



SRC PE Software

Network Guide

Release

4.6.x



Published: 2013-12-17

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Copyright © 2013, Juniper Networks, Inc. All rights reserved.

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

SRC PE Software Network Guide

Release 4.6.x

Copyright © 2013, Juniper Networks, Inc.

All rights reserved.

Revision History

December 2013—Revision 1

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

SOFTWARE LICENSE

The terms and conditions for using this software are described in the software license contained in the acknowledgment to your purchase order or, to the extent applicable, to any reseller agreement or end-user purchase agreement executed between you and Juniper Networks. By using this software, you indicate that you understand and agree to be bound by those terms and conditions.

Generally speaking, the software license restricts the manner in which you are permitted to use the software and may contain prohibitions against certain uses. The software license may state conditions under which the license is automatically terminated. You should consult the license for further details.

For complete product documentation, please see the Juniper Networks Web site at www.juniper.net/techpubs.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <http://www.juniper.net/support/eula.html>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Abbreviated Table of Contents

	About the Documentation	xxxiii
Part 1	Operating the SAE	
Chapter 1	Overview of the SAE	3
Chapter 2	Configuring the SAE (SRC CLI)	19
Chapter 3	Managing Subscriber and Service Session Data (SRC CLI)	29
Chapter 4	Managing SAE Data (SRC CLI)	37
Chapter 5	Managing SAE Data (C-Web Interface)	45
Part 2	Using Juniper Networks Routers in the SRC Network	
Chapter 6	Using JunosE Routers in the SRC Network (SRC CLI)	53
Chapter 7	Using Devices Running Junos OS in the SRC Network (SRC CLI)	77
Chapter 8	Managing Junos DMI Devices Using the SRC Software	105
Part 3	Using Network Devices in the SRC Network	
Chapter 9	Integrating Third-Party Network Devices into the SRC Network (SRC CLI)	113
Part 4	Locating Subscriber Management Information	
Chapter 10	Locating Subscriber Information with the NIC	133
Chapter 11	Configuring the NIC (SRC CLI)	149
Chapter 12	Obtaining Interface Configuration for OnePopStaticRouteIp or OnePopVrflp	171
Chapter 13	Configuring Applications to Communicate with an SAE	185
Chapter 14	Configuring SRC Applications to Communicate with an SAE (SRC CLI)	187
Chapter 15	Developing Applications That Use NIC	195
Chapter 16	NIC Resolution Process	203
Chapter 17	NIC Configuration Scenarios	209
Part 5	Providing Admission Control with SRC ACP	
Chapter 18	Overview of Providing Admission Control with SRC ACP	249
Chapter 19	Configuring Admission Control (SRC CLI)	259
Chapter 20	Configuring Congestion Point Classification (SRC CLI)	297

Chapter 21	Managing SRC ACP (SRC CLI)	307
Chapter 22	Monitoring Admission Control (SRC CLI)	311
Chapter 23	Monitoring Admission Control (C-Web Interface)	329
Part 6	Using External Subscriber Monitor	
Chapter 24	Configuring External Subscriber Monitor with the SRC CLI	347
Chapter 25	Monitoring External Subscriber Events with the SRC CLI	359
Chapter 26	Monitoring External Subscriber Events with the C-Web Interface	365
Part 7	Using Session State Registrar	
Chapter 27	Session State Registrar Overview	369
Chapter 28	Planning Your Session State Registrar Cluster	391
Chapter 29	Configuring the Session State Registrar (SRC CLI)	399
Chapter 30	Managing the SSR Cluster	419
Chapter 31	Monitoring the SSR Cluster	423
Part 8	Using the Subscriber Information Collector	
Chapter 32	Overview of the Subscriber Information Collector	433
Chapter 33	Configuring the Subscriber Information Collector with the SRC CLI	471
Chapter 34	Device and Service Templates for Dynamic Authorization (SRC CLI) ...	539
Chapter 35	Monitoring the Subscriber Information Collector with the SRC CLI	585
Part 9	Controlling Volume Usage with the SRC VTA	
Chapter 36	Overview of Controlling Volume Usage with the SRC VTA	593
Chapter 37	Prerequisites for Running the SRC VTA	619
Chapter 38	Configuring the SRC VTA (SRC CLI)	639
Chapter 39	Managing the SRC VTA (SRC CLI)	701
Chapter 40	Monitoring and Testing the SRC VTA	705
Part 10	Index	
	Index	717

Table of Contents

	About the Documentation	xxxiii
	SRC Documentation and Release Notes	xxxiii
	Audience	xxxiii
	Documentation Conventions	xxxiii
	Documentation Feedback	xxxv
	Requesting Technical Support	xxxv
	Self-Help Online Tools and Resources	xxxvi
	Opening a Case with JTAC	xxxvi
Part 1	Operating the SAE	
Chapter 1	Overview of the SAE	3
	Role of the SAE	3
	Connections to Managed Devices	3
	COPS Connection Between JunosE Routers and the SAE	4
	Beep Connection Between Devices Running Junos OS and the SAE	4
	COPS Connection Between CMTS Devices and the SAE	4
	COPS Connection Between Juniper Policy Servers and the SAE	5
	SAE Support for Dual-Stack Configuration	5
	Handling of Interface-Up Notification	6
	Handling of Interface-Up Notification with Delay Timer	7
	Handling of Interface-Down Notification	8
	Service Activation	8
	Service Deactivation	8
	Subscriber Attributes	8
	SAEAccess API Plug-in Attributes	9
	Subscriber Session Lookup	9
	SAE Plug-Ins	9
	Internal Plug-Ins	10
	External Plug-Ins	10
	Hosted Plug-Ins	11
	Tracking and Controlling Subscriber and Service Sessions with SAE APIs	11
	SAE Core API	12
	SAE CORBA Remote API	12
	SAE Accounting	13
	Accounting Policy	14
	Subscription Process	16
	Tracking Subscriber Sessions	17
	Accounting Plug-Ins	17
	Interim Accounting	17

Chapter 2	Configuring the SAE (SRC CLI)	19
	SRC Access to Directory Data	19
	Configuring LDAP Access to Directory Data (SRC CLI)	20
	Configuring Access Through LDAPS to Service and Subscriber Data (SRC CLI)	20
	Configuring Access to Subscriber Data (SRC CLI)	21
	Configuring Access to Service Data (SRC CLI)	23
	Configuring Access to Policy Data (SRC CLI)	24
	Configuring Access to the Persistent Login Cache (SRC CLI)	25
	Configuring the Location of Network Device Data (SRC CLI)	26
	Enabling Automatic Discovery of Changes in SAE Configuration Data (SRC CLI)	27
	Setting the Timeout and Number of Events for SAE Directory Eventing (SRC CLI)	28
Chapter 3	Managing Subscriber and Service Session Data (SRC CLI)	29
	Storing Subscriber and Service Session Data	29
	Session Store Files	29
	Active and Passive Session Stores	29
	Standby SAEs	30
	Session Store File Rotation	30
	Configuring the Session Store Feature (SRC CLI)	31
	Configuring Session Store Parameters for a Device Driver	31
	Configuring Global Session Store Parameters	33
	Reducing the Size of Objects for the Session Store Feature	35
	Configuring the Number of Threads for Sessions (SRC CLI)	35
Chapter 4	Managing SAE Data (SRC CLI)	37
	Commands to Manage SAE Data	37
	Reloading the SAE Data (SRC CLI)	38
	Reloading the SAE Configuration (SRC CLI)	38
	Reloading Services (SRC CLI)	39
	Reloading Subscriptions (SRC CLI)	39
	Reloading Interface Classification Scripts (SRC CLI)	39
	Reloading Domain Maps (SRC CLI)	40
	Removing the Directory Blacklist (SRC CLI)	40
	Removing Login Registrations (SRC CLI)	40
	Removing Equipment Registrations (SRC CLI)	41
	Modifying Failover Server Parameters (SRC CLI)	41
	Shutting Down the Device Drivers (SRC CLI)	42
Chapter 5	Managing SAE Data (C-Web Interface)	45
	Reloading the SAE Data (C-Web Interface)	45
	Reloading the SAE Configuration (C-Web Interface)	46
	Reloading Services (C-Web Interface)	46
	Reloading Subscriptions (C-Web Interface)	46
	Reloading Interface Classification Scripts (C-Web Interface)	47
	Reloading Domain Maps (C-Web Interface)	47
	Removing the Directory Blacklist (C-Web Interface)	47
	Removing Login Registrations (C-Web Interface)	48

	Removing Equipment Registrations (C-Web Interface)	48
	Modifying Failover Server Parameters (C-Web Interface)	49
	Shutting Down the Device Drivers (C-Web Interface)	49
Part 2	Using Juniper Networks Routers in the SRC Network	
Chapter 6	Using JunosE Routers in the SRC Network (SRC CLI)	53
	COPS Connection Between JunosE Routers and the SAE	53
	Highly Available Connections to JunosE Routers	54
	Adding JunosE Routers and Virtual Routers (SRC CLI)	54
	Adding Operative JunosE Routers and Virtual Routers	55
	Adding Routers Individually (SRC CLI)	55
	Adding Virtual Routers Individually (SRC CLI)	57
	Configuring the SAE to Manage JunosE Routers (SRC CLI)	59
	How SNMP Obtains Information from Routers for the SRC Software	62
	Developing Router Initialization Scripts for Network Devices and Juniper Networks	
	Routers	63
	Interface Object Fields	64
	Required Methods	65
	Example: Router Initialization Script	65
	Specifying JunosE Router Initialization Scripts on the SAE (SRC CLI)	66
	Updating Local IP Address Pools for JunosE Virtual Routers (SRC CLI)	67
	Updating Quality of Service Profiles for JunosE Virtual Routers (SRC CLI)	68
	Accessing the Router CLI	69
	Starting the SRC Client on a JunosE Router	70
	Stopping the SRC Client on a JunosE Router	70
	Monitoring Interactions Between the SAE and the Router Running JunosE	
	Software	71
	Troubleshooting Problems with Managing JunosE Routers	71
	Viewing the State of JunosE Device Drivers (SRC CLI)	72
	Viewing Statistics for Specific JunosE Device Drivers (SRC CLI)	74
	Viewing Statistics for All JunosE Device Drivers (SRC CLI)	74
	Viewing the State of JunosE Device Drivers (C-Web Interface)	75
	Viewing Statistics for All JunosE Device Drivers (C-Web Interface)	75
Chapter 7	Using Devices Running Junos OS in the SRC Network (SRC CLI)	77
	BEEP Connection Between Devices Running Junos OS and the SAE	78
	Managing DMI Devices on Routers Running Junos OS Using the SRC Software	
	and Junos Space	78
	Adding Devices Running Junos OS and Virtual Routers (SRC CLI)	79
	Adding Operative Devices Running Junos OS (SRC CLI)	79
	Adding Routers Individually (SRC CLI)	79
	Adding Virtual Routers Individually (SRC CLI)	81
	Updating Local IP Address Pools for Junos Virtual Routers (SRC CLI)	83
	Retrieving Local IP Address Pool Information from Junos Router During Router	
	Initialization and Pool Republishing (SRC CLI)	84
	Configuring the SAE to Manage Devices Running Junos OS (SRC CLI)	86
	Configuring Secure Connections Between the SAE and Devices Running Junos	
	OS	88
	Adding the Server Certificate on the Device	89

Creating a Client Certificate for the Router	90
Adding the Client Certificate on the Router	90
Configuring the SAE to Use TLS (SRC CLI)	91
Configuring TLS on the SAE (SRC CLI)	91
SAE Verification of Junos OS Configuration Changes	92
Setting Up Periodic Configuration Checking (SRC CLI)	92
Using SNMP to Retrieve Information from JunosE Routers and Devices Running Junos OSs (SRC CLI)	93
Specifying Router Initialization Scripts on the SAE (SRC CLI)	94
Configuring Devices Running Junos OS to Interact with the SAE	95
SAE Tracking for LSPs Configured on Devices Running Junos OS	96
SAE Tracking for LSPs Configured on Devices Running Junos OS Overview	96
Configuring Event Tracking for Junos LSPs (SRC CLI)	96
Configuring the Device Running Junos OS to Apply Changes It Receives from the SAE	97
Disabling Interactions Between the SAE and Devices Running Junos OS	97
Monitoring Interactions Between the SAE and Devices Running Junos OS	98
Troubleshooting Problems Between the SRC module and Device Drivers Running Junos OS	98
Troubleshooting Problems with the SRC Software Process	98
Viewing the State of Device Drivers Running Junos OS (SRC CLI)	99
Viewing Statistics for Specific Device Drivers Running Junos OS (SRC CLI)	100
Viewing Statistics for All Device Drivers Running Junos OS (SRC CLI)	101
Viewing the State of Device Drivers Running Junos OS (C-Web Interface)	101
Viewing Statistics for Specific Device Drivers Running Junos OS (C-Web Interface)	102
Viewing Statistics for All Device Drivers Running Junos OS (C-Web Interface)	102
Chapter 8 Managing Junos DMI Devices Using the SRC Software	105
Managing DMI Devices on Routers Running Junos OS Using the SRC Software and Junos Space	105
Managing DMI Devices Using the SRC Software and Junos Space Overview	106
Configuration Overview	106
Redundancy	107
Summary of Tasks for Configuring the SRC Software to Manage DMI-Enabled Routers Running Junos OS (SRC CLI)	107
Adding the Router Running Junos OS as a DMI Network Device (SRC CLI)	108
Configuring the Junos DMI Driver (SRC CLI)	109
Migrating from the Junos (BEEP) Driver to the Junos DMI Driver (SRC CLI)	110

Part 3	Using Network Devices in the SRC Network	
Chapter 9	Integrating Third-Party Network Devices into the SRC Network (SRC CLI)	113
	Integrating Network Devices into the SRC Network Overview	113
	SAE Communities	114
	Storing Session Data	114
	Using Script Services to Provision Third-Party Devices	114
	Logging In Subscribers and Creating Sessions	115
	Assigned IP Subscribers	115
	Login Interactions with Assigned IP Subscribers	116
	Event Notification from an IP Address Manager	117
	Login with Event Notification	117
	Configuration Tasks for Integrating Third-Party Network Devices (SRC CLI)	118
	Setting Up Script Services	119
	Adding Objects for Network Devices (SRC CLI)	119
	Adding Virtual Router Objects (SRC CLI)	120
	Setting Up SAE Communities (SRC CLI)	122
	Configuring the SAE Community Manager	122
	Specifying the Community Manager in the SAE Device Driver	123
	Configuring SAE Properties for the Event Notification API (SRC CLI)	124
	Developing Router Initialization Scripts for Network Devices and Juniper Networks	
	Routers	125
	Interface Object Fields	126
	Required Methods	127
	Example: Router Initialization Script	127
	Copying Initialization Scripts to the C Series Controller	128
	Specifying Initialization Scripts on the SAE (SRC CLI)	128
	Using SNMP to Retrieve Information from Network Devices	129
	Configuring Global SNMP Communities in the SRC Software (SRC CLI)	129
	Using the NIC Resolver in Environments That Have Third-Party Devices (SRC CLI)	130
Part 4	Locating Subscriber Management Information	
Chapter 10	Locating Subscriber Information with the NIC	133
	Locating Subscriber Management Information	133
	NIC Client/Server Mode	134
	NIC Local Host Mode	134
	Mapping Subscribers to a Managing SAE	135
	NIC Proxies and NIC Locators	135
	NIC Hosts	135
	NIC Agents	136
	NIC Resolvers	136

	High Availability for NIC	137
	High Availability in Existing NIC Configurations	137
	NIC Replication	137
	Planning a NIC Implementation	139
	NIC Configuration Scenarios	140
	NIC Agents Used in the NIC Configuration Scenarios	144
	Router Initialization Scripts with NIC Configuration Scenarios	146
Chapter 11	Configuring the NIC (SRC CLI)	149
	Configuration Statements for the NIC	149
	Configuration Statements for NIC Operating Properties	150
	Configuration Statements for NIC Scenarios	150
	Configuration Statements for NIC Logging	151
	Before You Configure the NIC	151
	Configuring the NIC (SRC CLI)	152
	Reviewing and Changing Operating Properties for the NIC (SRC CLI)	153
	Reviewing the Default NIC Operating Properties	153
	Changing NIC Operating Properties	154
	Configuring NIC Replication (SRC CLI)	155
	Configuring a NIC Scenario (SRC CLI)	156
	Defining the NIC Configuration to Use	156
	Configuring Directory Agents	159
	Configuring SAE Client Agents	161
	Configuring SAE Plug-In Agents	162
	Configuring the SAE to Communicate with SAE Plug-In Agents When You Use NIC Replication	163
	Configuring Advanced NIC Features	165
	Verifying Configuration for the NIC (SRC CLI)	165
	Starting the NIC (SRC CLI)	166
	Testing a NIC Resolution (SRC CLI)	166
	Stopping a NIC Host on a C Series Controller (SRC CLI)	167
	Restarting the NIC (SRC CLI)	167
	Restarting a NIC Agent (SRC CLI)	168
	Restarting a NIC Resolver (SRC CLI)	168
	Changing NIC Configurations (SRC CLI)	169
Chapter 12	Obtaining Interface Configuration for OnePopStaticRouteIp or OnePopVrflp	171
	Network Publisher Overview	171
	NIC Document That Maps Subscriber IP Addresses to a Junos OS Interface	172
	Configuration Statements for the Network Publisher	172
	Before You Configure and Run the Network Publisher	173
	Configuring the Network Publisher (SRC CLI)	174
	Configuring Local Configuration for the Network Publisher	174
	Configuring Connections Between Devices Running Junos OS and the Network Publisher	175
	Configuring Router Authentication for the Network Publisher	176
	Configuring Routing Table Filters for the Network Publisher	177

	Configuring the Connection Between the Network Publisher and the Juniper Networks Database	178
	Running the Network Publisher (SRC CLI)	179
	Files Used to Test Network Publisher	180
	Configuring Information to Test the Network Publisher (SRC CLI)	180
	Troubleshooting Network Publisher Operations (SRC CLI)	181
	Reviewing the Information Collected from a Device Running Junos OS (SRC CLI)	182
Chapter 13	Configuring Applications to Communicate with an SAE	185
	NIC Proxy Configuration Overview	185
	Before You Configure a NIC Proxy	186
Chapter 14	Configuring SRC Applications to Communicate with an SAE (SRC CLI)	187
	Configuration Statements for NIC Proxies	187
	Configuring Resolution Information for a NIC Proxy (SRC CLI)	188
	Changing the Configuration for the NIC Proxy Cache (SRC CLI)	190
	Configuring a NIC Proxy for NIC Replication (SRC CLI)	191
	Configuring NIC Test Data (SRC CLI)	193
Chapter 15	Developing Applications That Use NIC	195
	External Application Requirements for NIC	195
	External Non-Java Applications That Use NIC	195
	Creating a NIC Locator to Include with a Non-Java Application	196
	External Java Applications That Use NIC	197
	Developing a Java Application to Communicate with a NIC Proxy	198
	Instantiating a Configuration Manager	198
	Passing a Reference to the Configuration Manager to the NIC Factory	198
	Instantiating the NIC Factory Class	198
	Initializing Logging	199
	Instantiating the NIC Proxy	199
	Managing a Resolution Request	200
	Deleting Invalid Results from the NIC Proxy's Cache	201
	Removing the NIC Proxies	202
	Updating Information About Address Pools	202
Chapter 16	NIC Resolution Process	203
	NIC Resolution Process Overview	203
	NIC Realms	203
	Key to Value Resolution	204
	NIC Data Types	204
	Constraints as NIC Data Types	206
Chapter 17	NIC Configuration Scenarios	209
	NIC Configuration Scenarios Overview	209
	OnePop Scenario	210
	Centralized Configuration	210
	Distributed Configuration	211

Redundancy	211
OnePopPcmm Scenario	212
Centralized Configuration	213
Distributed Configuration	214
OnePopDynamicIp Scenario	214
Centralized Configuration	215
Distributed Configuration	216
OnePopSharedIp Scenario	216
Centralized Configuration	217
Distributed Configuration	218
OnePopStaticRouteIp Scenario	218
Centralized Configuration	219
Distributed Configuration	220
OnePopVrflp Scenario	221
Centralized Configuration	221
Distributed Configuration	222
OnePopAcctId Scenario	223
OnePopLogin Scenario	225
Centralized Configuration	226
Distributed Configuration	227
OnePopLoginPull Scenario	227
OnePopPrimaryUser	228
Centralized Configuration	229
Distributed Configuration	229
OnePopDnSharedIp Scenario	230
Centralized Configuration	231
Distributed Configuration	231
OnePopAllRealms Scenario	234
OnePopTunnel Scenario	238
Centralized Configuration	238
OnePopPrefixIp Scenario	239
MultiPop Scenario	240
IP Realm	241
Shared IP Realm	242
DN Realm	244

Part 5

Chapter 18

Providing Admission Control with SRC ACP

Overview of Providing Admission Control with SRC ACP	249
SRC ACP Overview	249
Deriving Congestion Points Automatically	251
Deriving Edge Congestion Points	251
Deriving Congestion Points from a Profile	252
Deriving Backbone Congestion Points	253
Allocating Bandwidth to Applications Not Controlled by SRC ACP	254
Use of Multiple SRC ACPs	254
Interactions Between SRC ACP and Other Components	254
Redundancy and State Synchronization	256

Chapter 19

Fault Recovery	257
Creating an Application to Update Information for SRC ACP	258
Configuring Admission Control (SRC CLI)	259
Configuration Statements for SRC ACP	259
Configuring SRC ACP (SRC CLI)	262
Creating Grouped Configurations for SRC ACP (SRC CLI)	262
Configuring Local Properties for SRC ACP (SRC CLI)	263
Configuring Basic Local Properties for SRC ACP	264
Configuring Initial Properties for SRC ACP	265
Configuring Directory Connection Properties for SRC ACP	265
Configuring Initial Directory Eventing Properties for SRC ACP	266
Configuring the SAE for SRC ACP (SRC CLI)	267
Configuring SRC ACP as an External Plug-In	267
Configuring Event Publishers	267
Configuring the SAE to Monitor Interfaces for Congestion Points	267
Configuring SRC ACP Properties (SRC CLI)	269
Configuring Logging Destinations for SRC ACP	269
Configuring SRC ACP Operation	271
Configuring CORBA Interfaces	274
Configuring SRC ACP Redundancy	275
Configuring Connections to the Subscribers' Directory	277
Configuring Connections to the Services' Directory	278
Configuring SRC ACP Scripts and Classification	280
Configuring SRC ACP to Manage the Edge Network (SRC CLI)	281
Configuring Network Interfaces in the Directory for the Edge Network	281
Configuring Bandwidths for Subscribers	282
Assigning Network Interfaces to Subscribers	283
Configuring Bandwidths for Services in the Edge Network	284
Configuring SRC ACP to Manage the Backbone Network (SRC CLI)	285
Configuring Network Interfaces in the Directory for the Backbone Network	285
Extending SRC ACP Congestion Points for the Backbone Network	285
Configuring Action Congestion Points	286
Configuring Bandwidths for Services in the Backbone Network	287
Configuring Congestion Points for Services in the Backbone Network	287
Plug-In Attributes for Use with Backbone Congestion Point Expressions	289
Using Functions for Backbone Congestion Point Classification Scripts	292
Configuring Congestion Point Profiles in the Directory	293
Assigning Interfaces to Congestion Point Profiles	293
Defining SRC ACP Congestion Point Usage Trap Thresholds (SRC CLI)	294

Chapter 20	Configuring Congestion Point Classification (SRC CLI)	297
	Congestion Point Classification Overview	297
	Congestion Point Classification Scripts	297
	Congestion Point Profiles	298
	Configuration Statements for Congestion Point Classification	298
	Classifying Congestion Points (SRC CLI)	298
	Configuring Targets and Criteria for Classification Scripts	298
	Configuring Classification Scripts Contents for Classification Scripts	299
	Configuring Congestion Point Classification Targets	299
	Congestion Point Classification Criteria	300
	Defining a Congestion Point Profile (SRC CLI)	304
	Congestion Point Expressions	304
	Expressions in Templates for Congestion Point Profiles	305
	Methods for Use with Scripting Expressions	305
	Match Criteria for Congestion Point Classification	305
Chapter 21	Managing SRC ACP (SRC CLI)	307
	Starting SRC ACP	307
	Stopping SRC ACP	307
	Reorganizing the File That Contains ACP Data	308
	Modifying Congestion Points	308
Chapter 22	Monitoring Admission Control (SRC CLI)	311
	Viewing Information About Subscriber Sessions in the Edge Network (SRC CLI)	311
	Viewing Edge Congestion Point Information by DN (SRC CLI)	312
	Viewing Edge Congestion Point Information by Subscriber Session (SRC CLI)	313
	Viewing Information About Services in the Backbone Network (SRC CLI)	313
	Viewing Backbone Congestion Point Information by DN (SRC CLI)	314
	Viewing Backbone Congestion Point Information by Service (SRC CLI)	315
	Viewing Congestion Point Information by Subscriber IP Address and Associated Service Sessions (SRC CLI)	315
	Viewing Congestion Point Information by Session ID and Associated Service Sessions (SRC CLI)	318
	Viewing Congestion Point Information by Login Name and Associated Service Sessions (SRC CLI)	321
	Viewing Action Congestion Point Information by Service (SRC CLI)	324
	Viewing Action Congestion Point Information by Congestion Point (SRC CLI)	324
	Viewing Information About Subscribers Obtained from External Applications (SRC CLI)	325
	Viewing Congestion Point Information by DN (SRC CLI)	326
	Viewing Congestion Point Information by Name (SRC CLI)	327
	Viewing SNMP Information for Devices (SRC CLI)	327
	Viewing SNMP Information for the Directory (SRC CLI)	328
	Viewing SNMP Information for SRC ACP (SRC CLI)	328

Chapter 23	Monitoring Admission Control (C-Web Interface)	329
	Viewing Information About Subscriber Sessions in the Edge Network (C-Web Interface)	329
	Viewing Information About Edge Congestion Points by DN (C-Web Interface)	330
	Viewing Information About Edge Congestion Points by Subscriber Session (C-Web Interface)	331
	Viewing Information About Services in a Backbone Network (C-Web Interface)	332
	Viewing Information About Congestion Points in a Backbone Network by Expression (C-Web Interface)	334
	Viewing Information About Congestion Points in a Backbone Network by DN (C-Web Interface)	335
	Viewing Information about Action Congestion Points in a Backbone Network by Service (C-Web Interface)	336
	Viewing Information about Action Congestion Points in a Backbone Network by Expression (C-Web Interface)	338
	Viewing Information About Subscribers Obtained from External Applications (C-Web Interface)	339
	Viewing Information About Congestion Points from an External Application by DN (C-Web Interface)	341
	Viewing Information About Congestion Points from an External Application by Interface Name (C-Web Interface)	341
	Viewing Statistics for the SRC ACP Configuration (C-Web Interface)	342
	Viewing General Statistics for SRC ACP (C-Web Interface)	342
	Viewing Statistics for the SRC ACP Directory (C-Web Interface)	343
	Viewing Device Statistics for SRC ACP (C-Web Interface)	344
Part 6	Using External Subscriber Monitor	
Chapter 24	Configuring External Subscriber Monitor with the SRC CLI	347
	External Subscriber Monitor Overview	347
	Configuring External Subscriber Monitor (SRC CLI)	348
	Configuring Basic Local Properties for External Subscriber Monitor	348
	Configuring Initial Properties for External Subscriber Monitor	349
	Configuring Directory Connection Properties for External Subscriber Monitor	349
	Configuring Eventing Properties for External Subscriber Monitor	350
	Configuring Logging Destinations for External Subscriber Monitor	350
	Configuring the NIC Proxy for the Pseudo-RADIUS Server (SRC CLI)	352
	Configuring Resolution Information for a NIC Proxy	352
	Changing the Configuration for the NIC Proxy Cache	352
	Configuring a NIC Proxy for NIC Replication	353
	Configuring the Pseudo-RADIUS Server for External Subscriber Monitor (SRC CLI)	354
	Configuring the Client Secret for External Subscriber Monitor (SRC CLI)	355
	Configuring Event Notification for External Subscriber Monitor (SRC CLI)	356
	Starting External Subscriber Monitor (SRC CLI)	357
	Stopping External Subscriber Monitor (SRC CLI)	357

