provides information regarding the value of different the traffic counters.

trafficCountersTable (trafficCountersGrp 1)

A list of information for each traffic counter.

Access not-accessible

SYNTAX

Sequence of trafficCountersEntry

trafficCountersEntry (trafficCountersTable 1)

Entry containing information for a specified traffic counter .

Access not-accessible

INDEX

{trafficCounterIndex}

SYNTAX

SEQUENCE {

trafficCounterIndex

trafficCounterValue

trafficCounterName

trafficCounterType

}

trafficCounterIndex (trafficCountersEntry 1)

An index value that uniquely identifies the counter.

Access RO

SYNTAX

INTEGER (*1..255)*

trafficCounterValue (trafficCountersEntry 2)

The 64 bit counter value.

Access RO

SYNTAX

Counter64

trafficCounterName (trafficCountersEntry 3)

The name of the counter.

Access RO

SYNTAX

DisplayString

trafficCounterType (trafficCountersEntry 4)

Defines whether the traffic counters counts by packets (3) or by bytes (2).

Access RO

SYNTAX

INTEGER {

1 (other): none of the following

2 (bytes): counts by bytes

3 (packets): counts by packets

}

Attack Group: attackGrp (pcubeSEObjs 15)

The Attack group provides information regarding detected attacks, aggregated by attack type

attackTypeTable (attackGrp 1)

A list of information for defined attack types.

Access not-accessible

SYNTAX

Sequence of AttackTypeEntry

attackTypeEntry (attackTypeTable 1)

Entry containing information for a specified attack type.

Access not-accessible

INDEX

{moduleIndex, attackTypeIndex}

SYNTAX

```
SEQUENCE {
attackTypeIndex
attackTypeName
attackTypeCurrentNumAttacks
attackTypeTotalNumAttacks
attackTypeTotalNumFlows
attackTypeTotalNumSeconds
}
```

attackTypeIndex (attackTypeEntry 1)

An index value that uniquely identifies the attack type.

Access RO

SYNTAX

INTEGER *(1..255)*

attackTypeName (attackTypeEntry 2)

The name of the attack type.

Access RO

SYNTAX

DisplayString

attackTypeCurrentNumAttacks (attackTypeEntry 3)

The number of attacks currently detected of this type.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

attackTypeTotalNumAttacks (attackTypeEntry 4)

The total number of attacks of this type detected since last clear.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

attackTypeTotalNumFlows (attackTypeEntry 5)

The total number of flows in attacks of this type detected since last clear.

Access RO

SYNTAX

Counter64

attackTypeTotalNumSeconds (attackTypeEntry 6)

The total duration (in seconds) of attacks of this type detected since last clear.

Access RO

SYNTAX

Unsigned32 *(0...4294967295)*

attackTypeTableClearTime (attackTypeTable 2)

The time (in hundredths of a second) since the attack type table was cleared.

Writing a 0 to this object causes the counters to be cleared.

Access RW

SYNTAX

`TimeTicks`

Supported Standards

SCE platform supports the SNMP related standards listed in the following table.

Table A-1 Supported SNMP Standards

Document Name	Description
RFC 1155: Structure and Identification of Management Information for TCP/IP-based Internets	K. McCloghrie and M. T. Rose, (May 1990). Contains MIB object definitions. (Obsoletes RFC 1065)
RFC 1157: A Simple Network Management Protocol	J. D. Case, M. Fedor, M. L. Schoffstall, and C. Davin, (May 1990). Defines SNMP. (Obsoletes RFC 1098)
RFC 1212: Concise MIB Definitions	K. McCloghrie (March 1991). Defines a format for producing MIB modules
RFC 1213: Management Information Base Network Management of TCP/IP based internets: MIB-II	K. McCloghrie and M. T. Rose, eds., (March 1991). Defines MIB-II. (Obsoletes RFC 1158)
RFC 1215: Convention for Defining Traps for Use with the SNMP	M. T. Rose, ed. (March 1991).
RFC 1901: Introduction to Community-based SNMPv2	SNMPv2 WG, J.Case, K. McCloghrie, M.T.Rose, S. Waldbusser, (January 1996). Defines “Community-based SNMPv2.” (Experimental. Obsoletes RFC 1441)
RFC 1905: Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)	Obsoletes: 1448 (January 1996)
RFC 1906: Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)	Obsoletes: 1449 (January 1996)

Glossary of Terms

A

Access Control List (ACL)

Permits or denies incoming connections on any of the management interfaces. It is an ordered list of entries, each consisting of an IP address and an optional wildcard "mask" defining an IP address range, and a permit/deny field.

Active/Standby Device

The terms *Active* and *Standby* refer to the actual current status of a particular ***SCE*** platform, as opposed to *Primary* and *Secondary,* which refer to the configured default status.

Auto-negotiation

Ethernet auto-negotiation allows the link to synchronize with its peer automatically.

B

Bump-in-the-wire topology

The SCE Platform physically resides on the data link between the subscriber side and the network side, and can both receive and transmit traffic.

Bypass module

Hardware mechanism that forwards traffic with no intervention of the application running in the SCE platform.

The ***SCE*** platform is equipped with a hardwired bypass that acts as a physical shortcut between the two network elements to which the ***SCE*** platform is connected. It forwards traffic with no intervention of the application running in the ***SCE*** platform, either for control or for monitoring.

C

Cascade Ports

The ports that are used to connect two ***SCE*** platforms in a two-***SCE*** platform, redundant topology. All traffic and management information is exchanged via these ports.

Collection Manager (CM)

A software application running on the SCE Platform that is responsible for receiving RDRs from SCE Platforms and processing them.

Command Line Interface (CLI)

One of the management interfaces to the SCE Platform. It is accessed through a Telnet session or directly via the console port on the front panel of the SCE Platform.

Cutoff

Mechanism that cuts the link so that there is no forwarding of traffic, and the physical link is forced down (cutoff at layer 1).

D

DDoS Attack Filtering

The aim of DDoS attack filtering is to detect attacks that occur in the traffic flowing through the SCE platform, to report such attacks via management channels, and to handle these attacks by blocking them, if configured to do so. In addition, if the application loaded to the ***SCE*** platform supports the 'subscriber-notification' feature, a subscriber whose IP address is associated with an attack that was identified can be notified about the attack on-line by the ***SCE*** platform.

There are two main aspects of attack filtering:

- Attack detection: Detect attacks based on their common IP address and number of flows found to/from that IP address.
- Attack handling: Attack flows may be blocked or processed as usual.

In addition, a subscriber associated with the attack may be notified about the attack.

The ***SCE*** platform maintains a list of the most active IP addresses flowing through it, with a measure of the activity of each IP address. (Activity is measured by number of flows opened to/from that address). If there are IP addresses in the table whose number of flows is above the configured threshold, these IP addresses are assumed to be attacking, or being attacked. If the ***SCE*** platform is configured to block the attack, it drops the attack packets.

Duplex

Duplex refers to the bi-directional capacity of the link, that is, the link can both receive and transmit.

Full duplex data transmission means that data can be received and transmitted simultaneously.

Half duplex data transmission means that the line can transmit in only one direction at a time. When data is being transmitted, it cannot be received and vice versa.

F

Fail-over

In fail over solution, the two ***SCE*** platforms exchange keep alive messages via the cascade ports. This keep alive mechanism enables fast detection of failures between the ***SCE*** platforms and fast fail over to the standby ***SCE*** platform when required.

Flow

All packets travelling in both directions on a single application layer connection (such as a TCP or UDP connection). A flow is identified by the tuple information: <Source IP, Destination IP, Source Port, Destination Port, IP Protocol>. (Note that if the IP protocol is neither TCP or UDP, the port number is defined as '0'.)

IN this guide, the term 'flow' represents bi-directional flows (packets from both the client and server of each connection). When referencing a uni-directional flow, this is explicitly mentioned.

Flow Bundle

A group of one or more flows comprising the set of application-layer connections (such as a TCP or UDP connection) used in a single, logical application session. The semantics of flow-bundles are application dependant, and relate to the way each application spawns and negotiates additional flows as part of a single session. A few common examples are:

- An SIP (VoIP) flow bundle comprises the signaling flow as well as all the RTP/UDP flows containing the actual media data (voice).
- An RTSP (Streaming) flow bundle comprises the signaling flow as well as the RTP/UDP flows containing the audio and/or video transmissions.
- AN FTP (file transfer) flow bundle comprises the control flow (used to login an FTP server) and the actual file-transfer flows

In each of these cases, the ***SCE*** platform tracks the application communication to identify new connections created and bundle them into a single context. This is important for classification and accounting purposes, as otherwise these spawned flows would be unclassifiable.

H

Hot Standby

When two ***SCE*** platforms are deployed in a fail over topology, one ***SCE*** platform is active, while the second ***SCE*** platform is in standby, receiving from the active ***SCE*** platform all subscriber state updates and keep alive messages.

I

Inline connection mode

The ***SCE*** platform physically resides on the data link between the subscriber side and the network side, and can both receive and transmit traffic.

L

Link mode

A specified behavior that may be enforced on the link. This may be useful during installation and for debugging the network.

The available link modes are:

- forwarding
- sniffing
- bypass
- cutoff

P

Physically-connected-link

The number of the link (link 0 or Link 1) physically connected to an ***SCE*** platform. (Dual-***SCE*** platform configurations only)

PQI (Cisco Installation) File

An application package file that is installed on the network SCE Platforms and Collection Managers.

Primary/Secondary Device

The terms *Primary* and *Secondary* refer to the default status of a particular ***SCE*** platform. The Primary SCE platform is active by default, while the Secondary ***SCE*** platform is the default standby.

R

Raw Data Record (RDR)

A data record produced by the SCE Platform that reports on events in the traffic. RDRs produced by the SCE Platform are sent to the Collection Manager and then stored in the Collection Manager database or forwarded to third-party systems.

RDR Formatter

An internal component of the SCE Platform that gathers the Raw Data Records (RDRs), formats them, and sends them to an external Collection Manager.

Receive-only connection mode

The SCE Platform does not reside physically on the data link, and therefore can only receive data and not transmit.

S

SCE Platform

The SCE Platform is a purpose-built service component and active enforcing system designed for enhancing service providers and backbone carrier networks. By identifying, classifying, and manipulating complex traffic flows at wire-speed, the SCE Platform transforms simple transport networks into differentiated service delivery infrastructures for a wide variety of value-added IP applications, such as video streaming, VoIP, tiered services, and bilateral application-level SLAs.

The SCE Platform seamlessly interfaces with existing network elements—including routers, switches, aggregators, subscriber management devices, and operational support systems—using industry standard interfaces and communications protocols.

The need to guarantee that packets passing through the network are processed at the rate they arrive makes it necessary to provide a custom-made hardware solution.

The SCE Platform comes in three models: SCE 1000, SCE 2000 4xGBE, and SCE 2000 4/8xFE. There may be one or more of the SCE Platforms in the provider network. Within the SCE Platforms, network transactions are analyzed and mapped to services that enforce the provider's policies.

In addition, the SCE Platform implements the business logic of the system solution and performs the transaction analysis in real time. When so instructed, the SCE Platform creates a Raw Data Record (RDR) to be sent for storage to the system's data repository, the Collection Manager (CM); or carries out some other operation such as bandwidth and volume control.

SCMS Application

An SML program that determines how the ***SCE*** platform operates.

Service Control

The Cisco basic concept for enabling service providers to differentiate subscribers, detect real-time events, create premium services, actively control applications, and leverage their existing infrastructure.

SLI (SML Loadable Image) File

An SLI file is a software package that contains the SML application that is loaded onto a SCE Platform. The SML application determines the behavior of the SCE Platform. Different SCE Platforms can have different SML applications, even when they are within the same POP.

Subscriber Manager (SM)

A middleware software component used in cases where dynamic binding of subscriber information and service configurations is required. The SM manages subscriber information and provisions it in real time to multiple SCE Platforms. The SM can store subscriber service configurations information internally, and act as a state-full bridge between the AAA system (for example, RADIUS and DHCP) and the SCE Platforms.

SML (Service Modeling Language)

The Cisco scripting language, which enables the definition of service-related events and the execution of actions on those events.

Sniffing

A link mode that allows the ***SCE*** platform to forward traffic on the specified link through the bypass, while still analyzing the traffic passively.