Chapter 25	Monitoring External Subscriber Events with the SRC CLI	359
	Viewing Statistics for External Subscriber Monitor (SRC CLI)	359
	Monitoring Statistics for External Subscriber Monitor (SRC CLI)	360
	Viewing Statistics for External Subscriber Monitor Event Notifications (SRC CLI)	360
	Monitoring Statistics for External Subscriber Monitor Event Notifications (SRC CLI)	362
	Viewing Statistics for the Agent Process (SRC CLI)	362
Chapter 26	Monitoring External Subscriber Events with the C-Web Interface	365
	Viewing Statistics for External Subscriber Monitor (C-Web Interface)	365
	Viewing Statistics for External Subscriber Monitor Event Notifications (C-Web Interface)	366
	Viewing Statistics for the Agent Process (C-Web Interface)	366
Part 7	Using Session State Registrar	
Chapter 27	Session State Registrar Overview	369
	Session State Registrar Overview	369
	SSR Node Types	370
	SSR Node Groups	371
	C Series Controller Requirements	372
	SSR Cluster Configurations Overview	373
	Small-Scale Deployments	374
	Scaling the SSR Cluster	374
	Scaling the Front End of the Cluster	374
	Scaling the Back End of the Cluster	375
	SSR Cluster Network Requirements	375
	Supported SSR Cluster Configurations	377
	Failover Overview	378
	Failover Examples	379
	Possible Failure Scenarios	380
	Distributed Cluster Failure and Recovery	381
	SSR Database Schema	384
	Subscriber Sessions Table	384
	Attribute Associations	386
	Service Sessions Table	387
	Making Modifications to the SSR Database Schema Overview	388
	SSR Database Operating Modes	389
	Distributing the SSR Cluster Configuration and Enabling SSR Client Components	389
	Enabling, Restarting and Disabling the SSR Component Database	389
Chapter 28	Planning Your Session State Registrar Cluster	391
	Planning the SSR Cluster Topology	391
	Small-Scale Deployments	391
	Identifying the C Series Controllers in the SSR Cluster	392
	SSR Cluster Planning Worksheets	393

Chapter 29	Configuring the Session State Registrar (SRC CLI)	399
	Configuration Changes and Their Impact on the SSR Cluster	399
	Configuration Statements for the SSR Cluster	400
	Configuring the Initial SSR Cluster (SRC CLI)	401
	Configuring the SSR Cluster ID (SRC CLI)	402
	Configuring the SSR Cluster Geometry (SRC CLI)	403
	Configuring the Nodes in the SSR Cluster (SRC CLI)	404
	Configuring the Management Servers in the SSR Cluster (SRC CLI)	406
	Configuring the Database Memory Size When Running the Two-Shared-Data-Node Geometry (SRC CLI)	406
	Configuring the Fields in the Subscriber Sessions Table (SRC CLI)	407
	Mapping SAE Plug-In Attributes to Fields in the Subscriber Sessions Table (SRC CLI)	409
	Modifying the SSR Database Schema in an Active Cluster (SRC CLI)	410
	Modifying Attribute Mapping in an Active SSR Cluster (SRC CLI)	411
	Adding Data Nodes to an Active SSR Cluster (SRC CLI)	412
	Adding Client Nodes to an Active SSR Cluster (SRC CLI)	413
	Adding a Management Server to an Active SSR Cluster	414
	Removing Data Nodes from an Active SSR Cluster	415
	Removing a Client Node from an Active SSR Cluster	417
	Removing a Management Server from an Active SSR Cluster	418
Chapter 30	Managing the SSR Cluster	419
	Placing the SSR Database into Maintenance Mode (SRC CLI)	419
	Placing the SSR Database into Production Mode (SRC CLI)	419
	Deleting the SSR Database (SRC CLI)	420
	Creating the SSR Database (SRC CLI)	420
	Enabling the SSR Database (SRC CLI)	421
	Disabling the SSR Database (SRC CLI)	421
	Restarting the SSR Database (SRC CLI)	421
	Deleting All Subscriber Sessions in the SSR Database	422
	Deleting Subscriber Sessions in the SSR Database By IP Address	422
Chapter 31	Monitoring the SSR Cluster	423
	Viewing the SSR Database Mode (SRC CLI)	423
	Viewing the Status of the SSR Cluster (SRC CLI)	423
	Viewing the Database Memory Requirements (SRC CLI)	424
	Viewing the Running Configuration of the SSR Database (SRC CLI)	425
	Viewing All Subscriber Sessions in the SSR Database (SRC CLI)	426
	Viewing Subscriber Sessions in the SSR Database by IP Address (SRC CLI)	427
	Viewing Subscriber Sessions in the SSR Database by Indexed Field (SRC CLI)	428
	Viewing the Total Number of Subscriber Sessions in the SSR Database (SRC CLI)	430
Part 8	Using the Subscriber Information Collector	
Chapter 32	Overview of the Subscriber Information Collector	433
	Subscriber Information Collector Overview	433
	Managing Dynamic Services	434

SIC Dynamic Authorization Support Overview	435
Rendering	436
How the Dynamic Authorization Process Works in the SIC	437
Introduction	437
Initial Authorization	438
Accounting	438
Service Activation and Deactivation	439
Abort Session Requests	440
Dynamic Authorization Targets (SRC CLI)	441
RADIUS Authentication/Authorization and Accounting Data Flow	441
COA Authentication Data Flow	443
COA Accounting Data Flow	444
SIC Accounting Data Flow (Accounting Target=SSR)	444
Local and Shared Configurations for the SIC (SRC CLI)	445
Local Configuration	445
Shared Configuration	445
Accounting Methods and Targets (SRC CLI)	446
Using the SSR Database as the Accounting Method	446
How the SIC Seamlessly Processes IP Address Changes When the SSR Database Is the Accounting Method	447
Mapping Attributes When Using the Database Accounting Method . .	448
Using the Proxy RADIUS Accounting Method	449
Authentication Route Targets (SRC CLI)	450
Request Routing (SRC CLI)	450
Explicit Routing Rules	450
Implicit Routing Rules	452
SIC Editing Rules (SRC CLI)	453
RADIUS and Diameter Configuration for the SIC Overview (SRC CLI)	456
RADIUS Accounting and Authentication Listeners	457
SIC Diameter Server	457
SIC Diameter Server Configuration Overview	458
RADIUS Network Elements	458
Configuring Upstream RADIUS Network Elements Overview	459
Configuring Downstream RADIUS Network Elements Overview	460
Using the Proxy Function to Define Implicit Routing Rules	460
Failover Policy	460
Failover Mode	461
Round-Robin	461
Primary or Backup	462
RADIUS and Diameter Transports	462
Inbound and Outbound RADIUS Accounting Transports for the SIC Group	462
Inbound and Outbound RADIUS Authentication Transports for the SIC Group	462
SIC Server Inbound and Outbound RADIUS Transports	462
Diameter Transport	462

	SIC Dictionaries and Device Models Overview (SRC CLI)	463
	Dictionaries and the Device Models Supported by the SIC Group	463
	Configuring Device Models and Their Associated Dictionaries for the SIC Group Overview	463
	Modifying a Dictionary	464
	Configuring the Dictionaries Used by the SIC Group Overview	464
	SIC Local Realms Overview	464
	SIC Event Logging Overview (SRC CLI)	464
	Log File Options	465
	Event Levels	466
	Log Groups	467
	SNMP Support for the SIC Overview (SRC CLI)	467
Chapter 33	Configuring the Subscriber Information Collector with the SRC CLI	471
	SIC Configuration Summary	472
	SIC RADIUS Configuration Summary (SRC CLI)	472
	SIC RADIUS Dynamic Authorization Configuration Summary (SRC CLI)	473
	SIC Diameter Configuration Summary (SRC CLI)	474
	Configuring Management of RADIUS-Enabled Devices for the SIC (SRC CLI)	474
	Configuring the Connection Between the SIC and the Juniper Networks Database (SRC CLI)	474
	Creating an SIC Group and Server (SRC CLI)	476
	Creating an SIC Server Instance (SRC CLI)	477
	Configuring Dictionaries for the SIC Group (SRC CLI)	478
	Configuring the Device Models Supported by the SIC Group (SRC CLI)	480
	Configuring the RADIUS Accounting Listener for the SIC Group (SRC CLI)	480
	Configuring the RADIUS Accounting Listener Queue Limits (SRC CLI)	480
	Configuring the RADIUS Accounting Listener Transport (SRC CLI)	481
	Configuring the RADIUS Authentication Listener for the SIC Group (SRC CLI)	482
	Configuring the RADIUS Authentication Listener Queue Limits (SRC CLI)	482
	Configuring the RADIUS Authentication Listener Transport (SRC CLI)	483
	Configuring the Outbound RADIUS Transport of the SIC Group (SRC CLI)	484
	Configuring the RADIUS Transport for an SIC Server (SRC CLI)	485
	Configuring the SIC Diameter Server (SRC CLI)	486
	Configuration Statements for the SIC Diameter Server (SRC CLI)	486
	Configuring the SIC Diameter Server Identity (SRC CLI)	487
	Configuring the SIC Diameter Server Peer (SRC CLI)	488
	Configuring Upstream and Downstream RADIUS Network Elements (SRC CLI)	490
	Configuration Statements for Downstream Network Elements and Accounting and Authentication Targets (SRC CLI)	491
	Configuration Statements for Upstream Network Elements, Accounting and Authentication Clients, and Dynamic Authorization Targets (SRC CLI)	492
	Creating a Network Element (SRC CLI)	492
	Configuring the Device Models Supported in the Network Element (SRC CLI)	493

Configuring Upstream Network Elements and Accounting and Authentication Clients (SRC CLI)	494
Configuring Upstream Network Elements and Dynamic Authorization Targets (SRC CLI)	495
Configuring Downstream Network Elements and Accounting and Authentication Targets (SRC CLI)	496
Configuring SIC Accounting Targets (SRC CLI)	496
Configuring SIC Authentication Targets (SRC CLI)	497
Configuration Statements for SIC Group Failover Mode and Policy (SRC CLI)	498
Configuring Failover Mode and Policy (SRC CLI)	499
Configuring Failover Mode (SRC CLI)	499
Configuring Fast Fail Options for the Failover Policy	500
Configuring Retry Options for the Failover Policy	501
Configuring What Realms Are Local to the SIC Group (SRC CLI)	501
Configuration Statements for SIC Editing Rules (SRC CLI)	502
Configuring the Optional Editing Rules Used by the SIC Group (SRC CLI)	505
Configuring the Accounting Method Used by the SIC Group (SRC CLI)	507
Configuring the SSR Database as the Accounting Method (SRC CLI)	508
Configuring Proxy RADIUS as the Accounting Method (SRC CLI)	509
Configuring the Authentication Target Used by the SIC Server (SRC CLI)	510
Configuring Request Routing (SRC CLI)	510
Configuration Statements for SIC Explicit Accounting Routing Rules	510
Configuration Statements for SIC Explicit Authentication Routing Rules	511
Configuring Explicit Routing (SRC CLI)	512
Configuring Implicit Routing (SRC CLI)	515
Configuring Event Logging for an SIC Server (SRC CLI)	515
Configuring SNMP for the SIC Group (SRC CLI)	519
Example: Basic SIC Group Configuration (SRC CLI)	520
Configuring the NAS Groups (SRC CLI)	531
Configuring NAS Groups	531
Configuring the NAS Group Device Capabilities (SRC CLI)	532
Classifying Interfaces	532
Configuring NAS Group Routes	533
Configuring the SAE to Manage AAA Devices	535
Configuring AAA Policies (SRC CLI)	537
Configuring AAA Policy Lists	537
Configuring AAA Policy Rules	537
Configuring Template Activation Actions	537
Chapter 34 Device and Service Templates for Dynamic Authorization (SRC CLI) . . .	539
Device and Service Template Configuration Overview (SRC CLI)	539
Device Template Configuration Overview (SRC CLI)	539
Service and Global Service Template Configuration Overview (SRC CLI) . .	540
Mode	540
Attributes	542
Variables	544

Tagged Attributes	545
Configuring Device Templates (SRC CLI)	545
Configuring the Device Capabilities Supported in the Device Template (SRC CLI)	546
Configuration Statements for SIC Service Templates (SRC CLI)	548
Configuring SIC Service Templates (SRC CLI)	549
Creating an SIC Service Template (SRC CLI)	549
Configuring the Mode of the SIC Service Template (SRC CLI)	550
Configuring Variables for the SIC Service Template (SRC CLI)	550
Configuring Normal Attributes for the SIC Service Template (SRC CLI)	551
Configuring Required Attributes for the SIC Service Template (SRC CLI)	552
Configuring Default Attributes for the SIC Service Template (SRC CLI)	553
Configuring Parameterized Attributes for the SIC Service Template (SRC CLI)	554
Configuring Override Attributes for the SIC Service Template (SRC CLI)	555
Configuration Statements for Tagged Attributes in SIC Service Templates (SRC CLI)	557
Configuring Tagged Attributes in SIC Service Templates (SRC CLI)	558
Creating a Tagged Attribute Group in the SIC Service Template (SRC CLI)	558
Configuring Normal Attributes in a Tagged Attribute Group (SRC CLI)	559
Configuring Default Attributes in a Tagged Attribute Group (SRC CLI)	560
Configuring Required Attributes in a Tagged Attribute Group (SRC CLI)	561
Configuring Override Attributes in a Tagged Attribute Group (SRC CLI)	562
Configuring Parameterized Attributes in a Tagged Attribute Group (SRC CLI)	563
Configuration Statements for SIC Global Service Templates (SRC CLI)	564
Configuring Global Service Templates (SRC CLI)	565
Creating an SIC Global Service Template (SRC CLI)	566
Configuring the Mode of the SIC Global Service Template (SRC CLI)	566
Configuring Variables for the SIC Global Service Template (SRC CLI)	567
Configuring Normal Attributes for the SIC Global Service Template (SRC CLI)	567
Configuring Required Attributes for the SIC Global Service Template (SRC CLI)	568
Configuring Default Attributes for the SIC Global Service Template (SRC CLI)	569
Configuring Parameterized Attributes for the SIC Global Service Template (SRC CLI)	571
Configuring Override Attributes for the SIC Global Service Template (SRC CLI)	572
Sample Service Templates	573
Juniper Networks Routers Service Template	573
Cisco Router Service Template	578
Chapter 35	
Monitoring the Subscriber Information Collector with the SRC CLI	585
Viewing Statistics for RADIUS Client Accounting Requests (SRC CLI)	585
Viewing Statistics for RADIUS Client Authentication Requests (SRC CLI)	585
Viewing RADIUS Host Statistics for Accounting Transactions (SRC CLI)	586

Viewing RADIUS Host Statistics for Authentication Transactions (SRC CLI) . . .	586
Viewing RADIUS Target Statistics for Accounting Requests (SRC CLI)	587
Viewing RADIUS Target Statistics for Authentication Requests (SRC CLI)	587
Viewing Diameter Host Statistics (SRC CLI)	588
Viewing Diameter Peer Statistics (SRC CLI)	588

Part 9

Chapter 36

Controlling Volume Usage with the SRC VTA

Overview of Controlling Volume Usage with the SRC VTA 593

SRC VTA Overview	593
Terminology	594
SRC VTA Service and Subscriber Accounts	594
SRC VTA Sessions	595
Volume-Based Services	595
SRC VTA Architecture and Connections to Other SRC Components	595
How the SRC VTA Works	596
Events	597
Event Handlers	597
Actions	598
Processors	598
Db-Engine Processor	598
SAE Proxy Processor	600
Mailer Processor	600
Scripts Processor	600
SRC VTA Operation	601
Configuring Event Handlers Overview	602
Priority	603
Events	603
Event Attributes	604
Condition	605
Actions	605
Configuring Actions Overview	606
On Error	613
Managing SRC VTA Accounts and Sessions Overview	613
Identifying Subscribers, SAEs, and Sessions	613
Managing SRC VTA Accounts and Sessions	614
Adjusting the Interim Accounting Interval Overview	614
Locating the SAE That Manages a Subscriber for the SRC VTA	615
Using JavaScript Programs in SRC VTA Configurations	615
SRC VTA SOAP Interface	616
SRC VTA SOAP Interface URL	616
Example: Limiting Subscriber Access Based on Account Balances	616

Chapter 37

Prerequisites for Running the SRC VTA 619

Before You Configure the SRC VTA	619
Configuring the Web Application Server (SRC CLI)	621
Configuration Statements for the Web Application Server	621
Configuring Local Properties for the Web Application Server (SRC CLI)	622
Configuring the Web Application Server Shared Cluster Configuration (SRC CLI)	623

	Configuring the Nodes in the Web Application Server Cluster (SRC CLI)	624
	Configuring Remote Access to the Application Server (SRC CLI)	625
	Configuring Access to the Application Server Through Secure HTTP	625
	Configuring Access to the Application Server Through HTTP	626
	Configuring Virtual Hosts for the Web Applications (SRC CLI)	626
	Configuring User Accounts for Web Applications (SRC CLI)	627
	Installing Web Applications in the SRC Web Application Server	629
	Starting the Web Application Server on a C Series Controller	629
	Configuring the SAE to Send Tracking Events to the SRC VTA (SRC CLI)	630
	Specifying Tracking Plug-Ins for Enterprise Subscribers on Junos OS Routing Platforms	632
	Configuring the External Database	633
	Configuring a Database to Store Account and Session Data (SRC CLI)	633
	Installing the JDBC Driver .jar File	634
	Troubleshooting Database Deadlocks	634
	Configuring SRC VTA Services and Policies	635
	Configuring Subscribers and Subscriptions to SRC VTA Services	635
	Configuring a NIC for the SRC VTA (SRC CLI)	636
	Configuring NIC Proxies for the SRC VTA	636
Chapter 38	Configuring the SRC VTA (SRC CLI)	639
	Configuring an SRC VTA Shared Group Configuration (SRC CLI)	639
	Creating and Configuring an SRC VTA Shared Group Configuration (SRC CLI)	639
	Keys Used to Specify the Subscriber ID Solution (SRC CLI)	640
	Configuring the Connection Between the SRC VTA and the External Account and Session Database (SRC CLI)	642
	Cleaning Up the Sessions Records and SessionBalanceChanges Records from the SRC VTA Database (SRC CLI)	644
	Configuring Actions for SRC VTA Event Handlers (SRC CLI)	647
	Configuring Event Handlers (SRC CLI)	648
	Configuring Event Queues for SRC VTA Groups (SRC CLI)	650
	Configuring the DB-Engine Processor for the SRC VTA Group (SRC-CLI)	651
	Recording Balance Changes and Calculating the Average Usage Rate (SRC CLI)	651
	Configuring the Initial Balance and Status of a Subscriber Account in the External Database (SRC CLI)	652
	Configuring Scripts That Update Accounts (SRC CLI)	653
	Configuring the Interim Account Interval and Usage Metric of a Service in the External Database (SRC CLI)	654
	Configuring SRC VTA Database Optimization (SRC CLI)	655
	Variables Used to Define the Interim Accounting Interval for Services	656
	Current Service Variables	656
	Other Service Variables	657

	Account Balance Variable	658
	Sample Formulas for Interim Accounting Interval	658
	Variables Used to Define the Usage Metric for Services	659
	Sample Formulas for Usage Metrics for the SRC VTA	660
	Configuring the Mailer Processor for the SRC VTA Group (SRC CLI)	661
	Configuring the SRC VTA Mailer Processor to Send E-Mail Notifications (SRC CLI)	661
	Configuring the SRC VTA to Send E-Mail Notifications (SRC CLI)	662
	Configuring the SRC VTA Scripts Processor (SRC CLI)	662
	Configuring the SRC VTA to Run Scripts	663
	Configuring JavaScript Programs	663
	Configuring External Scripts	664
	Configuring SRC VTA Logging (SRC CLI)	666
	Logging SRC VTA Messages to a Text File	666
	Logging SRC VTA Messages to a System Logging Server	670
	Enabling, Disabling, and Restarting the SRC VTA (SRC CLI)	674
	Enabling the SOAP Interface for an SRC VTA Group (SRC CLI)	676
	Methods for the SRC Volume Tracking Application SOAP Interface	677
	SOAP Fault Codes for the SRC Volume Tracking Application	687
	Using One SRC VTA Account for Multiple Subscriber Sessions	698
Chapter 39	Managing the SRC VTA (SRC CLI)	701
	Deleting Balance Change History Records from the Database (SRC CLI)	701
	Deleting Session History Records from the Database (SRC CLI)	702
	Deleting Subscriber SRC VTA Accounts (SRC CLI)	702
	Modifying SRC VTA Accounts and Service Sessions (SRC CLI)	702
	Terminating Sessions (SRC CLI)	704
	Deleting Event Queues for SRC VTA Groups (SRC CLI)	704
Chapter 40	Monitoring and Testing the SRC VTA	705
	Viewing a Subscriber's SRC VTA Accounts (SRC CLI)	705
	Viewing Balance Change History for a Subscriber (SRC CLI)	705
	Viewing a Subscriber's Session History (SRC CLI)	706
	Viewing SRC VTA Performance Statistics (SRC CLI)	706
	Viewing SRC VTA SOAP API Statistics (SRC CLI)	709
	Testing the SRC VTA Configuration Overview	710
	Testing SRC VTA Events (SRC CLI)	713
Part 10	Index	
	Index	717

List of Figures

Part 1	Operating the SAE	
Chapter 1	Overview of the SAE	3
	Figure 1: SAE Plug-In Architecture	10
	Figure 2: SRC SAE APIs	12
	Figure 3: Remote Interface on the SAE	13
	Figure 4: Sending Accounting Data to a RADIUS Server	14
	Figure 5: Sending Accounting Data to an Accounting File	14
	Figure 6: Customer Choice for SRC Accounting Deployment	14
Part 3	Using Network Devices in the SRC Network	
Chapter 9	Integrating Third-Party Network Devices into the SRC Network (SRC CLI)	113
	Figure 7: SAE Community	114
	Figure 8: Login Interactions with Assigned IP Subscribers	116
	Figure 9: Login Interactions with Event Notification Application	117
Part 4	Locating Subscriber Management Information	
Chapter 10	Locating Subscriber Information with the NIC	133
	Figure 10: Communication Between a NIC Proxy and a NIC Host in Client/Server Mode	134
	Figure 11: Communication Between a NIC Host and a NIC Proxy in Local Host Mode	134
	Figure 12: NIC Groups	137
	Figure 13: NIC Group Selection by Round-Robin	138
	Figure 14: NIC Resolution Request	139
Chapter 17	NIC Configuration Scenarios	209
	Figure 15: Resolution Process for ip Realm	210
	Figure 16: OnePop Centralized Configuration	211
	Figure 17: OnePop Distributed Configuration	211
	Figure 18: Redundancy for OnePop Centralized Configuration	212
	Figure 19: Resolution Process for Pcmam Realm	213
	Figure 20: OnePopPcmam Centralized Configuration	214
	Figure 21: OnePopPcmam Distributed Configuration	214
	Figure 22: Resolution Process for dynamicip Realm	215
	Figure 23: OnePopDynamicip Centralized Configuration	216
	Figure 24: OnePopDynamicip Distributed Configuration	216
	Figure 25: Resolution Process for sharedip Realm	217
	Figure 26: OnePopSharedIP Centralized Configuration	218

	Figure 27: OnePopSharedIP Distributed Configuration	218
	Figure 28: Resolution Process for the StaticRouteIp Realm	219
	Figure 29: OnePopStaticRouteIp Centralized Configuration	220
	Figure 30: OnePopStaticRouteIp Distributed Configuration	220
	Figure 31: Resolution Process for the VrfIp Realm	221
	Figure 32: OnePopVrfIp Centralized Configuration	222
	Figure 33: OnePopStaticRouteIp Distributed Configuration	223
	Figure 34: Resolution Process for acctId Realm	223
	Figure 35: OnePopAcctId Centralized Configuration	225
	Figure 36: Resolution Processes login Realm	225
	Figure 37: OnePopLogin Centralized Configuration	227
	Figure 38: OnePopLogin Distributed Configuration	227
	Figure 39: OnePopLoginPull Distributed Configuration	228
	Figure 40: Resolution Processes for primary_user Realm	228
	Figure 41: OnePopPrimaryUser Centralized Configuration	229
	Figure 42: OnePopPrimaryUser Distributed Configuration	230
	Figure 43: OnePopDnSharedIp Realms Centralized Configuration	231
	Figure 44: OnePopDnSharedIp Realms Distributed Configuration	233
	Figure 45: OnePopAllRealms Centralized Configuration	235
	Figure 46: OnePopAllRealms Distributed Configuration	237
	Figure 47: Resolution Process for Tunnel Realm	238
	Figure 48: OnePopTunnel Centralized Configuration	239
	Figure 49: OnePopPrefixIp Configuration	240
	Figure 50: MultiPop Configuration	241
	Figure 51: IP Realm for MultiPop Configuration	242
	Figure 52: sharedIP Realm for MultiPop Configuration	243
	Figure 53: Resolution Graph for MultiPop dn Realm	244
	Figure 54: dn Realm for MultiPop Configuration	245
Part 5	Providing Admission Control with SRC ACP	
Chapter 18	Overview of Providing Admission Control with SRC ACP	249
	Figure 55: Position of SRC ACP in Network	250
Part 7	Using Session State Registrar	
Chapter 27	Session State Registrar Overview	369
	Figure 56: SSR with Four Data Nodes in Two Groups	372
	Figure 57: Basic Session State Registrar Cluster	373
	Figure 58: SSR Cluster with Four Data Nodes Forming Two-Node Groups	375
	Figure 59: SSR Cluster with Redundant Network	376
	Figure 60: SSR Cluster with Redundant Network	379
	Figure 61: SSR Cluster Divided Between Two Sites with Tertiary Management Server	382
	Figure 62: SSR Cluster Evenly Divided Between Two Sites	383
Chapter 28	Planning Your Session State Registrar Cluster	391
	Figure 63: Basic SSR Starter Kit Cluster	391

Part 8	Using the Subscriber Information Collector	
Chapter 32	Overview of the Subscriber Information Collector	433
	Figure 64: The Rendering Process	437
	Figure 65: Initial Authorization and Accounting Timing Sequence	439
	Figure 66: Activation and Deactivation Timing Sequences	440
	Figure 67: Abort Session Timing Sequence	441
	Figure 68: Data Flow	442
	Figure 69: Explicit Routing Rule Process	451
	Figure 70: SIC Editing Rule Process	453
	Figure 71: Editing and Accounting Routing Rule Conditions and Processes	455
	Figure 72: SIC RADIUS and Diameter Configuration	457
	Figure 73: SIC Diameter Configuration	458
	Figure 74: Upstream and Downstream Network Element Clients and Targets	459
	Figure 75: Upstream and Downstream Network Element Client and Target Configuration Options	459
	Figure 76: Round-Robin	461
	Figure 77: SNMP Support for the SIC	468
Part 9	Controlling Volume Usage with the SRC VTA	
Chapter 36	Overview of Controlling Volume Usage with the SRC VTA	593
	Figure 78: SRC VTA Architecture and Position in the SRC Network	596
	Figure 79: SRC VTA Event Handler Model	598
	Figure 80: Operation of the SRC VTA	601

List of Tables

	About the Documentation	xxxiii
	Table 1: Notice Icons	xxxiv
	Table 2: Text Conventions	xxxiv
Part 1	Operating the SAE	
Chapter 1	Overview of the SAE	3
	Table 3: Accounting Information Collection Based on the Accounting-Per-Attachment Feature for Previous Policy Lists	15
	Table 4: Accounting Information Collection Based on the Accounting-Per-Attachment Feature	16
Part 2	Using Juniper Networks Routers in the SRC Network	
Chapter 6	Using JunosE Routers in the SRC Network (SRC CLI)	53
	Table 5: Router Initialization Scripts	63
	Table 6: Exported Fields	64
Part 3	Using Network Devices in the SRC Network	
Chapter 9	Integrating Third-Party Network Devices into the SRC Network (SRC CLI)	113
	Table 7: Router Initialization Scripts	125
	Table 8: Exported Fields	126
Part 4	Locating Subscriber Management Information	
Chapter 10	Locating Subscriber Information with the NIC	133
	Table 9: Types of NIC Agents	136
	Table 10: NIC Configuration Scenarios	140
	Table 11: NIC Agents	144
	Table 12: Agents in Configuration Scenarios	145
	Table 13: Type of Router Initialization Script to Use for NIC Configuration Scenarios	147
Chapter 16	NIC Resolution Process	203
	Table 14: Available NIC Resolutions	203
Part 5	Providing Admission Control with SRC ACP	
Chapter 18	Overview of Providing Admission Control with SRC ACP	249
	Table 15: Congestion Points Derived Through NAS Port ID	252

Part 6	Using External Subscriber Monitor	
Chapter 25	Monitoring External Subscriber Events with the SRC CLI	359
	Table 16: Output Fields for show external-subscriber-monitor statistics	
	radius-accounting	359
	Table 17: Output Fields for show external-subscriber-monitor statistics	
	event-notifications	361
	Table 18: Output Fields for show external-subscriber-monitor statistics	
	process	362
Part 7	Using Session State Registrar	
Chapter 27	Session State Registrar Overview	369
	Table 19: Latency Between Servers and Its Effect on Performance	376
	Table 20: Supported Cluster Configurations	378
	Table 21: Subscriber Sessions Table Default Fields	384
	Table 22: Service Sessions Table Default Fields	387
Chapter 28	Planning Your Session State Registrar Cluster	391
	Table 23: Example Allocation of Node IDs	392
	Table 24: SSR Starter Kit Cluster Worksheet	393
	Table 25: Expanded SSR Cluster Planning Worksheet	394
Part 8	Using the Subscriber Information Collector	
Chapter 32	Overview of the Subscriber Information Collector	433
	Table 26: Example of SSR Database Mapping	449
	Table 27: SIC Log File Options	465
	Table 28: SIC Event Levels	466
	Table 29: SIC Log Groups	467
	Table 30: MIBs Used by the SIC for SNMP Statistics	468
	Table 31: SNMP Traps Supported for the SIC	468
Chapter 33	Configuring the Subscriber Information Collector with the SRC CLI	471
	Table 32: SIC Editing Rule Options	506
	Table 33: Explicit Routing Rule Conditions	513
	Table 34: Sample Configuration Attribute Associations	520
	Table 35: Log Groups and Associated Event Level for Log Stream=default	
	logger	521
	Table 36: Log Groups and Associated Event Level for Log	
	Stream=debug-logger	521
	Table 37: Log Groups and Associated Event Level for Log	
	Stream=error-logger	522
Chapter 34	Device and Service Templates for Dynamic Authorization (SRC CLI)	539
	Table 38: Device Template Capabilities and Associated Values	540
	Table 39: Service Template Modes	541
	Table 40: Global Service Template Modes	541
	Table 41: Attributes for All Modes	542
	Table 42: Variables	544
	Table 43: Capabilities and Associated Values	547

Part 9	Controlling Volume Usage with the SRC VTA	
Chapter 36	Overview of Controlling Volume Usage with the SRC VTA	593
	Table 44: SRC VTA Terms	594
	Table 45: SRC VTA Event Types	603
	Table 46: Event Attributes	604
	Table 47: SRC VTA Functions and Input Parameters	607
Chapter 37	Prerequisites for Running the SRC VTA	619
	Table 48: Settings for Filter Strings	631
Chapter 38	Configuring the SRC VTA (SRC CLI)	639
	Table 49: Keys That the SRC VTA Constructs to Manage Accounts and Sessions	641
	Table 50: Options Used to Configure the Cleanup Interval	645
	Table 51: Values Allowed for the Options	646
	Table 52: Examples of Interim Accounting Interval	658
	Table 53: Examples of Formulas That Calculate Use of Network Resources	660
	Table 54: Named Severity Levels	667
	Table 55: Examples of Filters for Event Messages	670
	Table 56: Named Severity Levels	671
	Table 57: Examples of Filters for Event Messages	674
	Table 58: SRC VTA Methods and SOAP Fault Codes	688
	Table 59: VTA SOAP Fault Codes Common to Methods	697
Chapter 39	Managing the SRC VTA (SRC CLI)	701
	Table 60: Arguments Used to Modify Accounts	703
Chapter 40	Monitoring and Testing the SRC VTA	705
	Table 61: SRC VTA Event Types and Test Event Configuration Statements	710
	Table 62: Attributes for Subscriber- and Service-Tracking Test Events	711

About the Documentation

- SRC Documentation and Release Notes on page xxxiii
- Audience on page xxxiii
- Documentation Conventions on page xxxiii
- Documentation Feedback on page xxxv
- Requesting Technical Support on page xxxv

SRC Documentation and Release Notes

For a list of related SRC documentation, see <http://www.juniper.net/techpubs/>.