Split flow

The splitting the packets from any one micro flow between two links. May occur when two links are used to provide redundancy.

splitter or switch topology

In this topology, the ***SCE*** platform does not reside physically on the data link. Data is forwarded to the ***SCE*** platform via an splitter or switch, which splits the traffic on the link, sending all information to the ***SCE*** platform in parallel with its transmission through the splitter or switch. The splitter or switch is connected physically on the Ethernet link and only the receive inputs of the data link Ethernet interfaces in the ***SCE*** platform are connected to the splitter or switch.

Subscriber

A Service Provider's client. There are two types of subscribers:

- Introduced Subscriber: A specific customer with an externally generated name. Maybe mapped to more than one IP address.
- Anonymous subscriber group: A subscriber with an internally generated name, generated automatically by the ***SCE*** platform according to an anonymous subscriber group specification. Always mapped to a single IP address. The actual identity of the customer(s) is unknown to the system.

T

TIR (Traffic Processor IP Range)

IP addresses (IP range and/or specific IP address) that are explicitly assigned to a specified traffic processor.

TIRs can be used in a cable environment to ensure that all IP addresses of each subscriber will actually be handled by the same traffic processor.

Tunneling protocols

A tunneling protocol adds headers to the basic protocol stack in order to route the packet across the network. Therefore, the system must be configured to recognize and either process or skip the additional tunnel headers as necessary.

W

Warm start

Restarting the computer by performing a reset operation.

Index

A

Access Control List (ACL) • 1
Access control lists • 5-1
Active/Standby Device • 1
Adding Entries to an Access List • 5-3
Adding IP Routing Entry to Routing Table • 5-23
Aging Subscribers • 9-4
Anonymous Groups and Subscriber Templates • 9-5
apIndex (appPropertiesEntry 1) • A-72
apName (appPropertiesEntry 2) • A-72
appDescription (appInfoEntry 2) • A-71
appInfoEntry (appInfoTable 1) • A-71
appInfoTable (applicationGrp 1) • A-70
Application
  configuring • 3-8
  installing • 3-9
  upgrading • 3-9
  upgrading (cascade topology) • 10-12
Application Group
  applicationGrp (pcubeSEObjs 13) • A-70
Application Upgrade • 10-12
appName (appInfoEntry 1) • A-71
appPropertiesEntry (appPropertiesTable 1) • A-72
appPropertiesTable (applicationGrp 2) • A-72
appPropertiesValueEntry (appPropertiesValueTable 1) • A-73
appPropertiesValuesTable (applicationGrp 3) • A-73
appVersion (appInfoEntry 3) • A-71
apType (appPropertiesEntry 3) • A-73
apvIndex (appPropertiesValueEntry 1) • A-74
apvPropertyCounter64Value (appPropertiesValueEntry 6) • A-75
apvPropertyName (appPropertiesValueEntry 2) • A-74
apvPropertyStringValue (appPropertiesValueEntry 4) • A-74
apvPropertyUintValue (appPropertiesValueEntry 5) • A-74
apvRowStatus (appPropertiesValueEntry 3) • A-74
Argument Help • 2-14
Associating an Access List to Telnet Interface • 5-5
Attack detection • 11-1
  configuring • 11-5
  managing • 11-12
  monitoring • 11-14
  subscriber notification • 11-4, 11-11
  thresholds • 11-3
Attack Detection • 11-2
Attack Detection Thresholds • 11-3
Attack detectors
  configuring • 11-5
  default • 11-7
  sample configuration • 11-10
  specific • 11-8
Attack Filtering • 11-1
Attack Group
  attackGrp (pcubeSEObjs 15) • A-76
Attack Handling • 11-3
attackTypeCurrentNumAttacks (attackTypeEntry 3) • A-77
attackTypeEntry (attackTypeTable 1) • A-77
attackTypeIndex (attackTypeEntry 1) • A-77
attackTypeName (attackTypeEntry 2) • A-77
attackTypeTable (attackGrp 1) • A-76

attackTypeTableClearTime (attackTypeTable 2) • A-78
attackTypeTotalNumAttacks (attackTypeEntry 4) • A-78
attackTypeTotalNumFlows (attackTypeEntry 5) • A-78
Audience • xvii
Authorization and Command Levels (Hierarchy) • 2-2
Auto-negotiation • 6-12, 1

B

Bump-in-the-wire topology • 1
Bypass module • 1

C

Cascade Ports • 1
cascaded topology • 10-3
  application upgrades for • 10-12
  CLI commands for • 10-9
  firmware upgrades for • 10-11
  installation procedure • 10-6
  recovery • 10-7
  replacing the SCE platform in • 10-8
  system upgrades for • 10-11
Categories • 8-2
Changing Directories • 4-4
Changing Passwords • 5-20
Chassis Group
  chassisGrp (pcubeSEObjs 2) • A-19
chassisFansAlarm (chassisGrp 3) • A-20
chassisFansAlarmOnTrap (pcubeSeEvents 8) • A-14
chassisLineFeedAlarm (chassisGrp 9) • A-22
chassisLineFeedAlarmOnTrap (pcubeSeEvents 36) • A-18
chassisNumSlots (chassisGrp 6) • A-21
chassisPowerSupplyAlarm (chassisGrp 2) • A-20
chassisPowerSupplyAlarmOnTrap (pcubeSeEvents 9) • A-15
chassisPsuType (chassisGrp 8) • A-22
chassisSlotConfig (chassisGrp 7) • A-21
ChassisSysType (chassisGrp 1) • A-20
chassisTempAlarm (chassisGrp 4) • A-21
chassisTempAlarmOffTrap (pcubeSeEvents 6) • A-14
chassisTempAlarmOnTrap (pcubeSeEvents 5) • A-14
chassisVoltageAlarm (chassisGrp 5) • A-21
chassisVoltageAlarmOnTrap (pcubeSeEvents 7) • A-14
Cisco Enterprise MIB • 5-18
Cisco Technical Support Website • xxi
Cisco.com • xx
Clearing the User Log • 4-10
CLI • 5-16
CLI (Command Line Interface) • 2-1
  authorization levels • 2-2, 2-10
  command hierarchy • 2-3
  for cascaded topology • 10-9
  help features • 2-13
  scrpts • 2-18
  shortcuts • 2-15
CLI Authorization Levels • 2-10
CLI Command Hierarchy • 2-3
CLI Commands • 10-9
CLI Help Features • 2-13
CLI Scripts • 2-18
Collection Manager (CM) • 1
Command History • 2-15
Command Line Interface (CLI) • 1
Command-Line Interface • 2-1
Community strings • 5-10
Config-Copy MIB Objects • A-3
Configuration
  IP • 5-22
  recovering • 3-6
  saving • 3-4
  viewing • 3-1
Configuration via SNMP • 5-10
Configuring Access Control Lists (ACLs) • 5-1
Configuring Applications • 3-8
Configuring Attack Detectors • 11-5
Configuring Daylight Saving Time • 5-29
Configuring GigabitEthernet Auto-Negotiation • 6-12
Configuring IP Advertising • 5-25
Configuring IP Tunnels • 6-2
Configuring SNMP Community Strings • 5-11
Configuring Subscriber Notifications • 11-11
Configuring the Available Interfaces • 5-1
Configuring the Connection • 7-1
Configuring the Connection Mode • 10-9
Configuring the Currently Installed Application • 3-11

Configuring the Duplex Operation of the FastEthernet Interface • 5-37
Configuring the Fast Ethernet Line Ports • 2-7
Configuring the Gigabit Ethernet Line Ports • 2-7
Configuring the L2TP Environment • 6-3
Configuring the Line Interface • 6-1
Configuring the Management Interface and Security • 5-1
Configuring the Management Interface Speed and Duplex Parameters • 5-37
Configuring the Management Port • 2-7
Configuring the MPLS Environment • 6-3
Configuring the Physical Ports • 2-6
Configuring the RDR Formatter • 8-1, 8-4
Configuring the RDR Formatter Categories • 8-5
Configuring the RDR Formatter Destinations • 8-5
Configuring the Speed of the FastEthernet Interface • 5-37
Configuring the VLAN Environment • 6-2
Configuring TIRs • 9-20
Configuring TOS Marking • 6-11
Configuring Traffic Counters • 6-6
Configuring Traffic Rules • 6-6
Configuring Traffic Rules and Counters • 6-4
Configuring Traps • 5-13
Configuring Tunneling Protocols • 6-1
Connection mode • 7-1, 10-9
Copying a File • 4-6
Copying the User Log • 4-8
Creating a Directory • 4-3
Cutoff • 2

D

DDoS attack detection • 11-1, 11-2
  configuring • 11-5
  managing • 11-12
  monitoring • 11-14
  threholds • 11-3
DDoS Attack Filtering • 2
Default Attack Detector • 11-7
Default Gateway • 5-23
Defining the Global Access List • 5-4
Defining the SNTP unicast update interval • 5-33
Definitions of Service Request Severity • xxi
Deleting a Directory • 4-3
Deleting a File • 4-5
Directories, working with • 4-3
Disable sending RDRs • 8-10
Disabling SNMP • 5-8
Disabling SNTP multicast client • 5-32
Disabling SNTP unicast client • 5-33
Disabling the LineCard from Sending RDRs • 8-10
Disk Group
  diskGrp (pcubeSEObjs 5) • A-30
diskNumFreeBytes (diskGrp 2) • A-31
diskNumUsedBytes (diskGrp 1) • A-31
Display SNTP information • 5-33
Displaying Anonymous Subscriber Information • 9-16
Displaying File Contents • 4-7
Displaying RDR Formatter Configuration and Statistics • 8-9
Displaying Subscriber Information • 9-15
Displaying Subscribers • 9-12
  By IP Address or VLAN ID • 9-14
  By Subscriber Property or Prefix • 9-12
Displaying Tunneling Configuration • 6-4
Displaying Working Directory • 4-4
Document
  conventions • xix
Document Conventions • xix
Documentation Feedback • xx
Domain Name • 5-35
Domain Name (DNS) Settings • 5-34
Domain Name Server (DNS) • 5-34
Duplex • 5-37, 2

E

Editing the Connection Mode • 7-1
Enabling and Disabling Link Failure Reflection • 7-5
Enabling and Disabling Link Failure Reflection on All Ports • 7-5
Enabling and Disabling the User Log • 4-8
Enabling and Disabling TOS Marking • 6-11
Enabling SNMP • 5-8
Enabling SNTP multicast client • 5-32
Enabling SNTP unicast client • 5-32
Enabling Specific-IP Detection • 11-7
Encryption • 5-22
Entering and Exiting Global Configuration Mode • 2-5

Entering Ethernet Line Interface Configuration Mode • 2-8, 6-12
Entering FastEthernet (Management) Interface Configuration Mode • 2-7
Entering LineCard Interface Configuration Mode • 2-8
Entering the Fast Ethernet Line Interface Configuration Mode • 2-8
Entering the Gigabit Ethernet Line Interface Configuration Mode • 2-9
Entering the Setup Utility • 4-1
Ethernet interface
auto-negotiation • 6-12
configuring • 2-8, 6-12
Exiting Modes • 2-10

F

Fail-over • 10-5, 2
Failure
detection • 10-3
forced • 7-3
Failure Detection • 10-3
Failure in the Cascade Connection • 10-6
Failure Recovery Mode • 7-4
Fast Ethernet interface
configuring • 2-7, 5-37
duplex • 5-37
speed • 5-37
File-system Operations • 4-3
Filtering Command Output • 2-17
Firmware Upgrade (package installation) • 10-11
Flow • 2
Flow Bundle • 3
Forced Failure • 7-3, 10-10
Forcing Attack Filtering • 11-13
Forwarding Modes • 8-4
FTP User Name and Password • 2-16

G

Generating a File for Technical Support • 4-10
Getting Help • 2-2
Global Configuration Mode Commands • 5-17
Global Controllers Group
globalControllersGrp (pcubeSEObjs 12) • A-67
globalControllersBandwidth (globalControllersEntry 5) • A-69
globalControllersClearCountersTime (globalControllersEntry 9) • A-70
globalControllersDescription (globalControllersEntry 4) • A-69
globalControllersDroppedBytes (globalControllersEntry 10) • A-70
globalControllersEntry (globalControllersTable 1) • A-68
globalControllersIndex (globalControllersEntry 3) • A-69
globalControllersModuleIndex (globalControllersEntry 1) • A-68
globalControllersPortIndex (globalControllersEntry 2) • A-68
globalControllersTable (globalControllersGrp 1) • A-67
globalControllersUtilization (globalControllersEntry 6) • A-69
globalControllersUtilizationPeak (globalControllersEntry 7) • A-69
globalControllersUtilizationPeakTime (globalControllersEntry 8) • A-69