If the information in the latest *SRC Release Notes* differs from the information in the SRC guides, follow the *SRC Release Notes*.

Audience

This documentation is intended for experienced system and network specialists working with routers running Junos OS and JunosE software in an Internet access environment. We assume that readers know how to use the routers, directories, and RADIUS servers that they will deploy in their SRC networks. If you are using the SRC software in a cable network environment, we assume that you are familiar with the PacketCable Multimedia Specification (PCMM) as defined by Cable Television Laboratories, Inc. (CableLabs) and with the Data-over-Cable Service Interface Specifications (DOCSIS) 1.1 protocol. We also assume that you are familiar with operating a multiple service operator (MSO) multimedia-managed IP network.

Documentation Conventions

[Table 1 on page xxxiv](#) defines the notice icons used in this guide. [Table 2 on page xxxiv](#) defines text conventions used throughout this documentation.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.

Table 2: Text Conventions

Convention	Description	Examples
Bold text like this	<ul style="list-style-type: none"> Represents keywords, scripts, and tools in text. Represents a GUI element that the user selects, clicks, checks, or clears. 	<ul style="list-style-type: none"> Specify the keyword exp-msg. Run the install.sh script. Use the pkgadd tool. To cancel the configuration, click Cancel.
Bold text like this	Represents text that the user must type.	user@host# set cache-entry-age cache-entry-age
Fixed-width text like this	Represents information as displayed on your terminal's screen, such as CLI commands in output displays.	<pre>nic-locators { login { resolution { resolver-name /realms/ login/A1; key-type LoginName; value-type SaeId; } } }</pre>
Regular sans serif typeface	<ul style="list-style-type: none"> Represents configuration statements. Indicates SRC CLI commands and options in text. Represents examples in procedures. Represents URLs. 	<ul style="list-style-type: none"> system ldap server{ stand-alone; Use the request sae modify device failover command with the force option user@host# ... http://www.juniper.net/techpubs/software/management/src/api-index.html
<i>Italic sans serif typeface</i>	Represents variables in SRC CLI commands.	user@host# set local-address local-address
Angle brackets	In text descriptions, indicate optional keywords or variables.	Another runtime variable is <gfwif>.
Key name	Indicates the name of a key on the keyboard.	Press Enter.

Table 2: Text Conventions (*continued*)

Key names linked with a plus sign (+)	Indicates that you must press two or more keys simultaneously.	Press Ctrl + b.
<i>Italic typeface</i>	<ul style="list-style-type: none"> Emphasizes words. Identifies book names. Identifies distinguished names. Identifies files, directories, and paths in text but not in command examples. 	<ul style="list-style-type: none"> There are two levels of access: <i>user</i> and <i>privileged</i>. <i>SRC PE Getting Started Guide</i> <i>o=Users, o=UMC</i> The <i>/etc/default.properties</i> file.
Backslash	At the end of a line, indicates that the text wraps to the next line.	Plugin.radiusAcct-1.class=\net.juniper.smgmt.sae.plugin\RadiusTrackingPluginEvent
Words separated by the symbol	Represent a choice to select one keyword or variable to the left or right of this symbol. (The keyword or variable may be either optional or required.)	diagnostic line

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to techpubs-comments@juniper.net, or fill out the documentation feedback form at <https://www.juniper.net/cgi-bin/docbugreport/>. If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release version (if applicable)

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at <http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf>.
- Product warranties—For product warranty information, visit <http://www.juniper.net/support/warranty/>.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: <http://www.juniper.net/customers/support/>
- Search for known bugs: <http://www2.juniper.net/kb/>
- Find product documentation: <http://www.juniper.net/techpubs/>
- Find solutions and answer questions using our Knowledge Base: <http://kb.juniper.net/>
- Download the latest versions of software and review release notes: <http://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications: <https://www.juniper.net/alerts/>
- Join and participate in the Juniper Networks Community Forum: <http://www.juniper.net/company/communities/>
- Open a case online in the CSC Case Management tool: <http://www.juniper.net/cm/>

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: <https://tools.juniper.net/SerialNumberEntitlementSearch/>

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at <http://www.juniper.net/cm/>.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see <http://www.juniper.net/support/requesting-support.html>.

PART 1

Operating the SAE

- Overview of the SAE on page 3
- Configuring the SAE (SRC CLI) on page 19
- Managing Subscriber and Service Session Data (SRC CLI) on page 29
- Managing SAE Data (SRC CLI) on page 37
- Managing SAE Data (C-Web Interface) on page 45

CHAPTER 1

Overview of the SAE

- [Role of the SAE on page 3](#)
- [Connections to Managed Devices on page 3](#)
- [SAE Support for Dual-Stack Configuration on page 5](#)
- [SAE Plug-Ins on page 9](#)
- [Tracking and Controlling Subscriber and Service Sessions with SAE APIs on page 11](#)
- [SAE Accounting on page 13](#)

Role of the SAE

The SAE is the core manager of the SRC network. It interacts with other systems, such as Juniper Networks routers, cable modem termination system (CMTS) devices, directories, Web application servers, and RADIUS servers, to retrieve and disseminate data in the SRC environment. The SAE authorizes, activates and deactivates, and tracks subscriber and service sessions. It also collects accounting information about subscribers and services.

The SAE makes decisions about the deployment of policies on JunosE routers and devices running Junos OS. When a subscriber's IP interface comes up on the router, the SAE determines whether it manages the interface. If the interface is managed—or controlled—by the SAE, the SAE sends the subscriber's default policy configuration to the router. These default policies define the subscriber's initial network access. When the subscriber activates a value-added service, the SAE translates the service into lists of policies and sends them to the router.

The SAE also provides plug-ins and application programming interfaces (APIs) that extend the capabilities of the SRC software.

Related Documentation

- [Tracking and Controlling Subscriber and Service Sessions with SAE APIs on page 11](#)
- [Connections to Managed Devices on page 3](#)

Connections to Managed Devices

This topic describes the connections between the SAE and Juniper Networks routers, CMTS devices, and the Juniper Policy Server (JPS).

COPS Connection Between JunosE Routers and the SAE

The SAE and JunosE routers communicate using the Common Open Policy Service (COPS) protocol. The SAE supports two versions of COPS:

- COPS usage for policy provisioning (COPS-PR)
- COPS External Data Representation Standard (XDR) mode

The version of COPS that you use depends on the version of COPS that your JunosE router supports. When you set up your JunosE router to work with the SAE, you enable either COPS-PR mode or COPS XDR mode. There are no configuration differences on the SAE between COPS-PR and COPS XDR.

The following SRC features require the use of COPS-PR:

- Policy sharing on JunosE routers
- Multiple classify traffic conditions in policy lists

Beep Connection Between Devices Running Junos OS and the SAE

The SAE interacts with a Junos OS process, referred to as the SRC software process, on a device running Junos OS. The SAE and the SRC software process communicate using the Blocks Extensible Exchange Protocol (BEEP).

When a device running Junos OS that the SAE manages goes online, it initiates a BEEP session for the SAE. The SAE gets configuration information from the router, and then it builds and installs the policies that control the router's behavior. If the policies are subsequently modified in the directory, the SAE builds a new configuration and reconfigures the interface on the device running Junos OS.



NOTE: The SAE manages interfaces on devices running Junos OS only when the interfaces are configured in the global configuration and the router sends added, changed, or deleted notifications to the SAE. Router administrators should not manually change the configuration of interfaces that the SAE is managing. If you manually change a configuration, you must remove the SAE from the system.

When there are configuration changes on the router, the router sends a notification to the SAE through the BEEP connection. The notification does not include the content of the configuration changes. When the SAE receives the notification, it uses its Junos XML management protocol client to get the changed configuration from the router.

Interfaces that have been deleted from the router along with their associated objects (sessions, policies) remain on the router until state synchronization occurs.

COPS Connection Between CMTS Devices and the SAE

The SAE uses the COPS protocol as specified in the PacketCable Multimedia Specification PKT-SP-MM-I03-051221 to manage *PacketCable Multimedia Specification*

(PCMM)-compliant CMTS devices in a cable network environment. The SAE connects to the CMTS device by using a COPS over Transmission Control Protocol (TCP) connection.

In cable environments, the SAE manages the connection to the CMTS device. The CMTS device does not provide address requests or notify the SAE of new subscribers, subscriber IP addresses, or any other attributes. IP address detection and all other subscriber attributes are collected outside of the COPS connection to the CMTS device. The SAE uses COPS only to push policies to the CMTS device and to learn about the CMTS status and usage data.

Because the CMTS device does not have the concept of interfaces, the SRC module uses pseudointerfaces to model CMTS subscriber connections similar to subscriber connections for devices running Junos OS and JunosE routers.

COPS Connection Between Juniper Policy Servers and the SAE

When the SAE is acting as an application manager in a PCMM environment, it connects to the JPS through an interface on the JPS. The JPS uses the COPS protocol as specified in the PacketCable Multimedia Specification PKT-SP-MM-I03-051221 for its interface connections. The JPS communicates with the application manager by using a COPS over TCP connection.

For more information, see .

Related Documentation

- *PCMM Environment Overview*
- *JPS Overview*
- [Configuring the SAE to Manage Devices Running Junos OS \(SRC CLI\) on page 86](#)
- [Adding JunosE Routers and Virtual Routers \(SRC CLI\) on page 54](#)

SAE Support for Dual-Stack Configuration

JunosE supports configuration of Internet protocol versions—IPv4 and IPv6—on a single interface. This creates two IP layer interfaces (IPv4 and IPv6), known as dual-stack, that run and report independently.

To support dual-stack configuration, the functionality of the SAE is configured to create a single subscriber session for a set of IP interfaces. All services activated for a subscriber session impacts the related interfaces. For example, when both IPv4 and IPv6 interfaces exist for a subscriber session, a service activation installs policies on both the interfaces.



NOTE: Each dual-stack interface consumes about twice the memory resources than a single interface. The total number of subscriber sessions supported by the SAE for dual-stack interfaces would be cut by half.

- For C3000, the maximum number of dual-stack subscriber sessions is 150,000 (using two active services per subscriber).
- For C5000, the maximum number of dual-stack subscriber sessions is 500,000 (using two active services per subscriber).

The performance in login/logout rate, as well as service activation/deactivation rate of a dual-stack interface would be about 50% when compared to the performance of a single-stack interface.

Handling of Interface-Up Notification

In dual-stack configuration, the interfaces are reported by the router independently. The router sends two interface-up notifications that are tied together by their interface name. The default policies for a subscriber session are then applied independently on these interfaces.

The dual-stack interface-up notification sequence is as follows:

1. When the router receives the first IP interface-up request, it acquires a lock on the interface.

For example, if an IPv4 interface is reported first, the router acquires a lock for the interface name and holds on to it until the request is processed.

The router keeps track of the underlying interfaces (IPv4 and IPv6) and communicates with the SAE to create appropriate policies for each interface.

2. The router processes the IP interface-up request, creates the user session, and installs default policies.

If default policies are not defined, processing for the interface stops and the interface remains unmanaged.

3. The activate-on-login service session provisions the relevant policies (IPv4 and IPv6) for the interface.
4. The router releases the lock on the IP interface after the request has been processed.
5. When the router receives the second IP interface-up request, it attempts to acquire a lock on the IP interface.

If the processing for the first IP interface request is not completed, the router handler thread blocks the second IP interface-up notification until the lock is released for the first IP interface.

6. The router processes the second interface-up request and installs the default policies.

If default policies are not defined, processing for the interface stops and the interface remains unmanaged.

7. The router checks for the user session that is currently associated with the interface name.

If a user session already exists, the request is handled in the similar way to an update request and a relogin is triggered. If the relogin does not result in the termination of the existing user session, all active service sessions are notified of the modification.

8. The activate-on-login service session provisions the relevant policies on the second IP interface.
9. The router releases the lock on the second IP interface after the request has been processed.

Handling of Interface-Up Notification with Delay Timer

The delay timer (dual-stack-delay) is configured to reduce the number of interface-up notifications. This is useful if most of the interfaces are expected to be dual-stack interfaces because it reduces the overhead of relogin and update plug-in events.



NOTE: The dual-stack-delay attribute is not enabled by default.

The sequence of interface-up notifications with delay timer is as follows:

1. When the router receives the first IP interface-up request, it processes the request and installs the default policies.
2. The router creates an IP interface context that represents the Common Open Policy Service (COPS) object.
3. The IP interface context schedules a timer and suspends further processing. This postpones the creation of user sessions.
4. After provisioning the default policies on the second IP interface, the IP interface context cancels the timer for the first IP interface.
5. The user session is created for the dual-stack interface.
6. The activate-on-login service provisions the service policies that are applied to both interfaces of the dual-stack.



NOTE: If the first IP interface request is not part of a dual-stack interface, then the creation of the user session is delayed until the timer expires. This causes a delay in the login rate for a single-stack interface.

If the second IP interface request is received after the expiry of the delay timer, the user session is created and a relogin is triggered and events are updated.

Handling of Interface-Down Notification

The sequence of interface-down notifications for dual-stack configuration is as follows:

1. When the first IP Interface begins to shut down, the router sends a COPS delete request (DRQ) message.
2. The router stores final accounting for all active policies of the first IP interface to the associated IP interface context and notifies the SAE that the interface is down.

Because the IP interface is also associated with a second IP interface context, no further action is triggered.
3. The router sends final accounting for all active policies of the second IP interface to the associated IP interface context.
4. The router notifies the SAE that the second IP interface is down.
5. The router driver initiates the logout of the subscriber session because both the interfaces are down.
6. The subscriber session deactivates all active services and the service session removes the installed policies.
7. After the subscriber session is logged out, the interface contexts for IPv4 and IPv6 are discarded.
8. The router driver returns the accounting data for the individual policies.

Service Activation

The router keeps track of the underlying interfaces and communicates with the policy engine to create the appropriate policies for each interface. The service session provisions the relevant policies (IPv4 and IPv6) for the interface.

Service Deactivation

When the interface is down, the router notifies the SAE and deactivates the service session. The service session passes the stored provisioning sets (IPv4 and IPv6) to the router driver. The router driver removes the policies and returns the accounting data for the individual policies. Accounting data for services contains the sum of IPv4 and IPv6 counters.

Subscriber Attributes

The `framedIpv6Prefix` and `delegatedIpv6Prefix` attributes are added to the subscriber object and can be queried through the `SAEAccess` API module.

- The `framedIpv6Prefix` attribute contains the IPv6 prefix for the subscriber.

`framedIpv6Prefix` is available for JunosE (COPS-PR), Junos OS (JSRC), as well as AAA (COA).
- Using the `delegatedIpv6Prefix` attribute, the NAS can receive a set of IPv6 prefixes that are delegated to subscribers.

An IPv6 subscriber can be identified through multiple prefixes by using the `delegatedIpv6Prefix` attribute together with the `framedIpv6Prefix` attribute.

`delegatedIpv6Prefix` is available for Junos OS (JSRC), AAA (COA), and DHCPv6 subscribers on the JunosE router.

SAEAccess API Plug-in Attributes

The SAEAccess API plug-in is extended to support the following attributes:

- `PA_FRAMED_IPV6_PREFIX`—An octet string formatted as specified in RFC3162. The first octet is 0, the second octet contains the length of the prefix in bits (1–128), and the following octet contains the actual prefix.
- `PA_DELEGATED_IPV6_PREFIX`—An octet string formatted as specified in RFC4818. The first octet is 0, the second octet contains the length of the prefix in bits (1–128), and the following octet contains the actual prefix.
- `PA_USER_IP_MASK`—Number of bits in the subscriber address (`PA_USER_INET_ADDRESS`). This attribute is available both for IPv4 and IPv6 addresses, if the underlying router driver provides the value, which is presently the case for JunosE (COPS-PR), Junos OS (JSRC), and AAA (COA).

For dual-stack interfaces, the `PA_USER_INET_ADDRESS` attribute contains the IPv4 address of the subscriber, whereas `PA_FRAMED_IPV6_PREFIX` contains the IPv6 address prefix.

For single-stack interfaces, the `PA_USER_INET_ADDRESS` attribute contains either the IPv4 or IPv6 address depending on the address family assigned to the subscriber.

Subscriber Session Lookup

Subscribers are identified by the set of IPv6 prefixes defined by the device driver. If both `framedIpv6Prefix` and `delegatedIpv6Prefix` are present for a subscriber session, then any IPv6 address that matches any one of the prefixes identifies the subscriber session.

For example, a subscriber session is reported by a device driver with the `framedIpv6Prefix` = 2001:db8:1:1::/64 and `delegatedIpv6Prefix` = 2001:db8:2:2::/64 attributes. Any IP address starting with one of these prefixes identifies the subscriber session (2001:db8:1:1:0:1:2:3, 2001:db8:2:2:4:3:2:1, and so on).

Related Documentation

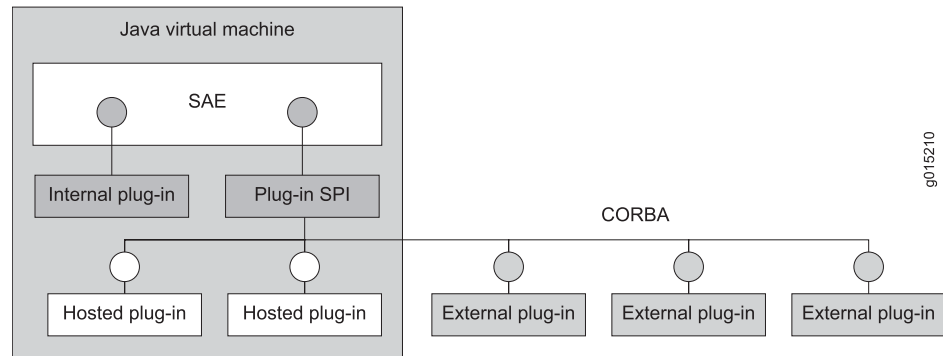
- *PPP Subscriber Login and Service Activation*
- [OnePopPrefixIp Scenario on page 239](#)

SAE Plug-Ins

Plug-ins are software programs that extend the capabilities of existing programs and make them more flexible. SRC plug-ins provide authentication, authorization, and tracking capabilities.

There are three types of plug-ins: internal, hosted, and external. Internal plug-ins communicate directly with the SAE. Hosted and external plug-ins implement a published Common Object Request Broker Architecture (CORBA)-based service provider interface (SPI), which means that anyone with access to the interface specification can create plug-ins that work with the SRC module. [Figure 1 on page 10](#) gives an overview of the plug-in architecture.

Figure 1: SAE Plug-In Architecture



Internal Plug-Ins

The SRC module provides internal plug-ins that perform a range of authentication, authorization, and tracking functions. With these plug-ins, you can, for example, authenticate subscribers, authorize subscriptions and sessions, authorize IP address requests from DHCP clients, track subscriber activity and service use, track quality of service (QoS) services and attach and remove QoS profiles as needed, and limit the number of authenticated subscribers who connect to an IP interface on the router.

Internal plug-ins implement an interface that communicates directly with the SAE. They have the following characteristics:

- Run within the SAE's Java Virtual Machine (JVM)
- Are started and stopped with the SAE
- Are implemented in Java

The core SRC module provides a set of internal plug-ins.

External Plug-Ins

The SRC module includes the SAE CORBA plug-in SPI. This SPI allows you to implement external plug-ins in any language that supports CORBA (for example, Java, C++, Python), which makes it easy to integrate the SAE with operations support system (OSS) software written in a wide variety of languages and distributed across a variety of hardware and operating system platforms.

External plug-ins link a service provider's OSS with the SAE so that the OSS is notified of events in the life cycle of SAE sessions. For example, plug-ins can be notified when a subscriber attempts to log in and begins the authentication and authorization process. This notification makes it possible for the plug-in to consult general data and resource

allocation information that is available to the OSS, and use that information to make authorization decisions.

The SPI also sends session-tracking events when sessions start, on an interim basis, and when sessions stop. Plug-ins can set session timeouts as a response to both session start and interim events. This capability enables the development of prepaid applications where the plug-in consults the subscriber's current account balance before it makes the decision to extend or reduce a session timeout.

External plug-ins have the following characteristics:

- Run outside the SAE's JVM, either in the same or in a different server
- Are implemented in any language that supports CORBA
- Communicate with the SAE using CORBA
- Support the admission control or prepaid demo plug-in, which can be purchased separately from the SRC module.

Hosted Plug-Ins

Hosted plug-ins, like the external ones, implement the CORBA interface. Unlike the external ones, hosted plug-ins are instantiated (that is, hosted) by the SAE. As a result, they live in the same JVM process as the host SAE, which means that hosted plug-ins must be implemented in Java.

Hosted plug-ins have the following characteristics:

- Run within the SAE's JVM
- Communicate with SAE using CORBA
- Are started and stopped with SAE
- Are implemented using a published interface

Related Documentation

- *How Internal Plug-Ins Work*
- [Connections to Managed Devices on page 3](#)
- *Configuring the SAE for External Plug-Ins (SRC CLI)*
- The interface definition language (IDL) code and online documentation for the SAE CORBA Plug-In SPI is on the Juniper Networks Web site at <https://www.juniper.net/support/products/src/index.html>

Tracking and Controlling Subscriber and Service Sessions with SAE APIs

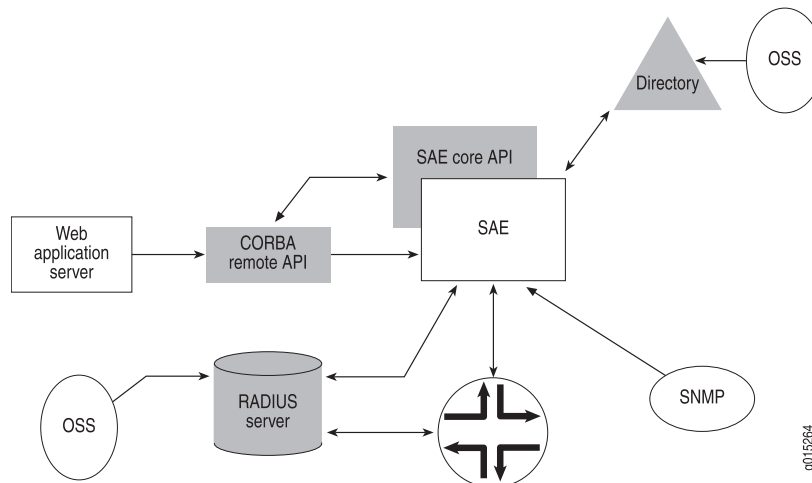
The SAE provides two public APIs:

- SAE core API
- SAE CORBA remote API

Through these interfaces, an external application can track and control subscriber and service sessions.

Figure 2 on page 12 illustrates the SAE APIs.

Figure 2: SRC SAE APIs



SAE Core API

The SAE core API is used to control the behavior of the SRC module. There are many uses of the SAE core API. For example, it can be used to provide:

- Subscriber credentials (username/password)
- Requests for service activation/deactivation for a subscriber

This API can be used by a Java application running in the same JVM as the SAE. For example, you can access the SAE core API from plug-ins that are hosted by the SAE, or you can use the SAE core API to write your own extensions of the SAE remote interface by using CORBA or the SAE script interface modules.

SAE CORBA Remote API

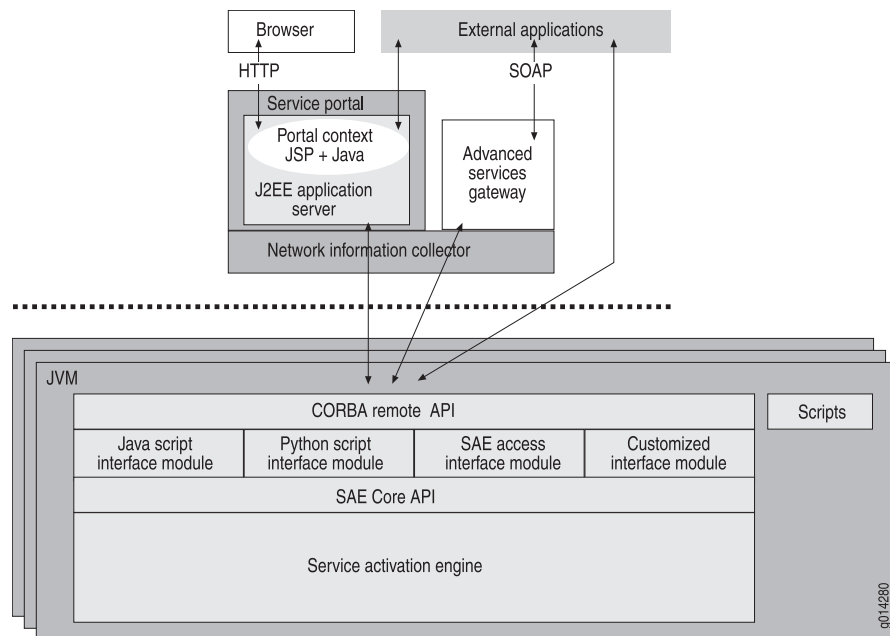
This API provides a way to use external applications with the SRC module (see [Figure 3 on page 13](#)). All functions that are available through the SAE core API are available through the CORBA remote API. The remote API provides several remote interfaces that allow customization of the API for special needs. The remote interface comprises an interface module manager and a set of interface modules. We provide the following interface modules with the SRC module:

- SAE access interface module—Provides remote access to the SAE core API
- Java script interface module—Allows you to control the SAE with a Java script
- Python script interface module—Allows you to control the SAE with a Python script
- Event notification interface module—Allows you to integrate the SAE with external IP address managers

You can also create custom interface modules that allow external applications to extend the capabilities of the SAE. To do so, you must define the interface module in CORBA IDL and implement it in Java.

The remote interface publishes one object reference that acts as the interface module manager. External applications communicate through CORBA with the interface module manager to retrieve a particular interface module. That interface module runs in the same JVM as the SAE and has full access to the SAE core API.

Figure 3: Remote Interface on the SAE



For more information about the SAE CORBA remote API, including the interfaces, properties, and methods, see the online documentation on the Juniper Networks Web site at <http://www.juniper.net/techpubs/software/management/src/api-index.html>.