H

Host Table • 5-36
Hot standby • 10-4
Hot Standby • 10-4, 3
Hot Standby and Fail-over • 10-4

I

Identifying And Preventing Distributed-Denial-Of-Service Attacks • 11-1
Importing and Exporting TIRs • 9-23
Importing/Exporting Anonymous Groups • 9-10
Importing/Exporting Subscriber Information • 9-6
Importing/Exporting Subscriber Templates • 9-7
Importing/Exporting Subscribers • 9-7
Inline connection mode • 3
In-line Dual Link Redundant Topology • 10-3
Installing a Cascaded System • 10-6
Installing an Application • 3-9
interface

configuring • 5-1
SCE platform management • 1-2
SNMP • 5-8
telnet • 5-4
Interface Configuration Modes • 2-6
Introduction • 1-1
IP
address of management interface • 5-26
advertising • 5-24
configuration • 5-22
routing • 5-22
IP Advertising • 5-24
IP Configuration • 5-22
IP Routing Table • 5-22

K

Key Management • 5-6
Keyboard Shortcuts • 2-15

L

Line Ethernet Interfaces • 6-12
Link
failure reflection parameter • 10-4
mode • 7-2
physically connected • 10-9
Link failure reflection • 10-4
Link Failure Reflection • 10-4
Link Group
linkGrp (pcubeSEObjs 4) • A-27
Link mode • 3
Link Mode • 7-2
linkAdminModeOnActive (linkEntry 3) • A-29
linkAdminModeOnFailure (linkEntry 4) • A-29
linkEntry (linkTable 1) • A-28
linkIndex (linkEntry 2) • A-29
linkModeBypassTrap (pcubeSeEvents 20) • A-15
linkModeCutoffTrap (pcubeSeEvents 22) • A-15
linkModeForwardingTrap (pcubeSeEvents 21) • A-15
linkModeSniffingTrap (pcubeSeEvents 28) • A-17
linkModuleIndex (linkEntry 1) • A-28
linkOperMode (linkEntry 5) • A-29
linkStatusReflectionEnable (linkEntry 6) • A-30
linkSubscriberSidePortIndex (linkEntry 7) • A-30
linkSubscriberSidePortIndex (linkEntry 8) • A-30
linkTable (linkGrp 1) • A-28
Listing Files in Current Directory • 4-4
Loading the MIB Files • 5-19
Logger Group
loggerGrp (pcubeSEObjs 7) • A-40
loggerUserLogClearCountersTime (loggerGrp 6) • A-41
loggerUserLogEnable (loggerGrp 1) • A-40
loggerUserLogIsFullTrap (pcubeSeEvents 18) • A-15
loggerUserLogNumError (loggerGrp 4) • A-41
loggerUserLogNumFatal (loggerGrp 5) • A-41
loggerUserLogNumInfo (loggerGrp 2) • A-40
loggerUserLogNumWarning (loggerGrp 3) • A-41
Login and User Levels • 2-13

M

Management Interface Configuration Mode • 5-37
Managing Attack Filtering • 11-12
Managing Command Output • 2-17
Managing Configurations • 3-1
Managing Subscribers • 9-1
Managing the SSH Server • 5-6
Managing Traffic Rules and Counters • 6-9
MIB • 5-17, A-1
MIB-II • 5-17
objects • A-5
reference • A-1
Service Control enterprise • 5-18, A-1
structure • A-7
MIB-II • 5-17
MIBs • 5-17
Modifying the TOS Table • 6-12
Module Group
moduleGrp (pcubeSEObjs 3) • A-22
moduleAdminStatus (moduleEntry 15) • A-27
moduleAttackFilterActivatedTrap (pcubeSeEvents 25) • A-16
moduleAttackFilterDeactivatedTrap (pcubeSeEvents 26) • A-16

moduleAttackObjectsClearTime (moduleEntry 14) • A-27
moduleConnectionMode (moduleEntry 8) • A-25
moduleDownStreamAttackFilteringTime (moduleEntry 12) • A-26
moduleDownStreamLastAttackFilteringTime (moduleEntry 13) • A-26
moduleEmAgentGenericTrap (pcubeSeEvents 27) • A-17
moduleEntry (moduleTable 1) • A-23
moduleHwVersion (moduleEntry 5) • A-25
moduleIndex (moduleEntry 1) • A-23
moduleLostRedundancyTrap (pcubeSeEvents 31) • A-17
moduleNumLinks (moduleEntry 7) • A-25
moduleNumPorts (moduleEntry 6) • A-25
moduleNumTrafficProcessors (moduleEntry 3) • A-24
moduleOperStatus (moduleEntry 16) • A-27
moduleOperStatusChangeTrap (pcubeSeEvents 34) • A-18
moduleRedundancyReadyTrap (pcubeSeEvents 29) • A-17
moduleRedundantConfigurationMismatchTrap (pcubeSeEvents 30) • A-17
moduleSerialNumber (moduleEntry 9) • A-26
moduleSlotNum (moduleEntry 4) • A-24
moduleSmConnectionDownTrap (pcubeSeEvents 32) • A-17
moduleSmConnectionUpTrap (pcubeSeEvents 33) • A-18
moduleTable (moduleGrp 1) • A-22
moduleType (moduleEntry 2) • A-24
moduleUpStreamAttackFilteringTime (moduleEntry 10) • A-26
moduleUpStreamLastAttackFilteringTime (moduleEntry 11) • A-26
Monitoring Attack Filtering • 11-14
Monitoring Subscribers • 9-10
Monitoring the Status of the SSH Server • 5-7
Monitoring the Subscriber Database • 9-11
Monitoring the System • 10-10
Monitoring TIRs • 9-24
Multiple entry parameters (Lists) • 4-2

N

Name Servers • 5-35
Navigating between the Interface Configuration Modes • 2-9
Navigational and Shortcut Features • 2-15

O

Obtaining Additional Publications and Information • xxii
Obtaining Documentation • xx
Obtaining Technical Assistance • xx
operationalStatusFailureTrap (pcubeSeEvents 3) • A-14
operationalStatusOperationalTrap (pcubeSeEvents 1) • A-14
operationalStatusWarningTrap (pcubeSeEvents 2) • A-14
Operations • 3-1
Organization • xvii

P

Partial Help • 2-13
Passwords • 5-20
pcubeCopyDestFileType (pcubeCopyEntry 4) • A-3
pcubeCopyEntryRowStatus (pcubeCopyEntry 2) • A-3
pcubeCopyIndex (pcubeCopyEntry 1) • A-3
pcubeCopySourceFileType (pcubeCopyEntry 3) • A-3
pcubeMgmt
pcubeConfigCopyMIB • A-2
pcubeSeEventGenericString1 (pcubeSeEvents 23) • A-15
pcubeSeEventGenericString2 (pcubeSeEvents 24) • A-16
pcubeSeEvents (pcubeWorkgroup 0) • A-4
pcubeSEObjs (pcubeWorkgroup 1) • A-5
pcubeWorkgroup
pcubeSeMIB • A-4
Physically connected link • 10-9
Physically-connected-link • 3
Port Group
portGrp (pcubeSEObjs 10) • A-60
portAdminDuplex (portEntry 7) • A-63
portAdminSpeed (portEntry 6) • A-62
portEntry (portTable 1) • A-61
portIfIndex (portEntry 5) • A-62
portIndex (portEntry 2) • A-61
portLinkIndex (portEntry 9) • A-63
portModuleIndex (portEntry 1) • A-61
portNumTxQueues (portEntry 4) • A-62

portOperDuplex (portEntry 8) • A-63
portOperStatus (portEntry 10) • A-64
portOperStatusChangeTrap (pcubeSeEvents 35) • A-18
portTable (portGrp 1) • A-60
portType (portEntry 3) • A-62
PQI (Cisco Installation) File • 3
Preface • xvii
Preventing Attack Filtering • 11-13
Preventing Telnet Access • 5-4
Primary/Secondary Device • 3
Priority • 8-3, 10-9
Privileged Exec Mode Commands • 5-16
Prompt Indications • 2-12
Proprietary MIB Reference • A-1

R

Raw Data Record (RDR) • 3
RDR formatter
  configuring • 8-4
RDR Formatter • 4
RDR Formatter Destinations • 8-1
RDR Formatter Group
  rdrFormatterGrp (pcubeSEObjs 6) • A-31
rdrActiveConnectionTrap (pcubeSeEvents 10) • A-15
rdrConnectionDownTrap (pcubeSeEvents 13) • A-15
rdrConnectionUpTrap (pcubeSeEvents 12) • A-15
rdrFormatterCategoryDestEntry (rdrFormatterCategoryDestTable 1) • A-39
rdrFormatterCategoryDestPriority (rdrFormatterCategoryDestEntry 1) • A-39
rdrFormatterCategoryDestStatus (rFormatterCategoryDestEntry 2) • A-40
rdrFormatterCategoryDestTable (rdrFormatterGrp 12) • A-39
rdrFormatterCategoryDiscardingReportsTrap (pcubeSeEvents 37) • A-18
rdrFormatterCategoryEntry (rdrFormatterCategoryTable 1) • A-37
rdrFormatterCategoryIndex (rdrFormatterCategoryEntry 1) • A-37
rdrFormatterCategoryName (rdrFormatterCategoryEntry 2) • A-37
rdrFormatterCategoryNumReportsDiscarded (rdrFormatterCategoryEntry 4) • A-38
rdrFormatterCategoryNumReportsQueued (rdrFormatterCategoryEntry 8) • A-38
rdrFormatterCategoryNumReportsSent (rdrFormatterCategoryEntry 3) • A-37
rdrFormatterCategoryReportRate (rdrFormatterCategoryEntry 5) • A-38
rdrFormatterCategoryReportRatePeak (rdrFormatterCategoryEntry 6) • A-38
rdrFormatterCategoryReportRatePeakTime (rdrFormatterCategoryEntry 7) • A-38
rdrFormatterCategoryStoppedDiscardingReportsTrap (pcubeSeEvents 38) • A-18
rdrFormatterCategoryTable (rdrFormatterGrp 11) • A-36
rdrFormatterClearCountersTime (rdrFormatterGrp 5) • A-35
rdrFormatterDestConnectionStatus (rdrFormatterDestEntry 5) • A-33
rdrFormatterDestEntry (rdrFormatterDestTable 1) • A-32
rdrFormatterDestIPAddr (rdrFormatterDestEntry 1) • A-32
rdrFormatterDestNumReportsDiscarded (rdrFormatterDestEntry 7) • A-34
rdrFormatterDestNumReportsSent (rdrFormatterDestEntry 6) • A-33
rdrFormatterDestPort (rdrFormatterDestEntry 2) • A-32
rdrFormatterDestPriority (rdrFormatterDestEntry 3) • A-33
rdrFormatterDestReportRate (rdrFormatterDestEntry 8) • A-34
rdrFormatterDestReportRatePeak (rdrFormatterDestEntry 9) • A-34
rdrFormatterDestReportRatePeakTime (rdrFormatterDestEntry 10) • A-34
rdrFormatterDestStatus (rdrFormatterDestEntry 4) • A-33
rdrFormatterDestTable (rdrFormatterGrp 2) • A-31
rdrFormatterEnable (rdrFormatterGrp 1) • A-31
rdrFormatterForwardingMode (rdrFormatterGrp 10) • A-36
rdrFormatterNumReportsDiscarded (rdrFormatterGrp 4) • A-34
rdrFormatterNumReportsSent (rdrFormatterGrp 3) • A-34
rdrFormatterProtocol (rdrFormatterGrp 9) • A-35

rdrFormatterReportRate (rdrFormatterGrp 6) • A-35
rdrFormatterReportRatePeak (rdrFormatterGrp 7) • A-35
rdrFormatterReportRatePeakTime (rdrFormatterGrp 8) • A-35
rdrNoActiveConnectionTrap (pcubeSeEvents 11) • A-15
Reboot only (fully automatic recovery) • 10-8
Rebooting and Shutting Down the SCE Platform • 3-12
Rebooting the SCE Platform • 3-12
Receive-only connection mode • 4
Recovering a Previous Configuration • 3-6
Recovery • 10-7
Redirecting Command Output to a File • 2-18
Redundancy
  and failover • 10-1
Redundancy and Fail-Over • 10-1
Redundant Topologies • 10-2
Related Publications • xviii
Removing an Access List • 5-3
Removing Current Time Zone Setting • 5-29
Removing Subscribers and Templates • 9-8
Removing TIRs and Subscriber Mappings • 9-21
Renaming a File • 4-5
Replacing the SCE platform (manual recovery) • 10-8
Reserving Rules for TIRs • 9-20