Related Documentation

- [Storing Subscriber and Service Session Data on page 29](#)
- [Configuring Access to Subscriber Data \(SRC CLI\) on page 21](#)
- [Configuring Access to Service Data \(SRC CLI\) on page 23](#)
- [Configuring Access Through LDAPS to Service and Subscriber Data \(SRC CLI\) on page 20](#)

SAE Accounting

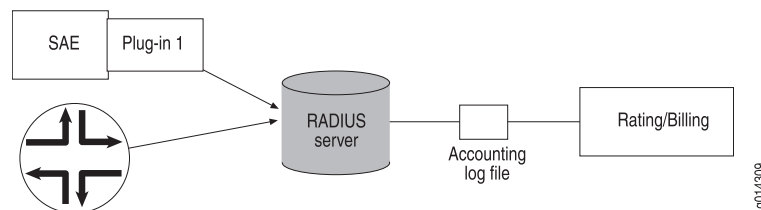
The router and the SAE generate RADIUS accounting records when subscribers access the Internet and use value-added services. The records are sent to RADIUS accounting servers and are logged in accounting log files, or they are sent to accounting flat files. External systems collect the accounting log files and feed them to a rating and billing system.

The SRC module allows a variety of accounting deployments. This topic shows the standard deployment that we supply, a second option that does not depend on a RADIUS server, and a third option in which customers develop their own deployment by choosing a CORBA plug-in.

In the standard SRC deployment (see [Figure 4 on page 14](#)), the router and the SAE are clients of the RADIUS accounting server. They pass subscriber accounting information to a designated RADIUS accounting server in an accounting request. The RADIUS accounting server receives the accounting request and creates accounting log files.

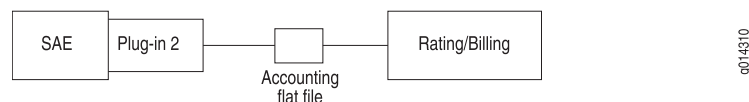
The SRC module works with other AAA RADIUS servers; however, we validate the SRC module only with Merit, Interlink RAD-Series AAA RADIUS Server, or Juniper Networks Steel-Belted Radius/SPE server.

Figure 4: Sending Accounting Data to a RADIUS Server



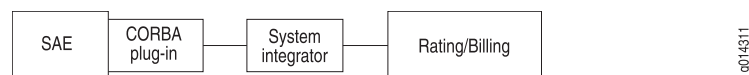
A second option, shown in [Figure 5 on page 14](#), uses an accounting flat file generated directly by the SAE, without a RADIUS server.

Figure 5: Sending Accounting Data to an Accounting File



[Figure 6 on page 14](#) illustrates a third possibility, one in which the customer uses a CORBA plug-in of his or her own choice.

Figure 6: Customer Choice for SRC Accounting Deployment



Accounting Policy

The SAE defines the policies that control the network traffic for the subscriber based on the subscriber's subscriptions. It also determines the accounting statistics collected for the subscribed service.

While defining the policies for a service, the SAE can choose the policy rules to be used for accounting per interface direction (ingress and egress). Statistics are collected for the chosen policy rules for the service and are sent to the RADIUS accounting server. The SAE can also decide not to collect any policy rule-specific statistics for the service. In this case, only session times are sent to the accounting system when the service is deactivated. When choosing multiple policy rules on traffic direction for statistics collection, the SAE summarizes the statistics by adding the individual values.

The SRC software allows you to enable or disable the accounting-per-attachment feature to calculate accounting for each individual policy list on the basis of the accounting flag status of each policy rule in the policy list. This capability optimizes the creation of statistics for the policy list. When the accounting-per-attachment feature is enabled for a policy list, accounting statistics are calculated per policy list only if the accounting flag is enabled on any one of the policy rules in that policy list. If the accounting flag of all policy rules in the policy list is disabled, the accounting statistics are not calculated for all policy lists even though the accounting-per-attachment feature is enabled. When the accounting-per-attachment feature is disabled for a policy list, accounting statistics are not calculated per policy list irrespective of the accounting flag status of all policy rules in that policy list. By default, the accounting-per-attachment feature is not set. If the accounting-per-attachment feature is not configured, the accounting statistics are calculated for all policy lists irrespective of the accounting flag status of all policy rules in that policy list.



NOTE: The accounting-per-attachment feature is supported only on E Series routers.

Any change in the enabling or disabling of the accounting-per-attachment feature takes effect only after the sssc connection is restarted. After you restart the sssc connection, accounting statistics are calculated for the updated policy list on the basis of the configuration of the accounting-per-attachment feature. Accounting statistics are calculated when the policy list is attached to an interface or subscriber.

When the attached policy list is changed, the SRC software sends a new policy attachment message containing the recalculated accounting-per-attachment statistics flag to the subscriber who uses the policy list. If the policy list is shared by subscribers, then the SRC software sends a new policy attachment message containing the recalculated accounting-per-attachment statistics flag to all subscribers who use the policy list. The policy list attached to the interface or subscriber can be changed for the following reasons:

- Policy structure change made by the administrator
- Activation or deactivation of manual services

Table 3 on page 15 lists cases for the collection of various accounting information based on the accounting-per-attachment feature for previous policy lists attached to the interface or subscriber.

Table 3: Accounting Information Collection Based on the Accounting-Per-Attachment Feature for Previous Policy Lists

Accounting-Per-Attachment Feature for Previous Policy List	Final Accounting Information Collection	Policy Change Accounting Information Collection
Enabled	Information is collected from the router when the policy list attached to the interface or subscriber is changed.	Information for the previous policy is received from the router when the new policy is attached for the interface or subscriber.

Table 3: Accounting Information Collection Based on the Accounting-Per-Attachment Feature for Previous Policy Lists (*continued*)

Accounting-Per-Attachment Feature for Previous Policy List	Final Accounting Information Collection	Policy Change Accounting Information Collection
Disabled	Information is not collected from the router when the policy list attached to the interface or subscriber is changed.	Information for the previous policy is not received from the router when the new policy is attached for the interface or subscriber.
Null	Information is collected from the router when the policy list attached to the interface or subscriber is changed.	Information for the previous policy is received from the router when the new policy is attached for the interface or subscriber.

[Table 4 on page 16](#) lists cases for the collection of various accounting information based on the accounting-per-attachment feature.

Table 4: Accounting Information Collection Based on the Accounting-Per-Attachment Feature

Accounting-Per-Attachment Feature	Interim Accounting Information Collection	Rule-Level Accounting Information Collection
Enabled	Information is collected from the router for the interface or subscriber.	Information is collected from the router for a policy rule if the accounting flag for that rule is enabled. Otherwise, information is not collected from the router.
Disabled	Information is not collected from the router for the interface or subscriber.	Information is not collected from the router for a policy rule.
Null	Information is collected from the router for the interface or subscriber.	Information is collected from the router for a policy rule.

Subscription Process

After an outsourced service has been set up, subscribers can order primary access or value-added services from retailers, who in turn notify the wholesaler of the new end subscription. Conversely, accounting data is collected by the wholesaler and communicated to the retailer to provide enough data for the retailer to bill the subscriber.

The overall subscription process is simplified:

- The subscriber has no need to interact with another party or a device other than the router.
- When the subscriber goes to the Web portal and selects the service, the subscription activation is triggered.
- The subscriber's portal page adjusts to display the new service.
- Accounting data is generated, identifying the service being tracked for the subscriber.

Tracking Subscriber Sessions

The intelligent service accounting function of the SRC module tracks the subscription activity for each subscriber and each service session. It collects usage information and passes the information to the appropriate rating and billing system.

Multiple service sessions can be activated simultaneously for a subscriber and can be tracked separately from an accounting standpoint.

Events are generated when service sessions are activated and deactivated, and during interim accounting updates.

Accounting Plug-Ins

Plug-ins allow service providers to easily extend the capabilities of their systems through the use of plug-in software. See [“SAE Plug-Ins” on page 9](#).

Interim Accounting

The router and SAE generate interim accounting records for broadband primary services (through PPP) and value-added services, respectively. RADIUS servers log the interim records in their accounting log files when interim accounting is enabled.

The external rating system calculates the charges by using interim records instead of stop records for timeout sessions. The calculation occurs when the last record is interim and for open sessions whose last record at the end of a billing cycle is interim.

An accounting interim interval is defined for each service and applied to all subscriptions to that service. The router and SAE generate accounting requests with a status of interim for every period of time specified with the interim value.

The router receives an accounting interim value for a session through a RADIUS server when the router makes an authentication request. If the RADIUS server does not provide a value, then the router does not generate interim accounting records.

The SAE obtains an accounting interim value from the directory. When the accounting interim value is not stored, the SAE uses global values. When a value equals zero, the SAE does not generate interim accounting records.

Related Documentation

- [Role of the SAE on page 3](#)
- [Connections to Managed Devices on page 3](#)
- [SAE Plug-Ins on page 9](#)
- [Tracking and Controlling Subscriber and Service Sessions with SAE APIs on page 11](#)
- [Adding Routers Individually \(SRC CLI\) on page 55](#)
- [Configuring the SAE to Manage JunosE Routers \(SRC CLI\) on page 59](#)

CHAPTER 2

Configuring the SAE (SRC CLI)

- [SRC Access to Directory Data on page 19](#)
- [Configuring LDAP Access to Directory Data \(SRC CLI\) on page 20](#)
- [Configuring Access Through LDAPS to Service and Subscriber Data \(SRC CLI\) on page 20](#)
- [Configuring Access to Subscriber Data \(SRC CLI\) on page 21](#)
- [Configuring Access to Service Data \(SRC CLI\) on page 23](#)
- [Configuring Access to Policy Data \(SRC CLI\) on page 24](#)
- [Configuring Access to the Persistent Login Cache \(SRC CLI\) on page 25](#)
- [Configuring the Location of Network Device Data \(SRC CLI\) on page 26](#)
- [Enabling Automatic Discovery of Changes in SAE Configuration Data \(SRC CLI\) on page 27](#)
- [Setting the Timeout and Number of Events for SAE Directory Eventing \(SRC CLI\) on page 28](#)

SRC Access to Directory Data

The SRC module stores subscriber, service, persistent login, policy, router, and cached subscriber profiles and session data in a directory. The SAE uses LDAP to store and retrieve the data.

If you do not store data in the local directory, you need to configure the LDAP connections to the directories in which the data is stored. You can also select the filter that the SAE uses to search for subscriptions in the directory and directory eventing parameters for data stored in the directory.

Related Documentation

- [Storing Subscriber and Service Session Data on page 29](#)
- [Configuring LDAP Access to Directory Data \(SRC CLI\) on page 20](#)
- [Configuring Access to the Persistent Login Cache \(SRC CLI\) on page 25](#)
- [Configuring Access to Policy Data \(SRC CLI\) on page 24](#)
- [Configuring Access Through LDAPS to Service and Subscriber Data \(SRC CLI\) on page 20](#)

Configuring LDAP Access to Directory Data (SRC CLI)

The tasks to configure LDAP access to directory data are:

- (Optional) “Configuring Access Through LDAPS to Service and Subscriber Data (SRC CLI)” on page 20
- Configuring Access to Subscriber Data (SRC CLI) on page 21
- Configuring Access to Service Data (SRC CLI) on page 23
- Configuring Access to Policy Data (SRC CLI) on page 24
- Configuring Access to the Persistent Login Cache (SRC CLI) on page 25
- Configuring the Location of Network Device Data (SRC CLI) on page 26
- Enabling Automatic Discovery of Changes in SAE Configuration Data (SRC CLI) on page 27
- Setting the Timeout and Number of Events for SAE Directory Eventing (SRC CLI) on page 28

Related Documentation

- *Configuring LDAP Access to Directory Data (C-Web Interface)*
- *Storing Subscriber and Service Session Data on page 29*
- *SRC Access to Directory Data on page 19*
- *Connections to Managed Devices on page 3*

Configuring Access Through LDAPS to Service and Subscriber Data (SRC CLI)

You can secure connections between a router and an external directory that contains service data or subscriber data, and you can configure the router to use LDAPS when it connects to the same data source.

Use the following configuration statements to configure access through LDAPS to service data and subscriber data:

```
shared sae configuration ldap service-data {  
    (ldaps);  
}  
  
shared sae configuration ldap subscriber-data {  
    (ldaps);  
}
```

To use LDAPS to secure connections between a router and an external directory:

1. Configure the directory connection from the SAE to use LDAPS. For example:

```
user@host# set shared sae configuration ldap service-data ldaps  
user@host# set shared sae configuration ldap subscriber-data ldaps
```

2. In the router initialization script you specify the directory context.

The `/opt/UMC/sae/lib/poolPublisher.py` script and the `/opt/UMC/sae/lib/lorPublisher.py` script provide examples of how to configure a directory context. For example, from the `/opt/UMC/sae/lib/lorPublisher.py` script:

```
dirContext = Ssp.registry.get('ServiceDataSource.component').getContext()
```

In addition, you can change the directory context.

For information about how to use `InitialDirContext` class or the `DirContext` class to specify directory context, see:

<http://java.sun.com/j2se/1.4.2/docs/api/javax/naming/directory/InitialDirContext.html>
<http://java.sun.com/j2se/1.4.2/docs/api/javax/naming/directory/DirContext.html>

Related Documentation

- [Configuring Access to Subscriber Data \(SRC CLI\) on page 21](#)
- [Configuring Access Through LDAPS to Service and Subscriber Data \(C-Web Interface\)](#)
- [Configuring Access to Service Data \(SRC CLI\) on page 23](#)
- [Configuring Access to Policy Data \(C-Web Interface\)](#)
- [SRC Access to Directory Data on page 19](#)

Configuring Access to Subscriber Data (SRC CLI)

Use the following configuration statements to configure access to subscriber data:

```
shared sae configuration ldap subscriber-data {
  subscription-loading-filter (subscriberRefFilter | objectClassFilter);
  load-subscriber-schedules;
  login-cache-dn login-cache-dn ;
  session-cache-dn session-cache-dn ;
  server-address server-address ;
  dn dn ;
  authentication-dn authentication-dn ;
  password password ;
  directory-eventing;
  polling-interval polling-interval ;
  (ldaps);
}
```

To configure SAE access to subscriber data:

1. From configuration mode, access the configuration statement that configures SAE access to subscriber data in the directory. In this sample procedure, the subscriber data is configured in the `se-region` group.

```
user@host# edit shared sae group se-region configuration ldap subscriber-data
```

2. Select the filter that the SAE uses to search for subscriptions in the directory when the SAE loads a subscription to a subscriber reference filter.

```
[edit shared sae group se-region configuration ldap subscriber-data]
user@host# set subscription-loading-filter (subscriberRefFilter | objectClassFilter)
```

3. (Optional) Enable loading of subscriber schedules.

```
[edit shared sae group se-region configuration ldap subscriber-data]
user@host# set load-subscriber-schedules
```

4. Specify the subtree in the directory in which subscriber information is stored.

```
[edit shared sae group se-region configuration ldap subscriber-data]
user@host# set login-cache-dn login-cache-dn
```

5. Specify the subtree in the directory in which persistent session data is cached.

```
[edit shared sae group se-region configuration ldap subscriber-data]
user@host# set session-cache-dn session-cache-dn
```

6. (Optional) Specify the directory server that stores subscriber information.

```
[edit shared sae group se-region configuration ldap subscriber-data]
user@host# set server-address server-address
```

7. Specify the subtree in the directory where subscriber data is cached.

```
[edit shared sae group se-region configuration ldap subscriber-data]
user@host# set dn dn
```

8. (Optional) Specify the DN that the SAE uses to authenticate access to the directory server.

```
[edit shared sae group se-region configuration ldap subscriber-data]
user@host# set authentication-dn authentication-dn
```

9. (Optional) Specify the password used to authenticate access to the directory server.

```
[edit shared sae group se-region configuration ldap subscriber-data]
user@host# set password password
```

10. (Optional) Enable automatic discovery of changes in subscriber profiles.

```
[edit shared sae group se-region configuration ldap subscriber-data]
user@host# set directory-eventing
```

11. Set the frequency for checking the directory for updates.

```
[edit shared sae group se-region configuration ldap subscriber-data]
user@host# set polling-interval polling-interval
```

12. Enable LDAPS as the secure protocol for connections to the server that stores subscriber data.

```
[edit shared sae group se-region configuration ldap subscriber-data]
user@host# set ldaps
```

13. (Optional) Verify your configuration.

```
[edit shared sae group se-region configuration ldap subscriber-data]
user@host# show
subscription-loading-filter objectClassFilter;
load-subscriber-schedules;
login-cache-dn o=users,<base>;
session-cache-dn o=PersistentSessions,<base>;
server-address 127.0.0.1;
dn o=users,<base>;
authentication-dn cn=ssp,o=components,o=operators,<base>;
password *****;
directory-eventing;
```

```
polling-interval 30;
ldaps;
```

- Related Documentation**
- *Creating Grouped Configurations for the SAE (SRC CLI)*
 - [Configuring Access Through LDAPS to Service and Subscriber Data \(SRC CLI\) on page 20](#)
 - [Configuring Access to Service Data \(SRC CLI\) on page 23](#)
 - *Viewing General Information About Subscriber Sessions (SRC CLI)*
 - *Viewing Statistics for Subscriber and Service Sessions (SRC CLI)*

Configuring Access to Service Data (SRC CLI)

Use the following configuration statements to configure access to service data:

```
shared sae configuration ldap service-data {
  server-address server-address;
  dn dn;
  authentication-dn authentication-dn;
  password password;
  directory-eventing;
  polling-interval polling-interval;
  (ldaps);
}
```

To configure SAE access to service data:

1. From configuration mode, access the configuration statement that configures SAE access to service data in the directory. In this sample procedure, the service data is configured in the se-region group.

```
user@host# edit shared sae group se-region configuration ldap service-data
```

2. (Optional) Specify the directory server that stores service data.

```
[edit shared sae group se-region configuration ldap service-data]
user@host# set server-address server-address
```

3. Specify the subtree in the directory where service data is cached.

```
[edit shared sae group se-region configuration ldap service-data]
user@host# set dn dn
```

4. (Optional) Specify the DN that the SAE uses to authenticate access to the directory server.

```
[edit shared sae group se-region configuration ldap service-data]
user@host# set authentication-dn authentication-dn
```

5. (Optional) Specify the password used to authenticate access to the directory server.

```
[edit shared sae group se-region configuration ldap service-data]
user@host# set password password
```

6. (Optional) Enable or disable automatic discovery of changes to service data.

```
[edit shared sae group se-region configuration ldap service-data]
user@host# set directory-eventing
```

7. Set the frequency for checking the directory for updates.

```
[edit shared sae group se-region configuration ldap service-data]
user@host# set polling-interval polling-interval
```

8. Enable LDAPS as the secure protocol for connections to the server that stores service data.

```
edit shared sae group se-region configuration ldap service-data]
user@host# set ldaps
```

9. (Optional) Verify your configuration.

```
[edit shared sae group se-region configuration ldap service-data]
user@host# show
server-address 10.10.45.3;
dn <base>;
authentication-dn <base>;
password *****;
directory-eventing;
polling-interval 30;
ldaps;
```

Related Documentation

- [Creating Grouped Configurations for the SAE \(SRC CLI\)](#)
- [Configuring Access to Subscriber Data \(SRC CLI\) on page 21](#)
- [Configuring Access to Policy Data \(SRC CLI\) on page 24](#)
- [Configuring Access Through LDAPS to Service and Subscriber Data \(SRC CLI\) on page 20](#)

Configuring Access to Policy Data (SRC CLI)

Use the following configuration statements to configure access to policy data:

```
shared sae configuration ldap policy-data {
  policy-dn policy-dn ;
  parameter-dn parameter-dn ;
  directory-eventing;
  polling-interval polling-interval ;
}
```

To configure SAE access to subscriber data:

1. From configuration mode, access the configuration statement that configures SAE access to policy data in the directory. In this sample procedure, the policy data is configured in the se-region group.

```
user@host# edit shared sae group se-region configuration ldap policy-data
```

2. Specify the subtree in the directory in which policy data stored.

```
[edit shared sae group se-region configuration ldap policy-data]
user@host# set policy-dn policy-dn
```


3. Specify the subtree in the directory in which policy parameter data is cached.

```
[edit shared sae group se-region configuration ldap policy-data]
user@host# set parameter-dn parameter-dn
```

4. (Optional) Enable or disable automatic discovery of changes to policy data.

```
[edit shared sae group se-region configuration ldap policy-data]
user@host# set directory-eventing
```

5. Set the frequency for checking the directory for updates.

```
[edit shared sae group se-region configuration ldap policy-data]
user@host# set polling-interval polling-interval
```

6. (Optional) Verify your configuration.

```
[edit shared sae group se-region configuration ldap policy-data]
user@host# show
policy-dn o=Policy,<base>;
parameter-dn o=Parameters,<base>;
directory-eventing;
polling-interval 30;
```

Related Documentation

- [Creating Grouped Configurations for the SAE \(SRC CLI\)](#)
- [Configuring Access to Subscriber Data \(SRC CLI\) on page 21](#)
- [Configuring Access to Service Data \(SRC CLI\) on page 23](#)
- [Configuring Access to the Persistent Login Cache \(SRC CLI\) on page 25](#)
- [SRC Access to Directory Data on page 19](#)

Configuring Access to the Persistent Login Cache (SRC CLI)

Use the following configuration statements to configure access to persistent login cache data:

```
shared sae configuration ldap persistent-login-cache {
  server-address server-address;
  dn dn;
  authentication-dn authentication-dn;
  password password;
  directory-eventing;
  polling-interval polling-interval;
  (ldaps);
}
```

To configure SAE access to persistent login cache data:

1. From configuration mode, access the configuration statement that configures SAE access to persistent login cache data in the directory. In this sample procedure, the persistent login cache data is configured in the se-region group.

```
user@host# edit shared sae group se-region configuration ldap persistent-login-cache
```

2. (Optional) Specify the directory server that stores service data.

```
[edit shared sae group se-region configuration ldap persistent-login-cache]
user@host# set server-address server-address
```

3. Specify the subtree in the directory where persistent login cache data is cached.

```
[edit shared sae group se-region configuration ldap persistent-login-cache]
user@host# set dn dn
```

4. (Optional) Specify the DN that the SAE uses to authenticate access to the directory server.

```
[edit shared sae group se-region configuration ldap persistent-login-cache]
user@host# set authentication-dn authentication-dn
```

5. (Optional) Specify the password used to authenticate access to the directory server.

```
[edit shared sae group se-region configuration ldap persistent-login-cache]
user@host# set password password
```

6. (Optional) Enable automatic discovery of changes to persistent login cache data.

```
[edit shared sae group se-region configuration ldap persistent-login-cache]
user@host# set directory-eventing
```

7. Set the frequency for checking the directory for updates.

```
[edit shared sae group se-region configuration ldap persistent-login-cache]
user@host# set polling-interval polling-interval
```

8. Enable LDAPS as the secure protocol for connections to the server that stores persistent login cache data.

```
[edit shared sae group se-region configuration ldap persistent-login-cache]
user@host# set ldaps
```

9. (Optional) Verify your configuration.

```
[edit shared sae group se-region configuration ldap persistent-login-cache]
user@host# show
dn "o=authCache, <base>";
directory-eventing;
polling-interval 30;
ldaps;
```

Related Documentation

- *Creating Grouped Configurations for the SAE (SRC CLI)*
- [Configuring Access to Subscriber Data \(SRC CLI\) on page 21](#)
- [Configuring Access to Service Data \(SRC CLI\) on page 23](#)
- *Configuring Access to the Persistent Login Cache (C-Web Interface)*
- [SRC Access to Directory Data on page 19](#)

Configuring the Location of Network Device Data (SRC CLI)

Use the following configuration statement to configure access to network device data:

```
shared sae configuration ldap {
  network-dn network-dn;
}
```

To configure SAE access to network device data:

1. From configuration mode, access the configuration statement that configures SAE access to network device data in the directory. In this sample procedure, the network device data is configured in the se-region group.

```
user@host# edit shared sae group se-region configuration ldap
```

2. Specify the subtree in the directory where network device data is stored.

```
[edit shared sae group se-region configuration ldap]
user@host# set network-dn network-dn
```

3. Verify your configuration.

```
[edit shared sae group se-region configuration ldap]
user@host# show network-dn
network-dn o=Network,<base>;
```

Related Documentation

- [Creating Grouped Configurations for the SAE \(SRC CLI\)](#)
- [Enabling Automatic Discovery of Changes in SAE Configuration Data \(SRC CLI\) on page 27](#)
- [Configuring Access to the Persistent Login Cache \(SRC CLI\) on page 25](#)
- For more information about monitoring the SAE data with the SRC CLI, see *Viewing Information About the Directory Blacklist (SRC CLI)*

Enabling Automatic Discovery of Changes in SAE Configuration Data (SRC CLI)

Use the following configuration statement to enable automatic discovery of changes in SAE configuration data:

```
shared sae configuration ldap {
  enable-directory-eventing;
}
```

To enable automatic discovery of changes in SAE configuration data:

1. From configuration mode, access the configuration statement that enables automatic discovery of changes in SAE configuration data in the directory. In this sample procedure, automatic discovery is configured in the se-region group.

```
user@host# edit shared sae group se-region configuration ldap
```

2. Enable automatic discovery of changes to SAE configuration data.

```
[edit shared sae group se-region configuration ldap]
user@host# enable-directory-eventing
```

Related Documentation

- [Creating Grouped Configurations for the SAE \(SRC CLI\)](#)
- [Enabling Automatic Discovery of Changes in SAE Configuration Data \(C-Web Interface\)](#)
- [Setting the Timeout and Number of Events for SAE Directory Eventing \(SRC CLI\) on page 28](#)

Setting the Timeout and Number of Events for SAE Directory Eventing (SRC CLI)

Use the following configuration statements to set the directory eventing timeout and the number of simultaneous events that the SAE can receive from the directory:

```
shared sae configuration ldap directory-eventing {  
    timeout;  
    dispatcher-pool-size dispatcher-pool-size ;  
}
```

To configure the directory eventing timeout and the number of simultaneous events that the SAE can receive from the directory:

1. From configuration mode, access the configuration statement that configures SAE directory eventing. In this sample procedure, directory eventing is configured in the se-region group.

```
user@host# edit shared sae group se-region configuration ldap directory-eventing
```

2. Specify the maximum time that the directory eventing system waits for the directory to respond.

```
[edit shared sae group se-region configuration ldap directory-eventing]  
user@host# set timeout timeout
```

3. Specify the number of events that the SAE can receive from the directory simultaneously.

```
[edit shared sae group se-region configuration ldap directory-eventing]  
user@host# set dispatcher-pool-size dispatcher-pool-size
```

4. (Optional) Verify your configuration.

```
[edit shared sae group se-region configuration ldap directory-eventing]  
user@host# show  
timeout 60;  
dispatcher-pool-size 1000;
```

Related Documentation

- [Creating Grouped Configurations for the SAE \(SRC CLI\)](#)
- [Setting the Timeout and Number of Events for SAE Directory Eventing \(C-Web Interface\)](#)
- [Enabling Automatic Discovery of Changes in SAE Configuration Data \(SRC CLI\) on page 27](#)

CHAPTER 3

Managing Subscriber and Service Session Data (SRC CLI)

- [Storing Subscriber and Service Session Data on page 29](#)
- [Configuring the Session Store Feature \(SRC CLI\) on page 31](#)
- [Configuring the Number of Threads for Sessions \(SRC CLI\) on page 35](#)

Storing Subscriber and Service Session Data

To aid in recovering from an SAE failover, the SAE stores subscriber and service session data in flat files on the SAE host. The SRC component that controls the storage of session data on the SAE is called the session store. The session store queues data and then writes the data to session store files on the SAE host's disk. After the data has been written to disk, it can survive a server reboot.

You can configure how the SAE stores session data for JunosE routers, devices running Junos OS, simulated routers, and *PacketCable Multimedia Specification* (PCMM) devices.

Session Store Files

Session store files are numbered flat files. Session store files are located in a directory on the SAE host. You can configure the size of session store files. After the maximum size has been reached, the session store creates a new file and begins writing data to the new file.

Store operations, such as adding a session to the store (put store operations) or removing a session from the store (remove store operations), are queued in a buffer before they are written to the session store file. You can configure parameters that determine when the session store writes a queue to a session store file.

Session store files are deleted if they have not been modified and if no session activity has taken place for one week. All the data files that contain the sessions associated with a particular virtual router are deleted at the same time.

Active and Passive Session Stores

You can have a community of SAEs and duplicate session store data on each SAE in the community in case of an SAE failover. SAE communities are made up of SAEs that you configure as connected SAEs for a virtual router object.

SAEs in a community are given the role of either active SAE or passive SAE. The active SAE keeps session data up to date within the community. Each active session store opens a Transmission Control Protocol (TCP) connection to its passive SAE. The TCP connection triggers the creation of a passive session store in that SAE. When the active session store writes operations to the session store file, it passes them to passive session stores on all SAEs in the community.

When you modify a community, wait for passive session stores on the new community members to be updated before you shut down the currently active SAE. Otherwise, if you add a new member to a community, and then a failover from the current active SAE to the new member is triggered immediately, the new member's session store may not have received all data from the active SAE's session store.

Standby SAEs

You can configure standby SAEs for a configuration that include JunosE routers. In a community of SAEs, a standby SAE can provide redundancy for the active SAE. The active SAE connects to the standby SAE through a COPS-PR connection on port 3228. The active SAE maintains a separate session store connection with the standby SAE through port 8820 (default).