S

Sample Attack Detector Configuration • 11-10
Saving the Configuration Settings • 3-4
SCE Events • A-4
  pcubeSeEvents • A-14
SCE Platform • 4
SCE Platform Management Interfaces • 1-2
SCE Platform/SM Connection • 7-4, 9-27
SCE-MIB Objects • A-5
SCE-MIB Structure • A-7
SCMS Application • 4
Scrolling the Screen Display • 2-17
Security considerations • 5-10
Security Considerations • 5-10
Selecting the Tunneling Mode • 6-2
Service Control • 5
Service Control Enterprise MIB • A-1
sessionBadLoginTrap (pcubeSeEvents 42) • A-18
sessionDeniedAccessTrap (pcubeSeEvents 41) • A-18
sessionEndedTrap (pcubeSeEvents 40) • A-18
sessionStartedTrap (pcubeSeEvents 39) • A-18
Setting the Calendar • 5-28
Setting the Clock • 5-27
Setting the IP Address and Subnet Mask of the FastEthernet Management Interface • 5-26
Setting the Time Zone • 5-28
setup • 4-1
Setup Utility • 4-1
show hosts • 5-36
Show IP Advertising • 5-25
Show IP Route • 5-23
Showing Calendar Time • 5-27
Showing System Time • 5-27
Shutting Down the SCE Platform • 3-12
Simultaneous Upgrade of Firmware and Application • 10-12
SLI (SML Loadable Image) File • 5
smartSUB Manager (SM) • 5
SML (Service Modeling Language) • 5
Sniffing • 5
SNMP (Simple Network Management Protocol) • 5-9
  CLI • 5-16
  community strings • 5-10
  traps • 5-12
SNMP Community Strings • 5-10
SNMP Configuration and Management • 5-9
SNMP Interface • 5-8
SNMP Protocol • 5-9
SNTP • 5-31
sntpClockDriftWarnTrap (pcubeSeEvents 19) • A-15
Specific Attack Detectors • 11-8
spIndex (subscribersPropertiesEntry 1) • A-46
Split flow • 5
splitter or switch topology • 5
spName (subscribersPropertiesEntry 2) • A-47
spType (subscribersPropertiesEntry 3) • A-47

spvIndex (subscriberPropertiesValueEntry 1) • A-48
spvPropertyCounter64Value (subscriberPropertiesValueEntry 7) • A-49
spvPropertyName (subscriberPropertiesValueEntry 3) • A-48
spvPropertyStringValue (subscriberPropertiesValueEntry 5) • A-49
spvPropertyUintValue (subscriberPropertiesValueEntry 6) • A-49
spvRowStatus (subscriberPropertiesValueEntry 4) • A-49
spvSubName (subscriberPropertiesValueEntry 2) • A-48
SSH Server • 5-5
Submitting a Service Request • xxi
Subscriber • 5
Subscriber Aging • 9-26
Subscriber anonymous groups csv file format • 9-6
Subscriber default csv file format • 9-5
Subscriber Files • 9-5
Subscriber Mapping Conflicts • 9-19
Subscriber Mapping Modes • 9-19
Subscriber modes
  Subscriber notification • 11-4
configuring • 11-11
Subscriber Modes in Service Control Solutions • 9-3
Subscriber Notification • 11-4
Subscriber Notification Ports • 11-11
Subscriber Overview • 9-1
Subscriber Rules for TIRs • 9-19
Subscriber Traffic Processor IP Ranges • 9-18
subscriberPropertiesValueEntry (subscriberPropertiesValueTable 1) • A-48
subscriberPropertiesValuesTable (subscribersGrp 3) • A-47
Subscribers
  aging • 9-26
  anonymous groups • 9-5, 9-10, 9-16
  csv files • 9-5, 9-6
  importing/exporting • 9-6
  managing via SCE CLI • 9-1
  monitoring • 9-10
  removing • 9-8
  templates • 9-5, 9-7, 9-8
Subscribers Group
  subscribersGrp (pcubeSEObjs 8) • A-41
subscribersCountersClearTime (subscribersInfoEntry 13) • A-45
subscribersInfoEntry (subscribersInfoTable 1) • A-42
subscribersInfoTable (subscribersGrp 2) • A-42
subscribersNumActive (subscribersInfoEntry 9) • A-44
subscribersNumActivePeak (subscribersInfoEntry 10) • A-44
subscribersNumActivePeakTime (subscribersInfoEntry 11) • A-44
subscribersNumAnonymous (subscribersInfoEntry 16) • A-45
subscribersNumFree (subscribersInfoEntry 2) • A-43
subscribersNumIntroduced (subscribersInfoEntry 1) • A-43
subscribersNumIpAddrMappings (subscribersInfoEntry 3) • A-43
subscribersNumIpAddrMappingsFree (subscribersInfoEntry 4) • A-43
subscribersNumIpRangeMappings (subscribersInfoEntry 5) • A-43
subscribersNumIpRangeMappingsFree (subscribersInfoEntry 6) • A-44
subscribersNumTpIpRangeMappings (subscribersInfoEntry 14) • A-45
subscribersNumTpIpRangeMappingsFree (subscribersInfoEntry 15) • A-45
subscribersNumUpdates (subscribersInfoEntry 12) • A-45
subscribersNumVlanMappings (subscribersInfoEntry 7) • A-44
subscribersNumVlanMappingsFree (subscribersInfoEntry 8) • A-44
subscribersNumWithSessions (subscribersInfoEntry 17) • A-46
subscribersPropertiesEntry (subscribersPropertiesTable 1) • A-46
subscribersPropertiesTable (subscribersGrp 2) • A-46
support • xx
Supported Standards • A-78
Syntax and Conventions • 2-12
sysFailureRecovery (systemGrp 2) • A-19
sysOperationalStatus (systemGrp 1) • A-19
System Group

systemGrp (pcubeSEObjs 1) • A-19
System Upgrades • 10-11
systemResetTrap (pcubeSeEvents 4) • A-14
sysVersion (systemGrp 3) • A-19