The active SAE replicates state and session data, including COPS messages received from JunosE routers, to the standby SAE. This replication reduces the failover time from one SAE to another. The active SAE detects a connection failure when a subsequent COPS message needs to be replicated because it has to wait for the standby to respond to the replication message. Both the active and standby SAEs detect a connection failure when the keep-alive timeout occurs (1 second).



NOTE: We recommend that you use a highly reliable and available connection between an active SAE and a standby SAE to ensure availability of the two SAEs.

For standby SAEs, you configure an SAE community and the session store at the same time by configuring SAE identifiers for in the configuration for the shared network device virtual router. In the configuration, an exclamation point identifies standby SAEs.

Session Store File Rotation

The session store periodically rotates the session store files. During rotation, the session store copies put store operations for live sessions from the oldest file to the end of the newest file. (Live sessions are sessions that have been created but not yet deleted.) It then deletes the oldest file. Sessions are rotated in batches, and you can configure the number of sessions that are rotated at the same time, and how much disk space is used by live sessions before files are rotated. No session store activity can take place while a batch of sessions is rotated.

Related Documentation

- [COPS Connection Between JunosE Routers and the SAE on page 53](#)
- [Configuring the Session Store Feature \(SRC CLI\) on page 31](#)

- [Adding JunosE Routers and Virtual Routers \(SRC CLI\) on page 54](#)
- [SRC Data Storage](#)
- [Configuring the Number of Threads for Sessions \(SRC CLI\) on page 35](#)
- [Viewing Statistics for Subscriber and Service Sessions \(SRC CLI\)](#)
- [Viewing Information About Subscriber Sessions by Session ID \(C-Web Interface\)](#)

Configuring the Session Store Feature (SRC CLI)

You can configure three things for the session store feature:

1. [Configuring Session Store Parameters for a Device Driver on page 31](#)
2. [Configuring Global Session Store Parameters on page 33](#)
3. [Reducing the Size of Objects for the Session Store Feature on page 35](#)

Configuring Session Store Parameters for a Device Driver

Use the following configuration statements to configure session store parameters within a device driver configuration:

```
shared sae configuration driver ( aaa | junos | junos-dmi | junose | junos-ptsp | pcmm |
simulated | third-party ) session-store {
  maximum-queue-age maximum-queue-age;
  maximum-queued-operations maximum-queued-operations;
  maximum-queue-size maximum-queue-size;
  maximum-file-size maximum-file-size;
  minimum-disk-space-usage minimum-disk-space-usage;
  rotation-batch-size rotation-batch-size;
  maximum-session-size maximum-session-size;
  disk-load-buffer-size disk-load-buffer-size;
  network-buffer-size network-buffer-size;
  retry-interval retry-interval;
  communications-timeout communications-timeout;
  load-timeout load-timeout;
  idle-timeout idle-timeout;
  maximum-backlog-ratio maximum-backlog-ratio;
  minimum-backlog minimum-backlog;
}
```

To configure session store parameters within a device driver configuration:

1. From configuration mode, access the configuration statement that configures the session store for your device driver. In this sample procedure, the session store for a Junos device driver is configured in the se-region group.

```
user@host# edit shared sae group se-region configuration driver junos session-store
```

2. (Optional) Specify the maximum age that a queue of buffered store operations (such as adding a session to the store or removing a session from the store) can reach before the queue is written to a session store file.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set maximum-queue-age maximum-queue-age
```

3. (Optional) Specify the number of buffered store operations that are queued before the queue is written to a session store file.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set maximum-queued-operations maximum-queued-operations
```

4. (Optional) Specify the maximum size that a queue of buffered store operations can reach before the queue is written to a session store file.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set maximum-queue-size maximum-queue-size
```

5. (Optional) Specify the maximum size of session store files.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set maximum-file-size maximum-file-size
```

6. (Optional) Specify the percentage of space in all session store files that is used by live sessions.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set minimum-disk-space-usage minimum-disk-space-usage
```

7. (Optional) Specify the number of sessions that are rotated from the oldest file to the newest file at the same time that the oldest session store file is rotated.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set rotation-batch-size rotation-batch-size
```

8. (Optional) Specify the maximum size of a single subscriber or service session.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set maximum-session-size maximum-session-size
```

9. (Optional) Specify the size of the buffer that is used to load all of a session store's files from disk at startup.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set disk-load-buffer-size disk-load-buffer-size
```

10. (Optional) Specify the size of the buffer that holds messages or message segments that are waiting to be sent to passive session stores.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set network-buffer-size network-buffer-size
```

11. (Optional) Specify the time interval between attempts by the active session store to connect to missing passive session stores.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set retry-interval retry-interval
```


12. (Optional) Specify the amount of time that a session store waits before closing when it is blocked from reading or writing a message.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set communications-timeout communications-timeout
```

13. (Optional) Specify the time that an active session store waits for a passive session store or a passive session store waits for an active session store to load its data from disk before it closes the connection to the session store.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set load-timeout load-timeout
```

14. (Optional) Specify the time that a passive session store waits for activity from the active session store before it closes the connection to the active session store.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set idle-timeout idle-timeout
```

15. (Optional) Specify when the active session store closes the connection to a passive session store because of a backlog of messages waiting to be sent.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set maximum-backlog-ratio maximum-backlog-ratio
```

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# set minimum-backlog minimum-backlog
```

16. (Optional) Verify your configuration.

```
[edit shared sae group se-region configuration driver junos session-store]
user@host# show
maximum-queue-age 5000;
maximum-queued-operations 50;
maximum-queue-size 51050;
maximum-file-size 25000000;
minimum-disk-space-usage 25;
rotation-batch-size 50;
maximum-session-size 10000;
disk-load-buffer-size 1000000;
network-buffer-size 51050;
retry-interval 5000;
communications-timeout 60000;
load-timeout 420000;
idle-timeout 3600000;
maximum-backlog-ratio 1.5;
minimum-backlog 5000000;
```

Configuring Global Session Store Parameters

This topic describes how to configure global session store parameters that are shared by all session store instances (active or passive) on the SAE. You can also configure session store parameters within a device driver configuration.

Use the following configuration statements to configure global session store parameters.

```
shared sae configuration driver session-store {  
  file-expiry-age file-expiry-age;  
  file-expiry-check-interval file-expiry-check-interval;  
  ip-address ip-address;  
  port port;  
  root-directory root-directory;  
}
```

To configure global session store parameters:

1. From configuration mode, access the configuration statement that configures the global session store parameters. In this sample procedure, the global session store is configured in the se-region group.

```
user@host# edit shared sae group se-region configuration driver session-store
```

2. (Optional) Specify the expiry age of the dormant session store file. The service activation engine (SAE) deletes the session store file if it is not modified for the configured period. The value range is 0–2,147,483,647 milliseconds. By default, the value is set to 604,800,000 milliseconds—that is, 7 days.

```
[edit shared sae group se-region configuration driver session-store]  
user@host# set file-expiry-age file-expiry-age
```

3. (Optional) Specify the frequency at which the server monitors the age of the session store file. The value range is 1000–2,147,483,647 milliseconds. By default, the value is set to 3,600,000 milliseconds—that is, 1 hour.

```
[edit shared sae group se-region configuration driver session-store]  
user@host# set file-expiry-check-interval file-expiry-check-interval
```

4. (Optional) Specify the IP address or hostname that the session store infrastructure on this SAE uses to listen for incoming TCP connections from active session stores.

```
[edit shared sae group se-region configuration driver session-store]  
user@host# set ip-address ip-address
```

5. (Optional) Specify the TCP port number on which the session store infrastructure on this SAE listens for incoming connections from active session stores.

```
[edit shared sae group se-region configuration driver session-store]  
user@host# set port port
```

6. (Optional) Specify the root directory in which the session store creates files.

```
[edit shared sae group se-region configuration driver session-store]  
user@host# set root-directory root-directory
```

7. (Optional) Verify your configuration.

```
[edit shared sae group se-region configuration driver session-store]  
user@host# show  
file-expiry-age 604800000;  
file-expiry-check-interval 3600000;  
ip-address 10.10.70.0;
```

```
port 8820;
root-directory var/sessionStore;
```



NOTE: For the `set file-expiry-age` and `set file-expiry-age-check-interval` parameters, we recommend that you configure a value that is nearer to the default value. You must restart the SAE for the configuration changes to take effect.

Reducing the Size of Objects for the Session Store Feature

You can use serialized data compression to reduce the size of sessions objects that the SAE sends across the network for the session store feature. Enabling this property reduces the size of objects, but increases the CPU load on the SAE.

Use the following configuration statement to specify whether or not session objects are compressed.

```
shared sae configuration {
  compress-session-data;
}
```

To specify whether or not session objects are compressed:

1. From configuration mode, access the sae configuration. In this sample procedure, data compression is configured in the se-region group.

```
user@host# edit shared sae group se-region configuration
```

2. Enable reducing the size of session objects (subscriber and service sessions) that the SAE sends across the network for the session store feature.

```
[edit shared sae group se-region configuration]
user@host# set compress-session-data
```

3. (Optional) Verify your configuration.

```
[edit shared sae group se-region configuration]
user@host# show compress-session-data
compress-session-data;
```

Configuring the Number of Threads for Sessions (SRC CLI)

Use the following configuration statement to set the number of threads used for session-related activity.

```
shared sae configuration session-job-manager {
  number-of-threads number-of-threads;
}
```

To configure the number of threads used to handle session-related activity:

1. From configuration mode, access the session job manager configuration. In this sample procedure, the number of threads is configured in the se-region group.

```
user@host# edit shared sae group se-region configuration session-job-manager
```

2. Specify the number of threads used for session-related activity.

```
[edit shared sae group se-region configuration session-job-manager]  
user@host# set number-of-threads number-of-threads
```

3. (Optional) Verify your configuration.

```
[edit shared sae group se-region configuration session-job-manager]  
user@host# show  
number-of-threads 10;
```

- Related Documentation**
- [Configuring the Session Store Feature \(SRC CLI\) on page 31](#)
 - [Storing Subscriber and Service Session Data on page 29](#)

CHAPTER 4

Managing SAE Data (SRC CLI)

- [Commands to Manage SAE Data on page 37](#)
- [Reloading the SAE Data \(SRC CLI\) on page 38](#)
- [Reloading the SAE Configuration \(SRC CLI\) on page 38](#)
- [Reloading Services \(SRC CLI\) on page 39](#)
- [Reloading Subscriptions \(SRC CLI\) on page 39](#)
- [Reloading Interface Classification Scripts \(SRC CLI\) on page 39](#)
- [Reloading Domain Maps \(SRC CLI\) on page 40](#)
- [Removing the Directory Blacklist \(SRC CLI\) on page 40](#)
- [Removing Login Registrations \(SRC CLI\) on page 40](#)
- [Removing Equipment Registrations \(SRC CLI\) on page 41](#)
- [Modifying Failover Server Parameters \(SRC CLI\) on page 41](#)
- [Shutting Down the Device Drivers \(SRC CLI\) on page 42](#)

Commands to Manage SAE Data

You can use the following operational mode commands to manage SAE data:

- **clear sae directory-blacklist**
- **clear sae registered equipment**
- **clear sae registered login**
- **request sae load configuration**
- **request sae load domain-map**
- **request sae load interface-classification**
- **request sae load services**
- **request sae load subscriptions**
- **request sae modify device failover**
- **request sae shutdown device**
- **show sae directory-blacklist**

- **show sae drivers**
- **show sae registered equipment**
- **show sae registered login**

For detailed information about each command, see the *SRC CLI Command Reference*.

**Related
Documentation**

- [Reloading the SAE Data \(SRC CLI\) on page 38](#)
- [Reloading the SAE Configuration \(SRC CLI\) on page 38](#)

Reloading the SAE Data (SRC CLI)

You can reload specified configuration components. You can reload the SAE server's current configuration for:

- SAE configuration
- Services
- Subscriptions
- Interface classifiers
- Domain map

**Related
Documentation**

- [Viewing Information About SAE Interfaces \(SRC CLI\)](#)
- [Viewing Information About SAE Device Drivers \(SRC CLI\)](#)
- [Viewing Information About Services \(SRC CLI\)](#)
- [Viewing Information About Policies on the SAE \(SRC CLI\)](#)

Reloading the SAE Configuration (SRC CLI)

To reload the SAE configuration data from the directory:

```
user@host> request sae load configuration
```

The new configuration takes effect immediately.

**Related
Documentation**

- [Initially Configuring the SAE](#)
- [Reloading the SAE Data \(SRC CLI\) on page 38](#)
- For more information about monitoring the SAE data with the SRC CLI, see [Viewing Information About the Directory Blacklist \(SRC CLI\)](#)
- [Reloading Services \(SRC CLI\) on page 39](#)

Reloading Services (SRC CLI)

To reload the services, scopes, virtual routers, policies, service mutex groups, and service schedules from the directory:

```
user@host> request sae load services
```

Related service sessions are activated, deactivated, or reactivated as needed.

Related Documentation

- [Reloading the SAE Configuration \(SRC CLI\) on page 38](#)
- [Viewing Information About Services \(SRC CLI\)](#)
- [Viewing Information About Services \(C-Web Interface\)](#)
- [Commands to Manage SAE Data on page 37](#)
- [Reloading the SAE Data \(SRC CLI\) on page 38](#)
- [Reloading Subscriptions \(SRC CLI\) on page 39](#)

Reloading Subscriptions (SRC CLI)

To reload all subscriptions from the directory:

```
user@host> request sae load subscriptions
```

Related service sessions are activated, deactivated, or reactivated as needed.

Related Documentation

- [Reloading the SAE Configuration \(SRC CLI\) on page 38](#)
- [Viewing Statistics for Subscriber and Service Sessions \(SRC CLI\)](#)
- For more information about viewing subscriber sessions with the SRC CLI, see [Viewing General Information About Subscriber Sessions \(SRC CLI\)](#)
- [Reloading the SAE Data \(SRC CLI\) on page 38](#)

Reloading Interface Classification Scripts (SRC CLI)

To reload the interface classification scripts from the directory, and apply the result of the interface classification changes to the router:

```
user@host> request sae load interface-classification
```

Related Documentation

- [Viewing Information About SAE Interfaces \(SRC CLI\)](#)
- [Reloading the SAE Data \(SRC CLI\) on page 38](#)
- [Commands to Manage SAE Data on page 37](#)

Reloading Domain Maps (SRC CLI)

To reload the mapping of domain names to retailer entries:

```
user@host> request sae load domain-map
```

This mapping is made available to the SAE's subscriber classification script.

Related Documentation

- [Reloading the SAE Data \(C-Web Interface\) on page 45](#)
- [Reloading Subscriptions \(SRC CLI\) on page 39](#)
- [Commands to Manage SAE Data on page 37](#)

Removing the Directory Blacklist (SRC CLI)

To remove the directory blacklist:

1. Issue the **show sae directory-blacklist** command to view information about the directory blacklist.
2. Issue the **clear sae directory-blacklist command** to remove the directory blacklist.

Related Documentation

- [Removing the Directory Blacklist \(C-Web Interface\) on page 47](#)
- [Removing Login Registrations \(SRC CLI\) on page 40](#)
- [Viewing Information About the Directory Blacklist \(SRC CLI\)](#)
- [Commands to Manage SAE Data on page 37](#)
- [Reloading the SAE Data \(SRC CLI\) on page 38](#)

Removing Login Registrations (SRC CLI)

You can delete all login registrations, or you can delete a specific registration.

To remove login registrations:

1. Issue the **show sae registered login** command to view the login registrations.
2. Issue the **clear sae registered login command** to remove all login registrations.

If you do not want to remove all login registrations, you can specify a single registration. You can also specify whether or not you want a confirmation before the registrations are deleted.

- To remove a specific registration, use the **mac-address** option and specify the media access control (MAC) address for the registration.

```
user@host> clear sae registered login mac-address mac-address
```

- To specify that no confirmation is requested before the software deletes the registration entries, use the **force** option.


```
user@host> clear sae registered login force
user@host> clear sae registered login mac-address mac-address force
```

**Related
Documentation**

- [Removing Login Registrations \(C-Web Interface\) on page 48](#)
- [Removing Equipment Registrations \(SRC CLI\) on page 41](#)
- [Viewing Login Registrations \(SRC CLI\)](#)
- [Viewing Login Registrations \(C-Web Interface\)](#)
- [Reloading the SAE Data \(SRC CLI\) on page 38](#)

Removing Equipment Registrations (SRC CLI)

You can delete all equipment registrations, or you can delete a specific registration. The demonstration residential portal provides an example of how to use equipment registration.

To remove equipment registrations:

1. Issue the **show sae registered equipment** command to view the equipment registrations.
 2. Issue the **clear sae registered equipment command** to remove all equipment registrations.
- To remove a specific registration, use the **mac-address** option and specify the media access control (MAC) address for the registration.

```
user@host> clear sae registered equipment mac-address mac-address
```

- To specify that no confirmation is requested before the software deletes the registration entries, use the **force** option.

```
user@host> clear sae registered equipment force
user@host> clear sae registered equipment mac-address mac-address force
```

**Related
Documentation**

- [Removing Equipment Registrations \(C-Web Interface\) on page 48](#)
- [Removing Login Registrations \(SRC CLI\) on page 40](#)
- [Viewing Equipment Registrations \(SRC CLI\)](#)
- [Viewing Equipment Registrations \(C-Web Interface\)](#)
- [Reloading the SAE Data \(SRC CLI\) on page 38](#)

Modifying Failover Server Parameters (SRC CLI)

To modify failover server parameters:

1. Issue the **show sae drivers brief** command to view the router or device instances.
2. Issue the **request sae modify device failover virtual-router-name *virtual-router-name* command** to modify failover server parameters.

- (Optional) To modify the IP address of an alternate SAE server to which a router can reconnect when this driver closes its connection, use the **ip-address** option. This option is not applicable to the PCMM device driver.

```
user@host> request sae modify device failover virtual-router-name  
virtual-router-name ip-address ip-address
```

- (Optional) To modify the port of an alternate SAE server to which a router can reconnect when this driver closes its connection, use the **tcp-port** option. This option is not applicable to the PCMM device driver.

```
user@host> request sae modify device failover virtual-router-name  
virtual-router-name tcp-port tcp-port
```

- (Optional) To specify whether the device driver sends its own failover IP address and port to the router when it closes its connection, use the **use-failover-server** option. This option is not applicable to the PCMM device driver.

```
user@host> request sae modify device failover virtual-router-name  
virtual-router-name use-failover-server
```

- (Optional) To specify that no confirmation is requested before the software modifies the parameters, use the **force** option.

```
user@host> request sae modify device failover virtual-router-name  
virtual-router-name force  
user@host> request sae modify device failover virtual-router-name  
virtual-router-name ip-address ip-address force  
user@host> request sae modify device failover virtual-router-name  
virtual-router-name tcp-port tcp-port force  
user@host> request sae modify device failover virtual-router-name  
virtual-router-name use-failover-server force
```

Related Documentation

- [Modifying Failover Server Parameters \(C-Web Interface\) on page 49](#)
- [Viewing Statistics for Device Drivers \(SRC CLI\)](#)
- [Viewing Statistics for Specific Device Drivers \(SRC CLI\)](#)
- [Viewing Information About Device Drivers \(C-Web Interface\)](#)

Shutting Down the Device Drivers (SRC CLI)

To shut down the specified router or device instance:

1. Issue the **show sae drivers brief** command to view the router or device instances.
2. Issue the **request sae shutdown device** command to shut down all device drivers.
 - To shut down specific drivers managing a virtual router, use the **filter** option and specify all or part of the name of the virtual router.

```
user@host> request sae shutdown device filter filter
```

- To specify that no confirmation is requested before the software shuts down the device drivers, use the **force** option.

```
user@host> request sae shutdown device force
```

```
user@host> request sae shutdown device filter filter force
```

- Related Documentation**
- [Shutting Down the Device Drivers \(C-Web Interface\) on page 49](#)
 - *Viewing Statistics for Device Drivers (SRC CLI)*

CHAPTER 5

Managing SAE Data (C-Web Interface)

- [Reloading the SAE Data \(C-Web Interface\) on page 45](#)
- [Reloading the SAE Configuration \(C-Web Interface\) on page 46](#)
- [Reloading Services \(C-Web Interface\) on page 46](#)
- [Reloading Subscriptions \(C-Web Interface\) on page 46](#)
- [Reloading Interface Classification Scripts \(C-Web Interface\) on page 47](#)
- [Reloading Domain Maps \(C-Web Interface\) on page 47](#)
- [Removing the Directory Blacklist \(C-Web Interface\) on page 47](#)
- [Removing Login Registrations \(C-Web Interface\) on page 48](#)
- [Removing Equipment Registrations \(C-Web Interface\) on page 48](#)
- [Modifying Failover Server Parameters \(C-Web Interface\) on page 49](#)
- [Shutting Down the Device Drivers \(C-Web Interface\) on page 49](#)

Reloading the SAE Data (C-Web Interface)

You can reload specified configuration components. You can reload the SAE server's current configuration for:

- SAE configuration
- Services
- Subscriptions
- Interface classifiers
- Domain map

Related Documentation

- [Reloading the SAE Data \(SRC CLI\) on page 38](#)
- [Viewing Information About Interfaces \(C-Web Interface\)](#)
- [Viewing Information About Device Drivers \(C-Web Interface\)](#)
- [Viewing Information About Services \(C-Web Interface\)](#)
- [Viewing Information About Policies \(C-Web Interface\)](#)

Reloading the SAE Configuration (C-Web Interface)

To reload the SAE configuration data from the directory:

1. Click **Manage>Request>SAE>Load>Configuration**.

The Configuration pane appears.

2. Enter information as described in the Help text in the main pane, and click **OK**.

The new configuration takes effect immediately.

Related Documentation

- *Initially Configuring the SAE (C-Web Interface)*
- [Reloading the SAE Data \(C-Web Interface\) on page 45](#)
- For more information about monitoring the SAE data with the C-Web interface, see *Viewing Information About the Directory Blacklist (C-Web Interface)*
- [Reloading Services \(C-Web Interface\) on page 46](#)

Reloading Services (C-Web Interface)

To reload the services, scopes, virtual routers, policies, service mutex groups, and service schedules from the directory:

1. Click **Manage>Request>SAE>Load>Services**.

The Services pane appears.

2. Enter information as described in the Help text in the main pane, and click **OK**.

Related service sessions are activated, deactivated, or reactivated as needed.

Related Documentation

- [Reloading the SAE Configuration \(C-Web Interface\) on page 46](#)
- *Viewing Information About Services (C-Web Interface)*
- [Commands to Manage SAE Data on page 37](#)
- [Reloading the SAE Data \(C-Web Interface\) on page 45](#)
- [Reloading Subscriptions \(C-Web Interface\) on page 46](#)

Reloading Subscriptions (C-Web Interface)

To reload all subscriptions from the directory:

1. Click **Manage>Request>SAE>Load>Subscriptions**.

The Subscriptions pane appears.

2. Enter information as described in the Help text in the main pane, and click **OK**.

Related service sessions are activated, deactivated, or reactivated as needed.

- Related Documentation**
- [Reloading the SAE Configuration \(C-Web Interface\) on page 46](#)
 - [Viewing Statistics for Subscriber and Service Sessions \(SRC CLI\)](#)
 - For more information about viewing subscriber sessions with the SRC CLI, see [Viewing General Information About Subscriber Sessions \(SRC CLI\)](#)
 - [Reloading the SAE Data \(C-Web Interface\) on page 45](#)

Reloading Interface Classification Scripts (C-Web Interface)

To reload the interface classification scripts from the directory, and apply the result of the interface classification changes to the router:

1. Click **Manage>Request>SAE>Load>Interface Classification**.
The Interface Classification pane appears.
2. Enter information as described in the Help text in the main pane, and click **OK**.

- Related Documentation**
- [Viewing Information About SAE Interfaces \(SRC CLI\)](#)
 - [Reloading the SAE Data \(C-Web Interface\) on page 45](#)
 - [Commands to Manage SAE Data on page 37](#)

Reloading Domain Maps (C-Web Interface)

To reload the mapping of domain names to retailer entries:

1. Click **Manage>Request>SAE>Load>Domain Map**.
The Domain Map pane appears.
2. Enter information as described in the Help text in the main pane, and click **OK**.

This mapping is made available to the SAE's subscriber classification script.

- Related Documentation**
- [Reloading the SAE Data \(SRC CLI\) on page 38](#)
 - [Viewing Information About Interfaces \(C-Web Interface\)](#)
 - [Viewing Information About Device Drivers \(C-Web Interface\)](#)
 - [Viewing Information About Services \(C-Web Interface\)](#)
 - [Viewing Information About Policies \(C-Web Interface\)](#)

Removing the Directory Blacklist (C-Web Interface)

To remove the directory blacklist:

1. To view information about the directory blacklist:
 - a. Click **Monitor>SAE>Directory Blacklist**.

The Directory Blacklist pane appears.

- b. Enter information as described in the Help text in the main pane, and click **OK**.

2. To remove the directory blacklist:

- a. Click **Manage>Clear>SAE>Directory Blacklist**.

The Directory Blacklist pane appears.

- b. Enter information as described in the Help text in the main pane, and click **OK**.

**Related
Documentation**

- [Removing the Directory Blacklist \(SRC CLI\) on page 40](#)
- [Removing Login Registrations \(C-Web Interface\) on page 48](#)
- [Viewing Information About the Directory Blacklist \(C-Web Interface\)](#)
- [Reloading the SAE Data \(C-Web Interface\) on page 45](#)

Removing Login Registrations (C-Web Interface)

You can delete all login registrations, or you can delete a specific registration.

To remove login registrations:

1. Click **Monitor>SAE>Registered>Login**.

The Login pane appears.

2. Enter information as described in the Help text in the main pane, and click **OK**.

To remove login registrations:

1. Click **Manage>Clear>SAE>Registered>Login**.

The Login pane appears.

2. Enter information as described in the Help text in the main pane, and click **OK**.

**Related
Documentation**

- [Removing Login Registrations \(SRC CLI\) on page 40](#)
- [Removing the Directory Blacklist \(C-Web Interface\) on page 47](#)
- [Viewing Login Registrations \(C-Web Interface\)](#)
- [Reloading the SAE Data \(C-Web Interface\) on page 45](#)

Removing Equipment Registrations (C-Web Interface)

You can delete all equipment registrations, or you can delete a specific registration. The demonstration residential portal provides an example of how to use equipment registration.

To remove equipment registrations:

1. Click **Monitor>SAE>Registered>Equipment**.

The Equipment pane appears.

2. Enter information as described in the Help text in the main pane, and click **OK**.

To remove login registrations:

1. Click **Manage>Clear>SAE>Registered>Equipment**.

The Equipment pane appears.

2. Enter information as described in the Help text in the main pane, and click **OK**.

**Related
Documentation**

- [Removing Equipment Registrations \(SRC CLI\) on page 41](#)
- [Removing Login Registrations \(C-Web Interface\) on page 48](#)
- [Viewing Equipment Registrations \(C-Web Interface\)](#)
- [Reloading the SAE Data \(C-Web Interface\) on page 45](#)

Modifying Failover Server Parameters (C-Web Interface)

To modify failover server parameters:

1. To view the router or device instances:

- a. Click **Monitor>SAE>Drivers**.

The Drivers pane appears.

- b. Enter information as described in the Help text in the main pane, and click **OK**.

2. To modify failover server parameters:

- a. Click **Manage>SAE>Request>Modify>Device>Failover**.

The Failover pane appears.

- b. Enter information as described in the Help text in the main pane, and click **OK**.

**Related
Documentation**

- [Modifying Failover Server Parameters \(SRC CLI\) on page 41](#)
- [Viewing Information About Device Drivers \(C-Web Interface\)](#)
- [Viewing Statistics for Device Drivers \(SRC CLI\)](#)
- [Viewing Statistics for Specific Device Drivers \(SRC CLI\)](#)

Shutting Down the Device Drivers (C-Web Interface)

To shut down the specified router or device instance:

1. To view the router or device instances:

- a. Click **Monitor>SAE>Drivers**.

The Drivers pane appears.

- b. Enter information as described in the Help text in the main pane, and click **OK**.

2. To shut down all device drivers:

- a. Click **Manage>SAE>Request>Shutdown>Device**.

The Device pane appears.

- b. Enter information as described in the Help text in the main pane, and click **OK**.

**Related
Documentation**

- [Shutting Down the Device Drivers \(SRC CLI\) on page 42](#)
- *Viewing Information About Device Drivers (C-Web Interface)*

PART 2

Using Juniper Networks Routers in the SRC Network

- [Using JunosE Routers in the SRC Network \(SRC CLI\) on page 53](#)
- [Using Devices Running Junos OS in the SRC Network \(SRC CLI\) on page 77](#)
- [Managing Junos DMI Devices Using the SRC Software on page 105](#)

CHAPTER 6

Using JunosE Routers in the SRC Network (SRC CLI)

- [COPS Connection Between JunosE Routers and the SAE on page 53](#)
- [Adding JunosE Routers and Virtual Routers \(SRC CLI\) on page 54](#)
- [Configuring the SAE to Manage JunosE Routers \(SRC CLI\) on page 59](#)
- [How SNMP Obtains Information from Routers for the SRC Software on page 62](#)
- [Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 63](#)
- [Specifying JunosE Router Initialization Scripts on the SAE \(SRC CLI\) on page 66](#)
- [Updating Local IP Address Pools for JunosE Virtual Routers \(SRC CLI\) on page 67](#)
- [Updating Quality of Service Profiles for JunosE Virtual Routers \(SRC CLI\) on page 68](#)
- [Accessing the Router CLI on page 69](#)
- [Starting the SRC Client on a JunosE Router on page 70](#)
- [Stopping the SRC Client on a JunosE Router on page 70](#)
- [Monitoring Interactions Between the SAE and the Router Running JunosE Software on page 71](#)
- [Troubleshooting Problems with Managing JunosE Routers on page 71](#)
- [Viewing the State of JunosE Device Drivers \(SRC CLI\) on page 72](#)
- [Viewing Statistics for Specific JunosE Device Drivers \(SRC CLI\) on page 74](#)
- [Viewing Statistics for All JunosE Device Drivers \(SRC CLI\) on page 74](#)
- [Viewing the State of JunosE Device Drivers \(C-Web Interface\) on page 75](#)
- [Viewing Statistics for All JunosE Device Drivers \(C-Web Interface\) on page 75](#)

COPS Connection Between JunosE Routers and the SAE

Configuring the SRC client on a JunosE router opens a Common Open Policy Service (COPS) protocol layer connection to the SAE. When the SRC client software establishes a TCP/IP connection to the SAE, the SAE starts to manage the JunosE router. Subsequently, the SRC client sends configuration changes made on the JunosE router to the SAE, and the SAE updates SRC configurations for services and policies accordingly.

The SAE supports two versions of COPS:

- [COPS usage for policy provisioning \(COPS-PR\)](#)
- [COPS External Data Representation Standard \(COPS-XDR\)](#)

The version of COPS that you use depends on the version of COPS that your JunosE router supports. When you set up your JunosE router to work with the SAE, you enable either COPS-PR mode or COPS-XDR mode.

Highly Available Connections to JunosE Routers

JunosE routers maintain state information, a feature that allows an active, managing SAE to reconnect to a JunosE router without performing a data resynchronization in the following instances:

- The network connection between the SAE and the JunosE router is disrupted, and the router reconnects to the SAE
- For JunosE routers with high availability configured, when the secondary SRP takes control from a failed SRP it can reconnect to the SAE

To maintain highly available connections to JunosE routers, configure an SAE community and the session store by configuring SAE identifiers in the configuration for the shared network device virtual router. In the configuration, an exclamation point identifies standby SAEs.

Related Documentation

- [Storing Subscriber and Service Session Data on page 29](#)
- [Adding JunosE Routers and Virtual Routers \(SRC CLI\) on page 54](#)
- [Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 63](#)
- [How SNMP Obtains Information from Routers for the SRC Software on page 62](#)
- [Configuring the SAE to Manage JunosE Routers \(SRC CLI\) on page 59](#)
- [Starting the SRC Client on a JunosE Router on page 70](#)
- [Monitoring Interactions Between the SAE and the Router Running JunosE Software on page 71](#)

Adding JunosE Routers and Virtual Routers (SRC CLI)

The SAE uses router and virtual router objects to manage interfaces on JunosE virtual routers. Each JunosE router in the SRC network and its virtual routers (VRs) must have a configuration.

There are three ways to add routers:

1. [Adding Operative JunosE Routers and Virtual Routers on page 55](#)
2. [Adding Routers Individually \(SRC CLI\) on page 55](#)
3. [Adding Virtual Routers Individually \(SRC CLI\) on page 57](#)

Adding Operative JunosE Routers and Virtual Routers

To add routers and JunosE VRs that are currently operative and have an operating SNMP agent:

- In operational mode, enter the following command:

```
user@host> request network discovery network network <community community>
```

where:

- **network** —Address (with or without mask) of the network to discover
- **community** —Name of the SNMP community to which the devices belong

If you add a router using the discover network feature, the software adds the IP address of the first SNMP agent on the router to respond to the discover request.

After you add routers and JunosE VRs through network discovery, configure the virtual router's managing SAE address.

Adding Routers Individually (SRC CLI)

Use the following configuration statements to add a router:

```
shared network device name {
  accounting-per-attachment;
  description description ;
  device-type (junose | junos | pcmm );
  management-address management-address ;
  no-accounting-per-attachment;
  qos-profile [ qos-profile ...];
}
```

To add a router:

1. From configuration mode, access the configuration statements that configure network devices. You must specify the name of a device with lowercase characters. This procedure uses `junose_boston` as the name of the router.

```
user@host# edit shared network device junose_boston
```

The same procedure can be used for routers running Junos OS.

2. (Optional) Disable the accounting-per-attachment feature to calculate accounting for each policy list on the basis of the accounting flag status of each policy rule in the policy list.

```
[edit shared network device junose_boston]
user@host# set no-accounting-per-attachment
```

**NOTE:**

- To enable the accounting-per-attachment feature, configure the **accounting-per-attachment** option at the [edit shared network device] hierarchy.
- By default, the accounting-per-attachment feature is not set.
- You must restart the ssc connection for the configuration changes to take effect.
- You can configure the accounting-per-attachment feature under the [edit shared sae group *group-name* configuration driver junose] or [edit shared network device] hierarchy. The settings configured under the [edit shared network device] hierarchy override the settings under the [edit shared sae group *group-name* configuration driver junose] hierarchy.

-
3. (Optional) Add a description for the router.

```
[edit shared network device junose_boston]
user@host# set description description
```

4. (Optional) Add the IP address of the router.

```
[edit shared network device junose_boston]
user@host# set management-address management-address
```

5. (Optional) Specify the type of device that you are adding.

```
[edit shared network device junose_boston]
user@host# set device-type junose
```

6. (Optional) Specify quality of service (QoS) profiles that are configured on the router.

```
[edit shared network device junose_boston]
user@host# set qos-profile [ qos-profile ...]
```

7. (Optional) Verify your configuration.

```
[edit shared network device junose_boston]
user@host# show
no-accounting-per-attachment;
description "Juniper Networks E320";
management-address 10.10.8.27;
device-type junose;
qos-profile dhcp-default;
interface-classifier {
  rule rule-0 {
    script #;
  }
}
```


Adding Virtual Routers Individually (SRC CLI)

You can add a virtual router to an existing router.

Use the following configuration statements to add a virtual router:

```
shared network device name virtual-router name {
  sae-connection [ sae-connection ...];
  snmp-read-community snmp-read-community ;
  snmp-write-community snmp-write-community ;
  scope [ scope ...];
  local-address-pools local-address-pools ;
  static-address-pools static-address-pools ;
  tracking-plug-in [ tracking-plug-in ...];
  user-tracking-plug-in user-tracking-plug-in ...];
  authentication-plug-in [ authentication-plug-in ...];
  dual-stack-delay delay
  vpn-id vpn-id;
}
```

To add a virtual router:

1. From configuration mode, access the configuration statements for virtual routers. You must specify the name of a device with lowercase characters. This procedure uses `junose_boston` as the name of the router and `vr1` as the name of the virtual router.

```
user@host# edit shared network device junose_boston virtual-router vr1
```

2. Specify the addresses of SAEs that can manage this router. This step is required for the SAE to work with the router.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set sae-connection [ sae-connection ...]
```

To specify the active SAE and the redundant SAE, enter an exclamation point (!) after the hostname or IP address of the connected SAE. For example:

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set sae-connection [sae1! sae2!]
```

3. (Optional) Specify an SNMP community name for SNMP read-only operations for this virtual router.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set snmp-read-community snmp-read-community
```

4. (Optional) Specify an SNMP community name for SNMP write operations for this virtual router.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set snmp-write-community snmp-write-community
```

5. (Optional) Specify service scopes assigned to this virtual router. The scopes are available for subscribers connected to this virtual router for selecting customized versions of services.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set scope [ scope ...]
```

6. (Optional) Specify the list of IP address pools that a virtual router currently manages and stores.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set local-address-pools local-address-pools
```

7. (Optional) Specify the list of IP address pools that a virtual router manages but does not store.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set static-address-pools static-address-pools
```

8. (Optional) Specify the plug-ins that track interfaces that the SAE manages on this virtual router.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set tracking-plug-in [ tracking-plug-in ...]
```

9. (Optional) Specify a list of plug-ins that are notified of interface events for this virtual router.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set authentication-plug-in [ authentication-plug-in ...]
```

10. (Optional) Specify a single-tracking plug-in or a list of tracking plug-ins used to track subscriber sessions associated with this virtual router.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set user-tracking-plug-in [ user-tracking-plug-in ...]
```

Set the **user-tracking-plug-in** option to the name of the configuration plug-in you configured with the **edit shared sae configuration plug-ins name *name* *ssr-writer*** statement.

11. Configure the delay time (in milliseconds) for dual-stack interfaces. **dual-stack-delay** is not configured by default.

```
[edit shared sae group west-region configuration driver junose]
user@host# dual-stack-delay delay
```

12. (Optional) Specify the VPN identifier used by this virtual router. For edge devices, you can specify VRF instead of a string to use the VRF instance reported by the device as the VPN identifier. For example, if you specify VRF for a JunosE router, the VPN identifier is the name of the virtual router.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set vpn-id (vpn-id | VRF)
```

13. (Optional) Verify your configuration.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# show
authentication-plugin ldap-auth;
dual-stack-delay 1000;
local-address-pools 10.0.0.1;
sae-connection 10.212.10.2;
scope POP-Ottawa;
snmp-read-community *****;
snmp-write-community *****;
static-address-pools 10.10.0.1;
user-tracking-plugin fileAuth;
vpn-id 123;
```

**Related
Documentation**

- [Configuring the SAE to Manage JunosE Routers \(SRC CLI\) on page 59](#)
- [Specifying JunosE Router Initialization Scripts on the SAE \(SRC CLI\) on page 66](#)
- [Configuring Service Scopes \(SRC CLI\)](#)
- [Types of Tracking Plug-Ins](#)
- [Classification Scripts Overview](#)

Configuring the SAE to Manage JunosE Routers (SRC CLI)

To set up the SAE to manage JunosE routers, configure a router driver that specifies a COPS server that can accept COPS connections from the COPS client in JunosE routers.

Use the following configuration statements to configure the SAE to manage JunosE routers:

```
shared sae configuration driver junose {
  accounting-per-attachment;
  backlog backlog ;
  cached-driver-expiration cached-driver-expiration ;
  cops-handler-threads cops-handler-threads ;
  cops-message-maximum-length cops-message-maximum-length ;
  cops-message-read-buffer-size cops-message-read-buffer-size ;
  cops-message-write-buffer-size cops-message-write-buffer-size ;
  drop-unmanaged-interfaces-xdr-driver;
  dual-stack-delay delay
  keepalive-interval keepalive-interval ;
  message-timeout message-timeout ;
  no-accounting-per-attachment;
  pending-address-timeout pending-address-timeout ;
  track-unmanaged-interfaces-xdr-driver;
}
```

To configure the SAE to manage JunosE routers:

1. From configuration mode, access the configuration statement that configures the JunosE router driver. In this sample procedure, the JunosE driver is configured in the west-region group.

```
user@host# edit shared sae group west-region configuration driver junose
```

2. (Optional) Disable the accounting-per-attachment feature to calculate accounting statistics for each policy list on the basis of the accounting flag status of each policy rule in the policy list.

```
[edit shared sae group west-region configuration driver junose]
user@host# set no-accounting-per-attachment
```



NOTE:

- To enable the accounting-per-attachment feature, configure the **accounting-per-attachment** option at the **[edit shared sae group *group-name* configuration driver junose]** hierarchy.
- By default, the accounting-per-attachment feature is not set.
- You must restart the ssc connection for the configuration changes to take effect.
- You can configure the accounting-per-attachment feature under the **[edit shared sae group *group-name* configuration driver junose]** or **[edit shared network device]** hierarchy. The settings configured under the **[edit shared network device]** hierarchy override the settings under the **[edit shared sae group *group-name* configuration driver junose]** hierarchy.

3. Configure the port number of the SAE COPS server. The port number must match the configuration of the SRC client in the JunosE router.

```
[edit shared sae group west-region configuration driver junose]
user@host# set cops-server-port cops-server-port
```

4. Configure the number of outstanding connection attempts before connections are dropped.

```
[edit shared sae group west-region configuration driver junose]
user@host# set backlog backlog
```

5. Configure the delay time (in milliseconds) for dual-stack interfaces. **dual-stack-delay** is not configured by default.

```
[edit shared sae group west-region configuration driver junose]
user@host# dual-stack-delay delay
```

6. Configure the interval between keepalive messages sent from the COPS client (the JunosE router).

```
[edit shared sae group west-region configuration driver junose]
user@host# set keepalive-interval keepalive-interval
```

7. Configure the timeout interval in which the COPS server waits for a response to COPS requests.

```
[edit shared sae group west-region configuration driver junose]
user@host# set message-timeout message-timeout
```

8. Configure the maximum length of a COPS message.

```
[edit shared sae group west-region configuration driver junose]
user@host# set cops-message-maximum-length cops-message-maximum-length
```

9. Configure the buffer size for receiving COPS messages from the JunosE client. We recommend that you use the default setting unless you are instructed to change it by Juniper Networks.

```
[edit shared sae group west-region configuration driver junose]
user@host# set cops-message-read-buffer-size cops-message-read-buffer-size
```

10. Configure the buffer size for sending COPS messages to the JunosE client. We recommend that you use the default setting unless you are instructed to change it by Juniper Networks.

```
[edit shared sae group west-region configuration driver junose]
user@host# set cops-message-write-buffer-size cops-message-read-buffer-size
```

11. Configure the maximum time that a DHCP address request remains pending.

```
[edit shared sae group west-region configuration driver junose]
user@host# set pending-address-timeout pending-address-timeout
```

12. Configure the size of the thread pool for handling unsolicited messages. These threads are shared among all JunosE router drivers.

```
[edit shared sae group west-region configuration driver junose]
user@host# set cops-handler-threads cops-handler-threads
```

13. Configure the minimum amount of time to keep the state of a router driver after its COPS connection has been closed.

```
[edit shared sae group west-region configuration driver junose]
user@host# set cached-driver-expiration cached-driver-expiration
```

14. (Optional) If you are using COPS-XDR, specify whether or not the JunosE router driver keeps a record of unmanaged interfaces.

```
[edit shared sae group west-region configuration driver junose]
user@host# set drop-unmanaged-interfaces-xdr-driver
```

15. (Optional) Enable or disable sending of interface-tracking events for unmanaged interfaces for the XDR router driver.

```
[edit shared sae group west-region configuration driver junose]
user@host# set track-unmanaged-interfaces-xdr-driver
```

16. (Optional) Verify your configuration.

```
[edit shared sae group west-region configuration driver junose]
user@host# show
no-accounting-per-attachment;
cops-server-port 3288;
backlog 50;
```

```
keepalive-interval 45;
message-timeout 120000;
cops-message-maximum-length 200000;
cops-message-read-buffer-size 30000;
cops-message-write-buffer-size 30000;
pending-address-timeout 5000;
cops-handler-threads 20;
cached-driver-expiration 600;
drop-unmanaged-interfaces-xdr-driver;
track-unmanaged-interfaces-xdr-driver;
```

Related Documentation

- [Creating Grouped Configurations for the SAE \(SRC CLI\)](#)
- [Configuring the SAE to Manage JunosE Routers \(C-Web Interface\)](#)
- [Monitoring Interactions Between the SAE and the Router Running JunosE Software on page 71](#)
- [Troubleshooting Problems with Managing JunosE Routers on page 71](#)
- [Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 63](#)
- [SAE Accounting on page 13](#)
- [Adding Routers Individually \(SRC CLI\) on page 55](#)

How SNMP Obtains Information from Routers for the SRC Software

Some scripts in the SRC software use SNMP to get information from the router. For example, the **poolPublisher** router initialization script uses SNMP to read the IP pools.

- On the router, you can configure access to the router's SNMP server. See [Configuring the SNMP Server on the JunosE Router](#).
- On the SAE, you can configure global default SNMP communities that are used for read and write access to the router.
- You can specify SNMP communities for each virtual router. We recommend that you specify communities for each virtual router instead of configuring global communities.

Related Documentation

- [Accessing the Router CLI on page 69](#)
- [Configuring the SNMP Server on the JunosE Router](#)
- [Configuring Global SNMP Communities in the SRC Software \(SRC CLI\) on page 129](#)
- [Configuring Global SNMP Communities in the SRC Software \(C-Web Interface\)](#)
- [Adding JunosE Routers and Virtual Routers \(SRC CLI\) on page 54](#)
- [Adding JunosE Routers and Virtual Routers \(C-Web Interface\)](#)
- [Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 63](#)
- [Specifying JunosE Router Initialization Scripts on the SAE \(SRC CLI\) on page 66](#)

Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers

When the SAE establishes a connection with a router or network device, it can run an initialization script to customize the setup of the connection. These initialization scripts are run when the connection between a router or network device and the SAE is established and again when the connection is dropped.

We provide the `lorPublisher` script in the `/opt/UMC/sae/lib` folder. The `lorPublisher` script publishes the interoperable object reference (IOR) of the SAE in the directory so that a NIC can associate a router with an SAE.

For JunosE VRs that supply IP addresses from a local pool, a router initialization script is provided that identifies which VR supplies each IP pool and writes the information to the configuration. The SAE runs the script only when a COPS connection is established to the JunosE router. Consequently, if you modify information about IP pools on a VR after the COPS connection is established, the SAE will not automatically register the changes, and you must update the configuration.

For Junos (only junos-ise device) virtual routers that supply IP addresses from a local pool, a router initialization script is provided to get the IP pool information from Junos router and update it in LDAP.

[Table 5 on page 63](#) describes the router initialization scripts that we provide with the SRC software in the `/opt/UMC/sae/lib` folder.

Table 5: Router Initialization Scripts

Script Name	Function	When to Use Script
<code>lorPublisher</code>	Publishes the IOR of the SAE into an internal part of the shared configuration so that a NIC can associate a router with an SAE.	<p>Use with JunosE routers that do not supply IP addresses from local pools, and with devices running Junos OS.</p> <p>Use with all devices running Junos OS.</p> <p>Use with third-party network devices.</p>
<code>poolPublisher</code>	Publishes the IOR of the SAE and local IP address pools in the directory so that a NIC can associate a router with an SAE and resolve the IP-to-SAE mapping.	Use with JunosE virtual routers that supply IP addresses from local pools.
<code>isePoolPublisher</code>	Publishes the local IP address pools in the directory so that a NIC can associate a router with an SAE and resolve the IP-to-SAE mapping.	Use with Junos (only junos-ise device) virtual routers that supply IP addresses from local pools.

Interface Object Fields

Router initialization scripts are written in the Python programming language (www.python.org) and executed in the Jython environment (www.jython.org).

Router initialization scripts interact with the SAE through an interface object called Ssp. The SAE exports a number of fields through the interface object to the script and expects the script to provide the entry point to the SAE.

Table 6 on page 64 describes the fields that the SAE exports.

Table 6: Exported Fields

Ssp Attribute	Description
Ssp.properties	System properties object (class: java.util.Properties)—The properties should be treated as read-only by the script.
Ssp.errorLog	Error logger—Use the Ssp.errorLog.println (message) to send error messages to the log.
Ssp.infoLog	Info logger—Use the Ssp.infoLog.println (message) to send informational messages to the log.
Ssp.debugLog	Debug logger—Use the Ssp.debugLog.println (message) to send debug messages to the log.

The router initialization script must set the field Ssp.routerInit to a factory function that instantiates a router initialization object:

- <VRName>—Name of the virtual router in which the COPS client has been configured, format: virtualRouterName@RouterName
- <virtualIp>—Virtual IP address of the SAE (string, dotted decimal; for example: 192.168.254.1)
- <realIp>—Real IP address of the SAE (string, dotted decimal; for example, 192.168.1.20)
- <VRIp>—IP address of the virtual router (string, dotted decimal)
- <transportVR>—Name of the virtual router used for routing the COPS connection, or None, if the COPS client is directly connected

The factory function must implement the following interface:

```
Ssp.routerInit(VRName,
virtualIp,
realIp,
VRIp,
transportVR)
```

The factory function returns an interface object that is used to set up and tear down a connection for a given COPS server. A common case of a factory function is the constructor of a class.

The factory function is called directly after a COPS server connection is established. In case of problems, an exception should be raised that leads to the termination of the COPS connection.

Required Methods

Instances of the interface object must implement the following methods:

- *setup()*—Is called when the COPS server connection is established and is operational. In case of problems, an exception should be raised that leads to the termination of the COPS connection.
- *shutdown()*—Is called when the COPS server connection to the virtual router is terminated. This method should not raise any exceptions in case of problems.

Example: Router Initialization Script

The following script defines a router initialization class named *SillyRouterInit*. The interface class does not implement any useful functionality. The interface class just writes messages to the infoLog when the router connection is created or terminated.

```
class SillyRouterInit:
    def __init__(self, vrName, virtualIp, realIp, vrIp, transportVr):
        """ initialize router initialization object """
        self.vrName = vrName
        Ssp.infoLog.printin("SillyRouterInit created")

    def setup(self):
        """ initialize connection to router """
        Ssp.infoLog.printin("Setup connection to VR %(vrName)s" %
                           vars(self))

    def shutdown(self):
        """ shutdown connection to router """
        Ssp.infoLog.printin("Shutdown connection to VR %(vrName)s" %
                           vars(self))

#
# publish interface object to Ssp core
#
Ssp.routerInit = SillyRouterInit
```

Related Documentation

- [How SNMP Obtains Information from Routers for the SRC Software on page 62](#)
- [Specifying JunosE Router Initialization Scripts on the SAE \(SRC CLI\) on page 66](#)
- [Accessing the Router CLI on page 69](#)
- [Viewing Statistics for Specific JunosE Device Drivers \(SRC CLI\) on page 74](#)
- [Troubleshooting Problems with Managing JunosE Routers on page 71](#)
- [Updating Local IP Address Pools for Junos Virtual Routers \(SRC CLI\) on page 83](#)
- [Updating Local IP Address Pools for JunosE Virtual Routers \(SRC CLI\) on page 67](#)

Specifying JunosE Router Initialization Scripts on the SAE (SRC CLI)

Use the following configuration statements to specify router initialization scripts for JunosE routers:

```
shared sae configuration driver scripts {  
  extension-path extension-path ;  
  general general ;  
  junose-pr junose-pr ;  
  junose-xdr junose-xdr ;  
}
```

To configure router initialization scripts for JunosE routers:

1. From configuration mode, access the configuration statements that configure router initialization scripts. In this sample procedure, the scripts are configured in the west-region group.

```
user@host# edit shared sae group west-region configuration driver scripts
```

2. Specify the script for JunosE routers when the JunosE driver uses COPS-PR mode when connecting to the SAE.

```
[edit shared sae group west-region configuration driver scripts]
```

```
user@host# set junose-pr junose-pr
```

3. Specify the script for JunosE routers when the JunosE driver uses COPS-XDR mode when connecting to the SAE.

```
[edit shared sae group west-region configuration driver scripts]
```

```
user@host# set junose-xdr junose-xdr
```

In COPS-XDR mode, the router does not send the network access server (NAS) IP address to the SAE. If your configuration requires this value, add the following line to a JunosE script:

```
import ERXnasip
```

4. Configure a router initialization script that can be used for all types of routers that the SRC module supports.

```
[edit shared sae group west-region configuration driver scripts]
```

```
user@host# set general general
```

5. Configure a path to router initialization scripts that are not in the default location, */opt/UMC/sae/lib*.

```
[edit shared sae group west-region configuration driver scripts]
```

```
user@host# set extension-path extension-path
```

6. (Optional) Verify your router initialization script configuration.

```
[edit shared sae group west-region configuration driver scripts]
user@host# show
junose-xdr poolPublisher;
```

Related Documentation

- [Accessing the Router CLI on page 69](#)
- [Configuring the SAE to Manage JunosE Routers \(SRC CLI\) on page 59](#)
- [Monitoring Interactions Between the SAE and the Router Running JunosE Software on page 71](#)
- [Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 63](#)

Updating Local IP Address Pools for JunosE Virtual Routers (SRC CLI)

When you reconfigure local IP address pools on a router running JunosE Software, you must update the LDAP directory with the local IP addresses that the virtual router provides.

Before you update local IP address pools, make sure that:

- The JunosE router and VR appear in the LDAP directory.
- The VR has an operating SNMP agent.
- The SAE can communicate with the VR through SNMP
- You have write permissions for the O=Network subtree.

The IP pools are updated in LDAP directory only if there is a change in value of IP address pools on virtual router running JunosE Software.

Use the following command to update local IP address pools on a router running JunosE Software:

```
request sae update ip-pools virtual-router virtual-router <management-address
management-address> <SNMP-community SNMP-community> <server server> <base-dn
base-dn> <principal principal> <credentials credentials>
```

To update IP pools on a router running JunosE Software:

1. Issue the **request sae update ip-pools** command, and specify the **virtual-router** option in the format **virtualRouterName@deviceName**. This text string is case sensitive and must match the name in the JunosE configuration. In this sample procedure the virtual router is **vr1@junose_boston**.

```
user@host> request sae update ip-pools virtual-router vr1@junose_boston
```

2. (Optional) To specify the IP address of the virtual router, use the **management-address** option.
3. (Optional) To specify the SNMP community for the virtual router, use the **SNMP-community** option.



NOTE: When you execute the `request sae update ip-pools` command without the `SNMP-community` option, the SRC software checks if the virtual router has any configured SNMP read community.

- If the virtual router has any configured SNMP community, then that community value from LDAP is set as the SNMP read community.
- If the virtual router does not have any configured SNMP community, then the default public community is set as the SNMP read community.

4. (Optional) To specify the IP address or name of the host that supports the directory, use the **server** option.
5. (Optional) To specify the base DN for the root of the tree to be used, use the **base-dn** option.
6. (Optional) To specify the DN that defines the username with which an SRC component accesses the directory, use the **principal** option.
7. (Optional) To specify the password used for authentication with the directory server, use the **credentials** option.

**Related
Documentation**

- [Adding JunosE Routers and Virtual Routers \(SRC CLI\) on page 54](#)
- [How SNMP Obtains Information from Routers for the SRC Software on page 62](#)
- [Configuring the SAE to Manage JunosE Routers \(SRC CLI\) on page 59](#)
- [Configuring a Schedule for Executing the Commands or Scripts \(SRC CLI\)](#)

Updating Quality of Service Profiles for JunosE Virtual Routers (SRC CLI)

You can use SNMP to read the QoS profile information on routers running JunosE Software, and update the LDAP directory with a list of QoS profiles that are currently configured on the router.

Before you update local QoS profiles, make sure that:

- The JunosE router and VR appear in the directory.
- The VR has an operating SNMP agent.
- The SAE can communicate with the VR through SNMP
- You have write permissions for the O=Network subtree.

Use the following command to update QoS profiles on a router running JunosE Software:

```
request sae update qos-profiles virtual-router virtual-router <management-address  
management-address> <SNMP-community SNMP-community> <server server> <base-dn  
base-dn > <principal principal> <credentials credentials>
```

To update QoS profiles on a router running JunosE Software:

1. Issue the **request sae update qos-profiles** command, and specify the **virtual-router** option in the format `virtualRouterName@deviceName`. This text string is case sensitive and must match the name in the JunosE configuration. In this sample procedure the virtual router is `vr1@junose_boston`.

```
user@host> request sae update qos-profiles virtual-router vr1@junose_boston
```

2. (Optional) To specify the IP address of the virtual router, use the **management-address** option.
3. (Optional) To specify the SNMP community for the virtual router, use the **SNMP-community** option.
4. (Optional) To specify the IP address or name of the host that supports the directory, use the **server** option.
5. (Optional) To specify the base DN for the root of the tree to be used, use the **base-dn** option.
6. (Optional) To specify the DN that defines the username with which an SRC component accesses the directory, use the **principal** option.
7. (Optional) To specify the password used for authentication with the directory server, use the **credentials** option.

Related Documentation

- [Adding JunosE Routers and Virtual Routers \(SRC CLI\) on page 54](#)
- [How SNMP Obtains Information from Routers for the SRC Software on page 62](#)
- [Configuring the SAE to Manage JunosE Routers \(SRC CLI\) on page 59](#)

Accessing the Router CLI

You can access the CLIs of Juniper Networks routers through a Telnet or secure shell connection.

- To open a Telnet session to a router, use the **telnet** operational mode command. For example:

```
user@host> telnet 10.10.10.3
```

- To open a secure shell connection, use the **ssh** operational command. For example:

```
user@host> ssh host 10.10.10.3
```

Related Documentation

- [Specifying JunosE Router Initialization Scripts on the SAE \(SRC CLI\) on page 66](#)
- [Starting the SRC Client on a JunosE Router on page 70](#)
- [Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 63](#)

Starting the SRC Client on a JunosE Router

JunosE routers use an SRC client to interact with the SAE. See *JunosE Broadband Access Configuration Guide* for complete information about configuring the SRC client on a JunosE router.

To start the SRC client:

1. Access the router CLI.
2. Access Global configuration mode.