T

Tab Completion • 2-16
Telnet Interface • 5-4
Telnet Timeout • 5-5
Terminology and Definitions • 10-1
The [no] Prefix • 2-14
The Logging System • 4-7
The RDR Formatter • 8-1
The User Log • 4-7
Time Clocks and Time Zone • 5-26
Time settings • 5-26, 5-31
TIR (Traffic Processor IP Range) • 5
Topology
redundant • 10-2
related parameters • 10-9
Topology-Related Parameters for Redundant Topologies • 10-9
TOS marking • 6-11
tpClearCountersTime (tpInfoEntry 28) • A-57
tpCpuUtilization (tpInfoEntry 35) • A-58
tpCpuUtilizationPeak (tpInfoEntry 36) • A-59
tpCpuUtilizationPeakTime (tpInfoEntry 37) • A-59
tpFlowsCapacityUtilization (tpInfoEntry 38) • A-59
tpFlowsCapacityUtilizationPeak (tpInfoEntry 39) • A-59
tpFlowsCapacityUtilizationPeakTime (tpInfoEntry 40) • A-59
tpHandledFlowsRate (tpInfoEntry 32) • A-58
tpHandledFlowsRatePeak (tpInfoEntry 33) • A-58
tpHandledFlowsRatePeakTime (tpInfoEntry 34) • A-58
tpHandledPacketsRate (tpInfoEntry 29) • A-57
tpHandledPacketsRatePeak (tpInfoEntry 30) • A-58
tpHandledPacketsRatePeakTime (tpInfoEntry 31) • A-58
tpIndex (tpInfoEntry 2) • A-52
tpInfoEntry (tpInfoTable) • A-51
tpInfoTable (trafficProcessorGrp 1) • A-50
tpModuleIndex (tpInfoEntry 1) • A-52
tpNumActiveFlows (tpInfoEntry 5) • A-53
tpNumActiveFlowsPeak (tpInfoEntry 6) • A-53
tpNumActiveFlowsPeakTime (tpInfoEntry 7) • A-53
tpNumNonTcpUdpActiveFlows (tpInfoEntry 14) • A-55
tpNumNonTcpUdpActiveFlowsPeak (tpInfoEntry 15) • A-55
tpNumNonTcpUdpActiveFlowsPeakTime (tpInfoEntry 16) • A-55
tpNumTcpActiveFlows (tpInfoEntry 8) • A-54
TpNumTcpActiveFlowsPeak (tpInfoEntry 9) • A-54
tpNumTcpActiveFlowsPeakTime (tpInfoEntry 10) • A-54
tpNumUdpActiveFlows (tpInfoEntry 11) • A-54
tpNumUdpActiveFlowsPeak (tpInfoEntry 12) • A-54
tpNumUdpActiveFlowsPeakTime (tpInfoEntry 13) • A-54
tpServiceLoss (tpInfoEntry 41) • A-60
tpTotalNumBlockedFlows (tpInfoEntry 18) • A-55
tpTotalNumBlockedPackets (tpInfoEntry 17) • A-55
tpTotalNumDiscardedPacketsDueToBwLimit (tpInfoEntry 19) • A-56
tpTotalNumFragments (tpInfoEntry 21) • A-56
tpTotalNumHandledFlows (tpInfoEntry 4) • A-53
tpTotalNumHandledPackets (tpInfoEntry 3) • A-53
tpTotalNumIpBroadcastPackets (tpInfoEntry 25) • A-57
tpTotalNumIpCrcErrPackets (tpInfoEntry 23) • A-56
tpTotalNumIpLengthErrPackets (tpInfoEntry 24) • A-57
tpTotalNumNonIpPackets (tpInfoEntry 22) • A-56
tpTotalNumTcpUdpCrcErrPackets (tpInfoEntry 27) • A-57
tpTotalNumTtlErrPackets (tpInfoEntry 26) • A-57

tpTotalNumWredDiscardedPackets (tpInfoEntry 20) • A-56
Traffic counters • 6-5
Traffic Counters Group
trafficCountersGrp (pcubeSEObjs 14) • A-75
Traffic Processor Group
trafficProcessorGrp (pcubeSEObjs 9) • A-49
Traffic Rules • 6-5
trafficCounterIndex (trafficCountersEntry 1) • A-75
trafficCounterName (trafficCountersEntry 3) • A-76
trafficCountersEntry (trafficCountersTable 1) • A-75
trafficCountersTable (trafficCountersGrp 1) • A-75
trafficCounterType (trafficCountersEntry 4) • A-76
trafficCounterValue (trafficCountersEntry 2) • A-76
Transmit Queues Group
txQueuesGrp (pcubeSEObjs 11) • A-64
Traps • 5-12, A-14
Tunneling protocols • 6
Tunneling, configuring • 6-1
txQueuesBandwidth (txQueuesEntry 5) • A-66
txQueuesClearCountersTime (txQueuesEntry 9) • A-67
txQueuesDescription (txQueuesEntry 4) • A-66
txQueuesDroppedBytes (txQueuesEntry 10) • A-67
txQueuesEntry (txQueuesTable 1) • A-65
txQueuesModuleIndex (txQueuesEntry 1) • A-65
txQueuesPortIndex (txQueuesEntry 2) • A-65
txQueuesQueueIndex (txQueuesEntry 3) • A-66
txQueuesTable (txQueuesGrp 1) • A-64
txQueuesUtilization (txQueuesEntry 6) • A-66
txQueuesUtilizationPeak (txQueuesEntry 7) • A-66
txQueuesUtilizationPeakTime (txQueuesEntry 8) • A-67

U

Unzipping a File • 4-7
Upgrading SCE Platform Firmware • 3-7
Using this Reference • A-2
Utilities • 4-1

V

Viewing Configuration • 3-1
Viewing the User Log • 4-10
Viewing the User Log Counters • 4-9

W

Warm start • 6
Working with Directories • 4-3
Working with Files • 4-5