```
host1# configure terminal
```

3. Switch to the virtual router for which you want to create an SRC client.

```
host1(config)#virtual-router <vrName>
```

4. Enable the SRC client.

To enable COPS-PR mode:

```
host1:<vrName>(config)#sscc enable cops-pr
```

To enable COPS-XDR mode:

```
host1:<vrName>(config)#sscc enable
```

5. Set the primary address from the configuration directory.

```
host1:<vrName>(config)#sscc primary address <ipAddress> port 3288
```

Related Documentation

- [Stopping the SRC Client on a JunosE Router on page 70](#)
- [Accessing the Router CLI on page 69](#)
- [Specifying JunosE Router Initialization Scripts on the SAE \(SRC CLI\) on page 66](#)
- [Viewing Statistics for All JunosE Device Drivers \(SRC CLI\) on page 74](#)

Stopping the SRC Client on a JunosE Router

JunosE routers use an SRC client to interact with the SAE. See *JunosE Broadband Access Configuration Guide* for complete information about configuring the SRC client on the JunosE router.

To stop the SRC client:

1. Access the router CLI.

See [“Accessing the Router CLI” on page 69](#).

2. Access Global configuration mode.

```
host1#configure terminal
```

3. Switch to the virtual router for which you want to stop an SRC client.

```
host1(config)#virtual-router <vrName>
```

4. Disable the SRC client.

```
host1:<vrName>(config)#no ssrc enable
```

- Related Documentation**
- [Starting the SRC Client on a JunosE Router on page 70](#)
 - [Viewing Statistics for All JunosE Device Drivers \(SRC CLI\) on page 74](#)

Monitoring Interactions Between the SAE and the Router Running JunosE Software

Purpose Monitor connection between the SAE and a JunosE router.

Action To monitor the connection between the router and the SAE:

- Use the **show ssrc info** command on the JunosE router

To display the version number of the SRC client:

- Use the **show ssrc version** command on the JunosE router.

See the *JunosE Command Reference Guide* for details about these commands.

You can also monitor the interactions between the SRC module and the router in the log files for the SAE and in the log files generated by the JunosE router.

- For information about configuring logging on JunosE routers, see *JunosE System Event Logging Reference Guide*.

- Related Documentation**
- [Specifying JunosE Router Initialization Scripts on the SAE \(SRC CLI\) on page 66](#)
 - [Configuring the SAE to Manage JunosE Routers \(SRC CLI\) on page 59](#)
 - [Troubleshooting Problems with Managing JunosE Routers on page 71](#)

Troubleshooting Problems with Managing JunosE Routers

Problem SRC client or JunosE router is not working as expected.

Solution You can troubleshoot problems with the SRC client on JunosE routers and with managed JunosE routers, interfaces, and services on the SAE.

To troubleshoot SRC problems on the router:

1. Look at the log files for the SAE and the log files generated by the SRC client on the JunosE router.
 - If the log files indicate a problem with specific interfaces on the router, review the configuration of the associated policies in the SRC module, and fix any errors.

- If the log files indicate a problem with a specific service or its associated policy rules, review the configuration of the service or policies in the SRC module, and fix any errors.
 - If the log files indicate only that the SRC client is not responding, ensure that the values in the SAE configuration match the values in the SRC client configuration on the router.
2. Restart the SRC client on the JunosE router.

When you restart the SRC client, the SRC client removes all policies that were installed by the SRC module and reports all interfaces again.



NOTE: DHCP addresses that were managed are not reported again, so we recommend that you do not restart the SRC client if you are managing DHCP sessions.

To restart the SRC client in COPS-PR mode, enter the following commands:

```
host1:<vrName>(config)#no ssc enable
host1:<vrName>(config)#sscc enable cops-pr
```

To restart the SRC client in COPS-XDR mode, enter the following commands:

```
host1:<vrName>(config)#no ssc enable
host1:<vrName>(config)#sscc enable
```

If restarting the SRC client does not resolve the problem, rebuild the router configuration and restart the client.

**Related
Documentation**

- [Monitoring Interactions Between the SAE and the Router Running JunosE Software on page 71](#)
- [Viewing the State of JunosE Device Drivers \(SRC CLI\) on page 72](#)
- [Viewing Statistics for Specific JunosE Device Drivers \(SRC CLI\) on page 74](#)
- [Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 63](#)

Viewing the State of JunosE Device Drivers (SRC CLI)

Purpose Display the state of JunosE drivers.

Action Use the following operational mode command:

```
show sae drivers <device-name device-name> < (brief) > <maximum-results  
maximum-results>
```


For example:

```

user@host> show sae drivers device-name default@dryad
JunosE Driver
Device name                        default@dryad
Device type                        junose
Device IP                         10.227.7.244
Local IP                          10.227.7.172
TransportRouter                   default@dryad
Device version                     7.2.0
Start time                        Tue Feb 13 14:18:44 EST 2007
Number of notifications            20
Number of processed added          14
Number of processed changed        0
Number of processed deleted        6
Number of provisioning attempt     30
Number of provisioning attempt failed 0
Number of outstanding decisions    0
Number of SAP                      7
Number of PAP                      1
Number of active user sessions     100
Number of active service sessions  0
  Job Queue
  Size                0
  Age (ms)            1
  Total enqueued      28
  Total dequeued      28
  Average job time (ms) 426
  State Synchronization
  Number recovered subscriber sessions 0
  Number recovered service sessions    0
  Number recovered interface sessions  0
  Number invalid subscriber sessions    0
  Number invalid service sessions      0
  Number invalid interface sessions    0
  Background restoration start time     Tue Feb 13 14:18:49 EST 2007
  Background restoration end time       Tue Feb 13 14:18:49 EST 2007
  Number subscriber sessions restored in background 0
  Number of provisioning objects left to collect 0
  Total number of provisioning objects to collect 11
  Start time                           Tue Feb 13 14:18:45 EST 2007
  End time                             Tue Feb 13 14:18:47 EST 2007
  Number of synched contexts            7
  Number of post-sync jobs              6
Session Store Info
Session Store Status                   sessionsCollected
Status Last Update Time                Mon Jul 29 10:26:26 UTC 2013
Current Usage Ratio                    1.0
Last Modified Time                     Mon Jul 29 10:27:05 UTC 2013
Size(KB) Name LiveSessionsSize(KB)
117.9 storeOps_1_1 117.9

```

Related Documentation

- *COPS Connection Between JunosE Routers and the SAE*
- [Troubleshooting Problems with Managing JunosE Routers on page 71](#)
- [Viewing the State of JunosE Device Drivers \(C-Web Interface\) on page 75](#)

- [Monitoring Interactions Between the SAE and the Router Running JunosE Software on page 71](#)

Viewing Statistics for Specific JunosE Device Drivers (SRC CLI)

Purpose Display statistics for a specific JunosE device driver.

Action Use the following operational mode command:

```
show sae statistics device <name name> < (brief) >
```

For example:

```
user@host> show sae statistics device name default@dryad
SNMP Statistics
Add notification handle time      6
Change notification handle time   0
Client ID                        default@dryad
Delete notification handle time   0
Failover IP                      0.0.0.0
Failover port                    0
Handle message time              60
Job queue age                    0
Job queue time                   4
Number message send               158
Number of added jobs              9
Number of add notifications       4
Number of change notifications    0
Number of delete notifications    0
Number of managed interfaces      4
Number of message errors          0
Number of message timeouts        0
Number of removed jobs            9
Number of user session established 0
Number of user session removed    0
Router type                      JunosE COPS
Up time                          172286
Using failover server             false
Number of active user sessions    100
Number of active service sessions 0
```

- Related Documentation**
- [Troubleshooting Problems with Managing JunosE Routers on page 71](#)
 - [Viewing the State of JunosE Device Drivers \(SRC CLI\) on page 72](#)
 - [Viewing Statistics for All JunosE Device Drivers \(SRC CLI\) on page 74](#)
 - [Monitoring Interactions Between the SAE and the Router Running JunosE Software on page 71](#)

Viewing Statistics for All JunosE Device Drivers (SRC CLI)

Purpose Display SNMP statistics for all JunosE device drivers.

Action Use the following operational mode command:

```
show sae statistics device common junose-cops
```

For example:

```
user@host> show sae statistics device common junose-cops
SNMP Statistics
Driver type                JunosE COPS
Number of close requests    0
Number of connections accepted 2
Number of current connections 1
Number of open requests     2
Server address              0:0:0:0:0:0:0
Server port                 3288
Time since last redirect    186703
```

- Related Documentation**
- [Troubleshooting Problems with Managing JunosE Routers on page 71](#)
 - [Viewing the State of JunosE Device Drivers \(SRC CLI\) on page 72](#)
 - [Viewing Statistics for Specific JunosE Device Drivers \(SRC CLI\) on page 74](#)
 - [Viewing Statistics for All JunosE Device Drivers \(C-Web Interface\) on page 75](#)

Viewing the State of JunosE Device Drivers (C-Web Interface)

Purpose If the log files indicate a problem with a specific driver, review the configuration of the associated JunosE device driver with the C-Web interface.

- Action**
1. Click **Monitor>SAE>Drivers**.
The Drivers pane appears.
 2. Enter information as described in the Help text in the main pane, and click **OK**.
The Drivers pane displays information about the JunosE device driver.

- Related Documentation**
- [Troubleshooting Problems with Managing JunosE Routers on page 71](#)
 - [Monitoring Interactions Between the SAE and the Router Running JunosE Software on page 71](#)
 - [Viewing Statistics for All JunosE Device Drivers \(C-Web Interface\) on page 75](#)

Viewing Statistics for All JunosE Device Drivers (C-Web Interface)

Purpose To view SNMP statistics for all JunosE device driver:

- Action**
1. Click **Monitor>SAE>Statistics>Device>Common**.
The Common pane appears.
 2. Enter information as described in the Help text in the main pane, and click **OK**.
The Common pane displays statistics for the JunosE device driver.

**Related
Documentation**

- [Troubleshooting Problems with Managing JunosE Routers on page 71](#)
- [Viewing Statistics for Specific JunosE Device Drivers \(SRC CLI\) on page 74](#)
- [Viewing the State of JunosE Device Drivers \(C-Web Interface\) on page 75](#)
- [Viewing Statistics for All JunosE Device Drivers \(SRC CLI\) on page 74](#)

CHAPTER 7

Using Devices Running Junos OS in the SRC Network (SRC CLI)

- [BEEP Connection Between Devices Running Junos OS and the SAE on page 78](#)
- [Managing DMI Devices on Routers Running Junos OS Using the SRC Software and Junos Space on page 78](#)
- [Adding Devices Running Junos OS and Virtual Routers \(SRC CLI\) on page 79](#)
- [Updating Local IP Address Pools for Junos Virtual Routers \(SRC CLI\) on page 83](#)
- [Retrieving Local IP Address Pool Information from Junos Router During Router Initialization and Pool Republishing \(SRC CLI\) on page 84](#)
- [Configuring the SAE to Manage Devices Running Junos OS \(SRC CLI\) on page 86](#)
- [Configuring Secure Connections Between the SAE and Devices Running Junos OS on page 88](#)
- [Adding the Server Certificate on the Device on page 89](#)
- [Creating a Client Certificate for the Router on page 90](#)
- [Adding the Client Certificate on the Router on page 90](#)
- [Configuring the SAE to Use TLS \(SRC CLI\) on page 91](#)
- [Configuring TLS on the SAE \(SRC CLI\) on page 91](#)
- [SAE Verification of Junos OS Configuration Changes on page 92](#)
- [Setting Up Periodic Configuration Checking \(SRC CLI\) on page 92](#)
- [Using SNMP to Retrieve Information from JunosE Routers and Devices Running Junos OSs \(SRC CLI\) on page 93](#)
- [Specifying Router Initialization Scripts on the SAE \(SRC CLI\) on page 94](#)
- [Configuring Devices Running Junos OS to Interact with the SAE on page 95](#)
- [SAE Tracking for LSPs Configured on Devices Running Junos OS on page 96](#)
- [Configuring the Device Running Junos OS to Apply Changes It Receives from the SAE on page 97](#)
- [Disabling Interactions Between the SAE and Devices Running Junos OS on page 97](#)
- [Monitoring Interactions Between the SAE and Devices Running Junos OS on page 98](#)
- [Troubleshooting Problems Between the SRC module and Device Drivers Running Junos OS on page 98](#)

BEEP Connection Between Devices Running Junos OS and the SAE

For information about which devices running Junos OS and releases a particular SRC release supports, see the *SRC Release Notes*.

The SAE interacts with a Junos OS process, referred to as the SRC software process in this documentation, on the devices running Junos OS. The SAE and the SRC software process communicate using the Blocks Extensible Exchange Protocol (BEEP). You can secure the BEEP connection by using Transport Layer Security (TLS).

When the SRC software process establishes a BEEP session for the SAE, the SAE configures an interface on the devices running Junos OS. The SAE builds the configuration for an interface using the policies stored in the directory. If the policies are subsequently modified, the SAE builds a new configuration and reconfigures the interface on the devices running Junos OS. The device running Junos OS stores data about interfaces and services that the SAE manages in a configuration group called *sdx*. You must create this configuration group on the device running Junos OS.

Related Documentation

- [Adding Operative Devices Running Junos OS \(C-Web Interface\)](#)
- [Configuring the SAE to Manage Devices Running Junos OS \(SRC CLI\) on page 86](#)
- [Configuring Secure Connections Between the SAE and Devices Running Junos OS on page 88](#)

Managing DMI Devices on Routers Running Junos OS Using the SRC Software and Junos Space

For information about which devices running Junos OS and releases a particular SRC release supports, see the *SRC Release Notes*.

Using the SRC Device Management Interface (DMI) driver and Junos Space, the SRC software can manage DMI devices connected to routers running Junos OS. The SRC software communicates with Junos Space using the representational state transfer (REST) over HTTP(S), and Junos Space manages the router running Junos OS over the DMI. The SRC software recognizes and receives notifications for changes to DMI devices connected to the router, allowing you to offer dynamic services on those devices. In addition, you can define and automatically provision policies for DMI devices, provide per-subscriber accounting for services on DMI devices, and develop script services for service sessions residing on DMI-managed devices.

The router stores data about interfaces and services that the SAE manages in a configuration group called *sdx*. When you use the DMI driver, the SRC software automatically creates this configuration group on the router.

Related Documentation

- [Managing DMI Devices Using the SRC Software and Junos Space Overview on page 106](#)
- [Migrating from the Junos \(BEEP\) Driver to the Junos DMI Driver \(SRC CLI\) on page 110](#)

Adding Devices Running Junos OS and Virtual Routers (SRC CLI)

On devices running Junos OS, the SAE manages interfaces. The SRC module associates a virtual router called default with each device running Junos OS. Each device running Junos OS in the SRC network and its associated virtual router (VR) called default must appear in the directory. The VRs are not actually configured on the device running Junos OS; the VR in the directory provides a way for the SAE to manage the interfaces on the device running Junos OS.

You can add routers the following ways:

- [Adding Operative Devices Running Junos OS \(SRC CLI\) on page 79](#)
- [Adding Routers Individually \(SRC CLI\) on page 79](#)
- [Adding Virtual Routers Individually \(SRC CLI\) on page 81](#)

Adding Operative Devices Running Junos OS (SRC CLI)

To add to the directory routers and Junos OS VRs that are currently operative and have an operating SNMP agent:

- In operational mode, enter the following command:

```
request network discovery network network <community community >
```

where:

- *network* —Address (with or without mask) of the network to discover
- *community* —Name of the SNMP community to which the devices belong

If you add a router using the discover network feature, the software adds the IP address of the first SNMP agent on the router to respond to the discover request.

Adding Routers Individually (SRC CLI)

Use the following configuration statements to add a router:

```
shared network device name {
  accounting-per-attachment;
  description description ;
  device-type (junose | junos | pcmm );
  management-address management-address ;
  no-accounting-per-attachment;
  qos-profile [ qos-profile ...];
}
```

To add a router:

1. From configuration mode, access the configuration statements that configure network devices. You must specify the name of a device with lowercase characters. This procedure uses junose_boston as the name of the router.

```
user@host# edit shared network device junose_boston
```

The same procedure can be used for routers running Junos OS.

2. (Optional) Disable the accounting-per-attachment feature to calculate accounting for each policy list on the basis of the accounting flag status of each policy rule in the policy list.

```
[edit shared network device junose_boston]
user@host# set no-accounting-per-attachment
```



NOTE:

- To enable the accounting-per-attachment feature, configure the **accounting-per-attachment** option at the **[edit shared network device]** hierarchy.
 - By default, the accounting-per-attachment feature is not set.
 - You must restart the ssc connection for the configuration changes to take effect.
 - You can configure the accounting-per-attachment feature under the **[edit shared sae group *group-name* configuration driver junose]** or **[edit shared network device]** hierarchy. The settings configured under the **[edit shared network device]** hierarchy override the settings under the **[edit shared sae group *group-name* configuration driver junose]** hierarchy.
-

3. (Optional) Add a description for the router.

```
[edit shared network device junose_boston]
user@host# set description description
```

4. (Optional) Add the IP address of the router.

```
[edit shared network device junose_boston]
user@host# set management-address management-address
```

5. (Optional) Specify the type of device that you are adding.

```
[edit shared network device junose_boston]
user@host# set device-type junose
```

6. (Optional) Specify quality of service (QoS) profiles that are configured on the router.

```
[edit shared network device junose_boston]
user@host# set qos-profile [ qos-profile ...]
```

7. (Optional) Verify your configuration.

```
[edit shared network device junose_boston]
user@host# show
no-accounting-per-attachment;
description "Juniper Networks E320";
management-address 10.10.8.27;
device-type junose;
```



```

qos-profile dhcp-default;
interface-classifier {
    rule rule-0 {
        script #;
    }
}

```

Adding Virtual Routers Individually (SRC CLI)

You can add a virtual router to an existing router.

Use the following configuration statements to add a virtual router:

```

shared network device name virtual-router name {
    sae-connection [ sae-connection ...];
    snmp-read-community snmp-read-community ;
    snmp-write-community snmp-write-community ;
    scope [ scope ...];
    local-address-pools local-address-pools ;
    static-address-pools static-address-pools ;
    tracking-plug-in [ tracking-plug-in ...];
    user-tracking-plug-in user-tracking-plug-in ...];
    authentication-plug-in [ authentication-plug-in ...];
    dual-stack-delay delay
    vpn-id vpn-id;
}

```

To add a virtual router:

1. From configuration mode, access the configuration statements for virtual routers. You must specify the name of a device with lowercase characters. This procedure uses `junose_boston` as the name of the router and `vr1` as the name of the virtual router.

```

user@host# edit shared network device junose_boston virtual-router vr1

```

2. Specify the addresses of SAEs that can manage this router. This step is required for the SAE to work with the router.

```

[edit shared network device junose_boston virtual-router vr1]
user@host# set sae-connection [ sae-connection ...]

```

To specify the active SAE and the redundant SAE, enter an exclamation point (!) after the hostname or IP address of the connected SAE. For example:

```

[edit shared network device junose_boston virtual-router vr1]
user@host# set sae-connection [sae1! sae2!]

```

3. (Optional) Specify an SNMP community name for SNMP read-only operations for this virtual router.

```

[edit shared network device junose_boston virtual-router vr1]
user@host# set snmp-read-community snmp-read-community

```

4. (Optional) Specify an SNMP community name for SNMP write operations for this virtual router.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set snmp-write-community snmp-write-community
```

5. (Optional) Specify service scopes assigned to this virtual router. The scopes are available for subscribers connected to this virtual router for selecting customized versions of services.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set scope [ scope ...]
```

6. (Optional) Specify the list of IP address pools that a virtual router currently manages and stores.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set local-address-pools local-address-pools
```

7. (Optional) Specify the list of IP address pools that a virtual router manages but does not store.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set static-address-pools static-address-pools
```

8. (Optional) Specify the plug-ins that track interfaces that the SAE manages on this virtual router.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set tracking-plugin-in [ tracking-plugin-in ...]
```

9. (Optional) Specify a list of plug-ins that are notified of interface events for this virtual router.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set authentication-plugin-in [ authentication-plugin-in ...]
```

10. (Optional) Specify a single-tracking plug-in or a list of tracking plug-ins used to track subscriber sessions associated with this virtual router.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set user-tracking-plugin-in [ user-tracking-plugin-in ...]
```

Set the **user-tracking-plugin-in** option to the name of the configuration plug-in you configured with the **edit shared sae configuration plug-ins name *name* *ssr-writer*** statement.

11. Configure the delay time (in milliseconds) for dual-stack interfaces. **dual-stack-delay** is not configured by default.

```
[edit shared sae group west-region configuration driver junose]
user@host# dual-stack-delay delay
```

12. (Optional) Specify the VPN identifier used by this virtual router. For edge devices, you can specify VRF instead of a string to use the VRF instance reported by the device as the VPN identifier. For example, if you specify VRF for a JunosE router, the VPN identifier is the name of the virtual router.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# set vpn-id (vpn-id | VRF)
```

13. (Optional) Verify your configuration.

```
[edit shared network device junose_boston virtual-router vr1]
user@host# show
authentication-plugin ldap-auth;
dual-stack-delay 1000;
local-address-pools 10.0.0.1;
sae-connection 10.212.10.2;
scope POP-Ottawa;
snmp-read-community *****;
snmp-write-community *****;
static-address-pools 10.10.0.1;
user-tracking-plugin fileAuth;
vpn-id 123;
```

Related Documentation

- [Configuring the SAE to Manage JunosE Routers \(SRC CLI\) on page 59](#)
- [Specifying JunosE Router Initialization Scripts on the SAE \(SRC CLI\) on page 66](#)
- [Configuring Service Scopes \(SRC CLI\)](#)
- [Types of Tracking Plug-Ins](#)
- [Classification Scripts Overview](#)

Updating Local IP Address Pools for Junos Virtual Routers (SRC CLI)

When you reconfigure local IP address pools on a router running Junos Software, you must update the LDAP directory with the local IP addresses that the virtual router provides.



NOTE: The IP address pool updates feature is available only on Junos devices of type `junos-ise`. This feature is not available on other Junos devices such as `junos`, `junos-ptsp`, and `junos-dmi`.

Before you update local IP address pools, make sure that:

- The virtual router and `junos-ise` device appear in the LDAP directory.
- Telnet or SSH service is configured on the Junos router.
- Telnet or SSH users are created for the Junos router.
- Telnet or SSH connection authentication parameters are configured by using the **pool-retrieval** option.

For more information about the pool retrieval parameters, see [“Retrieving Local IP Address Pool Information from Junos Router During Router Initialization and Pool Republishing \(SRC CLI\)” on page 84](#).

The IP pools are updated in LDAP directory only if there is a change in value of IP address pools on virtual router running Junos Software and LDAP.

Use the following command to update local IP address pools on a router running Junos Software:

```
request sae update ip-pools virtual-router virtual-router
```

To update IP pools from a router running Junos Software:

1. Issue the **request sae update ip-pools** command, and specify the **virtual-router** option in the format `virtualRouterName@deviceName`. In this sample procedure the virtual router is `vr1@junos-ise`.

```
user@host> request sae update ip-pools virtual-router vvr1@junos-ise
```

**Related
Documentation**

- [Updating Local IP Address Pools for JunosE Virtual Routers \(SRC CLI\) on page 67](#)
- [Configuring a Schedule for Executing the Commands or Scripts \(SRC CLI\)](#)
- [Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 63](#)

Retrieving Local IP Address Pool Information from Junos Router During Router Initialization and Pool Republishing (SRC CLI)

During the router initialization and pool republishing, you need to configure the Telnet or SSH connection authentication parameters to retrieve the IP pool information from a router running Junos.

You can configure the Telnet or SSH connection authentication parameters by using the **pool-retrieval** option under the **[edit shared sae group group-name configuration driver junos-ise]** or **[edit shared network device]** hierarchy. The settings configured under the **[edit shared network device]** hierarchy override the settings under the **[edit shared sae group group-name configuration driver junos-ise]** hierarchy.



NOTE: The IP address pool retrieving feature is available only on Junos devices of type `junos-ise`. This feature is not available on other Junos devices such as `junos`, `junos-ptsp`, and `junos-dmi`.

To establish the connection with a router running Junos OS, you can configure any one of the following pool retrieval parameters:

- The Telnet username and password created for the Junos router.
- The SSH username and password created for the Junos router.
- The SSH key authentication file created by using the **request system generate-ssh-key ssh-key-name ssh-key-name** command.

For pool republishing from a router running Junos OS, you can use the **request sae update ip-pools** command. For more information, see [“Updating Local IP Address Pools for Junos Virtual Routers \(SRC CLI\)” on page 83](#).

Use the following statements to configure the pool retrieval parameter in driver level for communicating and retrieving the IP pool information from a Junos router.

```
shared sae group group-name configuration driver junos-ise pool-retrieval{
  junos-login-name junos-login-name;
  junos-password junos-password;
  junos-protocol (ssh | telnet);
  key-file-name key-file-name;
  netconf-port netconf-port;
}
```

Use the following statements to configure the pool retrieval parameter in device level for communicating and retrieving the IP pool information from a Junos router.

```
edit shared network device <name> pool-retrieval{
  junos-login-name junos-login-name;
  junos-password junos-password;
  junos-protocol (ssh | telnet);
  key-file-name key-file-name;
  netconf-port netconf-port;
}
```

To configure the pool retrieval parameters:

1. In configuration mode, specify the pool retrieval option.

```
[edit]
user@host# edit shared sae group POP-ID configuration driver junos-ise pool-retrieval
```

or

```
[edit]
user@host# edit shared network device <name> pool-retrieval
```

2. (Optional) Enter the Telnet or SSH login user name that is created for the junos-ise device.

```
[edit shared sae group POP-ID configuration driver junos-ise pool-retrieval]
user@host# set junos-login-name junos-login-name
```

3. (Optional) Enter the Telnet or SSH login password which is created for the junos-ise device.

```
[edit shared sae group POP-ID configuration driver junos-ise pool-retrieval]
user@host# set junos-password junos-password
```

4. Configure the protocol in which the communication happens with the junos-ise device. The communication protocol can be SSH or Telnet.

```
[edit shared sae group POP-ID configuration driver junos-ise pool-retrieval]
user@host# set junos-protocol (ssh | telnet)
```

5. (Optional) Configure the SSH key authentication file name for junos-ise device that connect through SSH protocol and authenticates through SSH key authentication mechanism. This option is valid only for the SSH protocol.

```
[edit shared sae group POP-ID configuration driver junos-ise pool-retrieval]
user@host# set key-file-name key-file-name
```



NOTE: You can use the SSH key authentication file which contains DSA public keys when you log in to the junos-ise device.

- To create a DSA public key, use the `request system generate-ssh-key ssh-key-name ssh-key-name` command. You can view the generated DSA public key by using the `show system generated-keys ssh-key-name` command.
- To delete the DSA public key, use the `request system remove-ssh-key ssh-key-name` command.

6. (Optional) Enter the port number for establishing the connection. The value ranges from 1 through 65,535. By default, the port value is set to 830 for SSH and 23 for Telnet.

```
[edit shared sae group POP-ID configuration driver junos-ise pool-retrieval]
user@host# set netconf-port netconf-port
```

7. (Optional) Verify your configuration.

```
user@host# show
pool-retrieval {
  junos-login-name regress;
  junos-protocol SSH;
  key-file-name ssh-key-name;
  netconf-port 830;
}
```

Related Documentation

- [Configuring a Schedule for Executing the Commands or Scripts \(SRC CLI\)](#)
- [Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 63](#)

Configuring the SAE to Manage Devices Running Junos OS (SRC CLI)

A device running Junos OS interacts with the SAE by using a Junos OS process called `sdx`. When the `sdx` process establishes a TCP/IP connection to the SAE, the SAE begins to manage the router. The Junos router driver configuration defines parameters related to the interactions between the SAE and the `sdx` process.

Use the following configuration statements to configure the Junos router driver:

```
shared sae configuration driver junos {
  beep-server-port beep-server-port ;
  tls-beep-server-port tls-beep-server-port ;
  connection-attempts connection-attempts ;
  keepalive-interval keepalive-interval ;
  message-timeout message-timeout ;
  batch-size batch-size ;
  transaction-batch-time transaction-batch-time ;
  sdx-group-name sdx-group-name ;
  sdx-session-group-name sdx-session-group-name ;
```

```

    send-commit-check send-commit-check ;
}

```

To configure the Junos router driver:

1. From configuration mode, access the configuration statement that configures the Junos router driver. In this sample procedure, the Junos driver is configured in the west-region group.

```

user@host# edit shared sae group west-region configuration driver junos

```

2. Specify the TCP port number that is used to communicate with the sdx process on devices running Junos OS. This port number must match the port number configured in the sdx process on the router.

If you set this value to zero and the TLS BEEP server port is set, the SAE accepts only TLS connections.

```

[edit shared sae group west-region configuration driver junos]
user@host# set beep-server-port beep-server-port

```

3. Specify the TLS port number that is used for TLS connections to the device running Junos OS.

If you set this value to zero, the SAE does not accept TLS connections.

```

[edit shared sae group west-region configuration driver junos]
user@host# set tls-beep-server-port tls-beep-server-port

```

4. Specify the number of outstanding connection attempts before new connection attempts are dropped.

```

[edit shared sae group west-region configuration driver junos]
user@host# set connection-attempts connection-attempts

```

5. Specify the interval between keepalive messages sent from the router.

```

[edit shared sae group west-region configuration driver junos]
user@host# set keepalive-interval keepalive-interval

```

6. Specify the amount of time that the router driver waits for a response from the sdx process.

Under a high load the router may not be able to respond fast enough to requests. Change this value only if a high number of timeout events appear in the error log.

```

[edit shared sae group west-region configuration driver junos]
user@host# set message-timeout message-timeout

```

7. Specify the minimum number of service configuration transactions that are committed at the same time

```

[edit shared sae group west-region configuration driver junos]
user@host# set batch-size batch-size

```

8. Specify the maximum time to collect configuration transactions in a batch.

```
[edit shared sae group west-region configuration driver junos]
user@host# set transaction-batch-time transaction-batch-time
```

9. Specify the name of a session group on the device running Junos OS in which provisioning objects are stored.

```
[edit shared sae group west-region configuration driver junos]
user@host# set sdx-session-group-name sdx-session-group-name
```

10. Enable or disable commit check. If enabled, a more detailed error message is logged if a batch fails, which lets you verify individual transactions in a batch.

```
[edit shared sae group west-region configuration driver junos]
user@host# set send-commit-check send-commit-check
```

11. (Optional) Verify your configuration.

```
[edit shared sae group west-region configuration driver junos]
user@host# show
beep-server-port 3333;
tls-beep-server-port 0;
connection-attempts 50;
keepalive-interval 45;
message-timeout 30000;
batch-size 10;
transaction-batch-time 2000;
sdx-group-name sdx;
sdx-session-group-name sdx-sessions;
send-commit-check true;
```

Related Documentation

- *Creating Grouped Configurations for the SAE (SRC CLI)*
- *Configuring the SAE to Manage Devices Running Junos OS (C-Web Interface)*
- [Configuring Secure Connections Between the SAE and Devices Running Junos OS on page 88](#)
- [Configuring Devices Running Junos OS to Interact with the SAE on page 95](#)
- [Monitoring Interactions Between the SAE and Devices Running Junos OS on page 98](#)

Configuring Secure Connections Between the SAE and Devices Running Junos OS

You can use TLS to protect communication between the SAE and devices running Junos OS.

To complete the handshaking protocol for the TLS connection, the client (device running Junos OS) and the server (SAE) must exchange and verify certificates. You need to create a client certificate and a server certificate. Both certificates must be signed by a certificate authority (CA). Junos OS supports VeriSign, Inc. (<http://www.verisign.com>). You must then install both certificates on the SAE and on the device running Junos OS.

You can use SRC CLI commands to manage certificates manually, or through the Simple Certificate Enrollment Protocol (SCEP).

Certificates are in the format defined in the X.509 standard for public key infrastructure. The certificate requests are in the Public Key Cryptology Standard (PKCS) #10 format.

Tasks to set up the SAE and the device running Junos OS to use TLS are:

1. [Adding the Server Certificate on the Device on page 89](#)
2. [Creating a Client Certificate for the Router on page 90](#)
3. [Adding the Client Certificate on the Router on page 90](#)
4. [Configuring the SAE to Use TLS \(SRC CLI\) on page 91](#)
5. [Configuring TLS on the SAE \(SRC CLI\) on page 91](#)

Related Documentation

- [Configuring Secure Connections Between the SAE and Devices Running Junos OS](#)
- [Configuring the SAE to Manage Devices Running Junos OS \(SRC CLI\) on page 86](#)
- [BEEP Connection Between Devices Running Junos OS and the SAE on page 78](#)

Adding the Server Certificate on the Device

The TLS client (device running Junos OS) needs a copy of the certificate that was used to sign the SAE certificate so that it can verify the SAE certificate. To install the SAE certificate on the device running Junos OS:

1. Include the following statements at the `[edit security certificates certificate-authority]` hierarchy level.

```
[edit security certificates certificate-authority]
security{
  certificates{
    certificate-authority SAE Cert{
      file /var/db/certs/cert.pem;
    }
  }
}
```

2. Include the following statements at the `[system services service-deployment]` hierarchy level.

```
system{
  services{
    service-deployment{
      servers {
        server-address port port-number{
          security-options {
            tls;
          }
        }
      }
    }
  }
}
```

```
    }  
  }  
}
```

Related Documentation

- [Configuring Secure Connections Between the SAE and Devices Running Junos OS on page 88](#)
- [Creating a Client Certificate for the Router on page 90](#)
- [Adding the Client Certificate on the Router on page 90](#)
- [Configuring the SAE to Use TLS \(SRC CLI\) on page 91](#)

Creating a Client Certificate for the Router

For information about how to obtain a certificate for the router from a certificate authority, see *Obtaining a Certificate from a Certificate Authority* in the *Junos OS System Basics Configuration Guide*.

Related Documentation

- [Configuring Secure Connections Between the SAE and Devices Running Junos OS on page 88](#)
- [Adding the Server Certificate on the Device on page 89](#)
- [Adding the Client Certificate on the Router on page 90](#)

Adding the Client Certificate on the Router

To install the client (router) certificate on the device running Junos OS:

1. Include the following statements at the **[edit security certificates certificate-authority]** hierarchy level.

```
[edit security certificates certificate-authority]  
security{  
  certificates{  
  }  
}
```

2. Include the following statements at the **[system services service-deployment]** hierarchy level.

```
system{  
  services{  
    service-deployment{  
      local-certificate clientCert;  
    }  
  }  
}
```

Related Documentation

- [Configuring Secure Connections Between the SAE and Devices Running Junos OS on page 88](#)

- [Adding the Server Certificate on the Device on page 89](#)
- [Creating a Client Certificate for the Router on page 90](#)
- [Removing a Certificate Request](#)

Configuring the SAE to Use TLS (SRC CLI)

To configure the SAE to accept TLS connections, enter a port number with the **set beep-server-port** command in the Junos router driver configuration.

See “[Configuring the SAE to Manage Devices Running Junos OS \(SRC CLI\)](#)” on page 86

Related Documentation

- [Configuring the SAE to Use TLS](#)
- [Configuring TLS on the SAE \(SRC CLI\) on page 91](#)
- [Configuring Devices Running Junos OS to Interact with the SAE on page 95](#)
- [Monitoring Interactions Between the SAE and Devices Running Junos OS on page 98](#)

Configuring TLS on the SAE (SRC CLI)

Use the following configuration statements to configure TLS on the SAE:

```
shared sae configuration driver junos security {
  need-client-authentication;
  certificate-identifier private-key;
}
```

To configure TLS on the SAE:

1. From configuration mode, access the configuration statement that configures security for the Junos TLS connection. In this sample procedure, the Junos driver is configured in the west-region group.

```
user@host# edit shared sae group west-region configuration driver junos security
```

2. (Optional) Specify whether or not the SAE requests a client certificate from the router when a connection to the router is established.

```
[edit shared sae group west-region configuration driver junos security]
user@host# set need-client-authentication
```

3. Specify the name of certificate to be used for TLS communications.

```
[edit shared sae group west-region configuration driver junos security]
user@host# set certificate-identifier private-key
```

4. (Optional) Verify your TLS configuration.

```
[edit shared sae group west-region configuration driver junos security]
user@host# show
```

```
need-client-authentication;  
certificate-identifier privatekey;
```

**Related
Documentation**

- [Configuring TLS on the SAE](#)
- [Configuring the SAE to Use TLS \(SRC CLI\) on page 91](#)
- [Configuring the SAE to Manage Devices Running Junos OS \(SRC CLI\) on page 86](#)
- [Monitoring Interactions Between the SAE and Devices Running Junos OS on page 98](#)
- [BEEP Connection Between Devices Running Junos OS and the SAE on page 78](#)

SAE Verification of Junos OS Configuration Changes

The SAE can check the configuration of a device running Junos OS under its control to detect whether the configuration has changed by a means other than through the SAE. If the SAE finds a disparity between the router and the SAE configurations, it can take several actions. The SAE checks the configuration installed on the router against the state of the SAE session layer (subscriber, service, and interface sessions). While the check is occurring, the SAE does not handle jobs from the router, and all provisioning activity is blocked, including event notifications.

The SAE can take the following actions if it finds a disparity between the router and SAE configurations:

- The SAE takes the state of the session layer on the router to be correct and updates its local state to be consistent with the router. The SAE then sends stop events for all sessions where the corresponding provisioning in the router has been removed.
- The SAE takes its local state to be the correct state and updates the router to be consistent with its local state.
- The SAE does not solve the state discrepancy. It reports disparities through the SAE device driver event trap called `routerConfOutOfSynch` and through the info log.

Note that it is not possible to check the consistency of individual objects that the SAE provisions. Therefore, modifications to a provisioning object while the SAE is disconnected from the router cannot be detected.

**Related
Documentation**

- [Setting Up Periodic Configuration Checking \(SRC CLI\) on page 92](#)
- [Setting Up the SAE to Periodically Check Junos OS Configuration \(C-Web Interface\)](#)

Setting Up Periodic Configuration Checking (SRC CLI)

To configure the SAE to periodically check the configuration of the device running Junos OS:

1. From configuration mode, access the configuration statement that configures the configuration checking feature.

```
user@host# edit shared sae configuration driver junos configuration-checking
```

2. Specify when the SAE checks the router configuration.

```
[edit shared sae configuration driver junos configuration-checking]
user@host# set configuration-checking-schedule configuration-checking-schedule
```

3. Specify the action that the SAE takes when it detects disparities between the configuration of the SAE and the configuration on the router.

```
[edit shared sae configuration driver junos configuration-checking]
user@host# set configuration-checking-action enforce | synchronize | detect
```

4. (Optional) From operational mode, verify your configuration checking configuration.

```
[edit shared sae configuration driver junos configuration-checking]
user@host# show
configuration-checking-schedule "0 0 * * * * *";
configuration-checking-action synchronize;
```

Related Documentation

- [Setting Up the SAE to Periodically Check Junos OS Configuration \(C-Web Interface\)](#)
- [SAE Verification of Junos OS Configuration Changes on page 92](#)

Using SNMP to Retrieve Information from JunosE Routers and Devices Running Junos OSs (SRC CLI)

You can use SNMP to retrieve information from the router. For example, if you create a router initialization script that uses SNMP, you need to specify the SNMP communities that are on the router.

We recommend that you specify SNMP communities for each virtual router. (See [“Adding JunosE Routers and Virtual Routers \(SRC CLI\)” on page 54.](#)) You can also configure global default SNMP communities.

You can configure global default SNMP communities that are used if a VR does not exist on the router or if the community strings have not been configured for the VR.

Use the following configuration statements to configure global default SNMP communities:

```
shared sae configuration driver snmp {
  read-only-community-string read-only-community-string;
  read-write-community-string read-write-community-string;
}
```

To configure global default SNMP communities:

1. From configuration mode, access the configuration statements that configure default SNMP communities.

```
user@host# edit shared sae configuration driver snmp
```

2. Configure the default SNMP community string used for read access to the router.

```
[edit shared sae configuration driver snmp]
```

```
user@host# set read-only-community-string read-only-community-string
```

3. Configure the default SNMP community string used for write access to the router.

```
[edit shared sae configuration driver snmp]  
user@host# set read-write-community-string read-write-community-string
```

4. (Optional) Verify your configuration.

```
[edit shared sae configuration driver snmp]  
user@host# show  
read-only-community-string *****;  
read-write-community-string *****;
```

Related Documentation

- [Using SNMP to Retrieve Information from Devices Running Junos OS and Other Network Devices](#)
- [Configuring Event Tracking for Junos LSPs \(SRC CLI\) on page 96](#)
- [Configuring the Device Running Junos OS to Apply Changes It Receives from the SAE on page 97](#)
- [Disabling Interactions Between the SAE and Devices Running Junos OS on page 97](#)
- [Monitoring Interactions Between the SAE and Devices Running Junos OS on page 98](#)

Specifying Router Initialization Scripts on the SAE (SRC CLI)

Use the following configuration statements to specify router initialization scripts for devices running Junos OS:

```
shared sae configuration driver scripts {  
  extension-path extension-path;  
  general general;  
  junos junos;  
}
```

To configure router initialization scripts for devices running Junos OS:

1. From configuration mode, access the configuration statements that configure router initialization scripts. In this sample procedure, the scripts are configured in the west-region group.

```
user@host# edit shared sae group west-region configuration driver scripts
```

2. Specify the router initialization script for devices running Junos OS.

```
[edit shared sae group west-region configuration driver scripts]  
user@host# set junos junos
```

3. Configure a router initialization script that can be used for all types of routers that the SRC module supports.

```
[edit shared sae group west-region configuration driver scripts]  
user@host# set general general
```

4. Configure a path to router initialization scripts that are not in the default location, `/opt/UMC/sae/lib`.

```
[edit shared sae group west-region configuration driver scripts]
user@host# set extension-path extension-path
```

5. (Optional) From operational mode, verify your router initialization script configuration.

```
[edit shared sae group west-region configuration driver scripts]
user@host# show
extension-path ;
junos iorPublisher;
```

Related Documentation

- [Specifying Initialization Scripts of Routers Running Junos OS on the SAE \(C-Web Interface\)](#)
- [Configuring Devices Running Junos OS to Interact with the SAE on page 95](#)
- [Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 63](#)

Configuring Devices Running Junos OS to Interact with the SAE

To configure the device running Junos OS to interact with the SAE:

1. Include the following statements at the **[edit system services service-deployment]** hierarchy level.

```
[edit system services service-deployment]
servers server-address {
  port port-number;
}
source-address source-address;
```

2. Use the following guidelines for the variables in these statements.

- **server-address** —Specifies the IP address of the host on which you install the SAE. Be sure this setting matches the corresponding value in the SAE configuration.
- **port-number**— Specifies the port number for the SAE. Be sure this setting matches the corresponding value in the SAE configuration.
- **source-address** —(Optional) Specifies the IP address of the source that sends traffic to the SAE.

Related Documentation

- [Specifying Router Initialization Scripts on the SAE \(SRC CLI\) on page 94](#)
- [Configuring the Device Running Junos OS to Apply Changes It Receives from the SAE on page 97](#)
- [Disabling Interactions Between the SAE and Devices Running Junos OS on page 97](#)
- [Monitoring Interactions Between the SAE and Devices Running Junos OS on page 98](#)

SAE Tracking for LSPs Configured on Devices Running Junos OS

- [SAE Tracking for LSPs Configured on Devices Running Junos OS Overview on page 96](#)
- [Configuring Event Tracking for Junos LSPs \(SRC CLI\) on page 96](#)

SAE Tracking for LSPs Configured on Devices Running Junos OS Overview

You can configure the SAE to track the status of LSPs that are configured on managed devices running Junos OS. Use LSP tracking with applications such as the sample IPTV application. This application uses LSP tracking to collect status information for LSPs that carry IPTV traffic from video servers to a network edge router in which user connections terminate.

LSP tracking can configure the system log on managed devices running Junos OS to send notification messages to the managing SAE when LSPs are created and removed, and when bandwidth allocation for an LSP changes. You can enable LSP tracking for all managed devices running Junos OS or a set of devices running Junos OS.

The SAE creates a pseudointerface when each LSP becomes active (that is, when the RPD_MPLS_LSP_UP syslog event is logged) to:

- Track session status by sending interface-tracking plug-in events for each pseudointerface.
- Create subscriber sessions for the pseudo-interfaces.

The SAE does not support policy installation, including default policies, through an LSP pseudointerface.

Configuring Event Tracking for Junos LSPs (SRC CLI)

Configure event tracking for Junos LSPs to provide information to an application, such as the sample IPTV application, that needs information about LSP status.

To configure LSP tracking:

1. From configuration mode, access the configuration statement that specifies the configuration for tracking LSPs.

```
[edit]
user@host# edit shared sae configuration driver junos lsp-tracking
```

2. (Optional) Specify a regular expression to identify a set of LSP names. If you do not define an expression, the SAE tracks all LSPs.

```
[edit shared sae configuration driver junos lsp-tracking]
user@host# set match SRC123
```

3. (Optional) Specify the name of the file to store system log event messages (that provide information about LSP state changes in a device running Junos OS).

For example, to store messages in the junos-1 file:

```
[edit shared sae configuration driver junos lsp-tracking]
user@host# file junos-1
```


Configuring the Device Running Junos OS to Apply Changes It Receives from the SAE

To configure the device running Junos OS to receive configuration statements from the SAE and apply those statements to the configuration:

1. Create a configuration group called **sdx** that contains the configuration statements that the SAE sends to the device running Junos OS. To do so, include the **groups** statement at the **[edit]** level and specify the name **sdx**.

```
[edit]
groups {
  sdx;
}
```

2. Configure the device running Junos OS to apply these statements to the configuration. To do so, include the **apply-groups** statement at the **[edit]** level.

```
[edit]
set apply-groups sdx;
```

Related Documentation

- [Configuring Devices Running Junos OS to Interact with the SAE on page 95](#)
- [Disabling Interactions Between the SAE and Devices Running Junos OS on page 97](#)
- [Monitoring Interactions Between the SAE and Devices Running Junos OS on page 98](#)
- [SAE Verification of Junos OS Configuration Changes on page 92](#)

Disabling Interactions Between the SAE and Devices Running Junos OS

To disable the SRC software process, enter the following command:

```
root@ui1#set system processes service-deployment disable
root@ui1# commit
```

When you disable the SRC software process, it is still available on the device running Junos OS.

To reenable the SRC software process, enter the following command:

```
root@ui1# delete system processes service-deployment disable
root@ui1# commit
```

The SRC software process attempts to reconnect the device running Junos OS to the SAE.

Related Documentation

- [Configuring Devices Running Junos OS to Interact with the SAE on page 95](#)
- [Configuring the Device Running Junos OS to Apply Changes It Receives from the SAE on page 97](#)
- [Monitoring Interactions Between the SAE and Devices Running Junos OS on page 98](#)

Monitoring Interactions Between the SAE and Devices Running Junos OS

Purpose Monitor the connection between the SAE and a device running Junos OS.

Action Use the following command on devices running Junos OS to monitor the connection between the device running Junos OS and the SAE.

```
root@ui1>  
show system services service-deployment
```

```
Connected to 172.17.20.151 port 3333 since 2004-02-06 14:50:31 PST  
Keepalive settings: Interval 15 seconds  
Keepalives sent: 100, Last sent: 6 seconds ago  
Notifications sent: 0  
Last update from peer: 00:00:06 ago
```

You can also monitor the interactions between the SRC software and devices running Junos OS in the log files for the SAE and in the log files generated by the SRC software process on the device running Junos OS.

- Related Documentation**
- [Configuring Devices Running Junos OS to Interact with the SAE on page 95](#)
 - [Configuring the Device Running Junos OS to Apply Changes It Receives from the SAE on page 97](#)
 - [Disabling Interactions Between the SAE and Devices Running Junos OS on page 97](#)
 - [Logging for SRC Components Overview](#)
 - For information about configuring logging on devices running Junos OS, see the *Junos OS System Basics Configuration Guide*.

Troubleshooting Problems Between the SRC module and Device Drivers Running Junos OS

- [Troubleshooting Problems with the SRC Software Process on page 98](#)
- [Viewing the State of Device Drivers Running Junos OS \(SRC CLI\) on page 99](#)
- [Viewing Statistics for Specific Device Drivers Running Junos OS \(SRC CLI\) on page 100](#)
- [Viewing Statistics for All Device Drivers Running Junos OS \(SRC CLI\) on page 101](#)
- [Viewing the State of Device Drivers Running Junos OS \(C-Web Interface\) on page 101](#)
- [Viewing Statistics for Specific Device Drivers Running Junos OS \(C-Web Interface\) on page 102](#)
- [Viewing Statistics for All Device Drivers Running Junos OS \(C-Web Interface\) on page 102](#)

Troubleshooting Problems with the SRC Software Process

Problem The SRC process on a device running Junos OS is not working as expected.

Solution Review the log files for the SAE and the log files generated by the SRC software process on the router. If the log files indicate that the SRC software process on the device running Junos OS is not responding:

1. Look at the status of the process on the device running Junos OS.

```
root@ui1>show system services service-deployment
Connected to 172.17.20.151 port 3333 since 2004-02-06 14:50:31 PST
Keepalive settings: Interval 15 seconds
Keepalives sent: 100, Last sent: 6 seconds ago
Notifications sent: 0
Last update from peer: 00:00:06 ago
```

2. If you see the message “error: the service-deployment subsystem is not running,” reenable the SRC software process. See [“Disabling Interactions Between the SAE and Devices Running Junos OS” on page 97.](#)
3. If the process is already enabled, review the configurations of the router and the SAE in the directory, and fix any problems.
4. Restart the SRC software process on the router.

```
root@ui1>restart service-deployment
```

The SAE synchronizes with the SRC software process and deletes unnecessary data from the router.

If deleting parts of the SRC data on a device running Junos OS fails to solve problems, delete all the SRC data and restart the SRC software process. To do so:

1. Delete all SRC interfaces and services.

```
delete groups sdx
root@ui1#commit
```

2. Restart the SRC software process on the router.

```
root@ui1>restart service-deployment
```

Viewing the State of Device Drivers Running Junos OS (SRC CLI)

Purpose Display the state of drivers running Junos OS.

Action Use the following operational mode command:

```
show sae drivers <device-name device-name > < (brief) > <maximum-results
maximum-results >
```

For example:

```
user@host> show sae drivers device-name default@router
Driver running Junos OS
Device name                default@router
Device type                device running junos OS
Device IP                  /10.10.6.113:1879
Local IP                   10.10.6.113
TransportRouter
```

```
Device version                8.2R1.7
Start time                    Thu Mar 08 21:00:50 UTC 2007
Number of notifications       0
Number of processed added     0
Number of processed changed   0
Number of processed deleted   0
Number of provisioning attempt 0
Number of provisioning attempt failed 0
Device type                   JunosRouterDriver
Job queue size                0
Number of SAP                 3
Number of PAP                 0
Number of active user sessions 100
Number of active service sessions 0
Start time                    Thu Mar 08 21:00:55 UTC 2007
End time                      Thu Mar 08 21:00:55 UTC 2007
Transaction Manager
Transaction queue size 0
Router name                    default@troll
Session Store Info
Session Store Status          sessionsCollected
Status Last Update Time       Mon Jul 29 10:26:26 UTC 2013
Current Usage Ratio           1.0
Last Modified Time            Size(KB) Name LiveSessionsSize(KB)
Mon Jul 29 10:27:05 UTC 2013  117.9 storeOps_1_1 117.9
```

Viewing Statistics for Specific Device Drivers Running Junos OS (SRC CLI)

Purpose Display statistics for a specific device driver running Junos OS.

Action Use the following operational mode command:

```
show sae statistics device <name name> < (brief) >
```

For example:

```
user@host> show sae statistics device name default@jrouter
SNMP Statistics
Add notification handle time    7
Change notification handle time 0
Client ID                      default@troll
Delete notification handle time 0
Failover IP                    0.0.0.0
Failover port                  0
Handle message time            40
Job queue age                  0
Job queue time                  0
Number message send             3
Number of added jobs            0
Number of add notifications     0
Number of change notifications  0
Number of delete notifications  0
Number of managed interfaces    3
Number of message errors        0
Number of message timeouts      0
Number of removed jobs          0
Number of user session established 0
Number of user session removed  0
Router type                    Device running Junos OS
Up time                        7036120
```

```

Using failover server           false
Number of active user sessions 100
Number of active service sessions 0

```

Viewing Statistics for All Device Drivers Running Junos OS (SRC CLI)

Purpose Display SNMP statistics for all device drivers running Junos OS.

Action Use the following operational mode command:

```
show sae statistics device common junos
```

For example:

```

user@host> show sae statistics device common junos
SNMP Statistics
Driver type           Driver running Junos OS
Number of close requests 0
Number of connections accepted 0
Number of current connections 0
Number of open requests 0
Server address         0.0.0.0
Server port            3288
Time since last redirect 0

```

Viewing the State of Device Drivers Running Junos OS (C-Web Interface)

Problem Log files indicate a problem with a specific driver.

Solution Review the configuration of the associated device driver running Junos OS with C-Web:

1. Select **SAE** from the side pane, and click **Drivers**.

The Drivers pane appears.

2. In the Name of Device Driver box, enter a full or partial device driver name for which you want to display information, or leave the box blank to display all devices. Use the format:

```
default@<router name>
```

3. Select an output style from the Style list.
4. In the Maximum Results box, enter the maximum number of results that you want to receive.
5. Click **OK**.

The Drivers pane displays information about the device driver running Junos OS.

Viewing Statistics for Specific Device Drivers Running Junos OS (C-Web Interface)

Purpose View SNMP statistics about devices.

- Action** 1. Select **SAE** from the side pane, click **Statistics**, and then click **Device**.

The Device pane appears.

2. In the Device Name box, enter a full or partial device name for which you want to display information, or leave the box blank to display all devices.
3. Select an output style from the Style list.
4. Click **OK**.

The Device pane displays statistics for all devices.

Viewing Statistics for All Device Drivers Running Junos OS (C-Web Interface)

Purpose View SNMP statistics about specific devices.

- Action** 1. Select **SAE** from the side pane, click **Statistics**, click **Device**, and then click **Common**.

The Common pane appears.

The screenshot shows the Juniper SAE web interface. On the left is a navigation pane with a tree structure including Monitor, ACP, CLI, Component, Date, Disk, Interfaces..., JPS, NIC, NTP, Redirect Server, Route..., SAE (selected), Security, and System. The main content area is titled 'Common' and contains two input fields: 'Device Name' (a text box) and 'Type' (a dropdown menu). Below these fields are 'OK' and 'Reset' buttons. To the right of the 'Type' dropdown is a help text area that reads: 'Name of a device. Please enter: All or part of the device name. For JUNOS router drivers and PCMM drivers, use the format default@routerName. Display SNMP statistics for a specified device driver type. Choices: junos: Display SNMP statistics for JUNOS router drivers junose-cops: Display SNMP statistics for JUNOSe router drivers packetable-cops: Display SNMP statistics for PCMM device drivers proxy: Display SNMP statistics for third-party drivers'. At the top right of the main area, there are links for 'About', 'Refresh', and 'Logout'. The bottom of the page features a copyright notice: 'Copyright © 2007, Juniper Networks, Inc. All Rights Reserved. Trademark Notice. Privacy.' and the Juniper logo.

2. In the Device Name box, enter a full or partial device name for which you want to display information, or leave the box blank to display all devices.
3. Select the **junos** from the Type list:
4. Click **OK**.

The Common pane displays statistics for the specified device.

CHAPTER 8

Managing Junos DMI Devices Using the SRC Software

- [Managing DMI Devices on Routers Running Junos OS Using the SRC Software and Junos Space on page 105](#)
- [Managing DMI Devices Using the SRC Software and Junos Space Overview on page 106](#)
- [Summary of Tasks for Configuring the SRC Software to Manage DMI-Enabled Routers Running Junos OS \(SRC CLI\) on page 107](#)
- [Adding the Router Running Junos OS as a DMI Network Device \(SRC CLI\) on page 108](#)
- [Configuring the Junos DMI Driver \(SRC CLI\) on page 109](#)
- [Migrating from the Junos \(BEEP\) Driver to the Junos DMI Driver \(SRC CLI\) on page 110](#)

Managing DMI Devices on Routers Running Junos OS Using the SRC Software and Junos Space

For information about which devices running Junos OS and releases a particular SRC release supports, see the *SRC Release Notes*.

Using the SRC Device Management Interface (DMI) driver and Junos Space, the SRC software can manage DMI devices connected to routers running Junos OS. The SRC software communicates with Junos Space using the representational state transfer (REST) over HTTP(S), and Junos Space manages the router running Junos OS over the DMI. The SRC software recognizes and receives notifications for changes to DMI devices connected to the router, allowing you to offer dynamic services on those devices. In addition, you can define and automatically provision policies for DMI devices, provide per-subscriber accounting for services on DMI devices, and develop script services for service sessions residing on DMI-managed devices.

The router stores data about interfaces and services that the SAE manages in a configuration group called `sdx`. When you use the DMI driver, the SRC software automatically creates this configuration group on the router.

Related Documentation

- [Managing DMI Devices Using the SRC Software and Junos Space Overview on page 106](#)
- [Migrating from the Junos \(BEEP\) Driver to the Junos DMI Driver \(SRC CLI\) on page 110](#)

Managing DMI Devices Using the SRC Software and Junos Space Overview

Using the SRC Junos Device Management Interface (DMI) router driver and Junos Space, you can manage DMI-enabled routers running Junos OS. Junos Space provides the ability to manage all Junos devices that provide a DMI. Using the Junos Space GUI, you can discover and manage DMI devices. The SRC software uses the Junos Space REST API to configure, monitor, and synchronize with DMI devices.

The SRC Junos DMI router driver provides the integration between the SRC software and Junos Space to manage Junos devices using the Junos Space REST API. The SRC Junos DMI router driver is an alternative to the SRC Junos BEEP router driver implementation, which is obsolete and is not supported on all devices running Junos OS.

All currently supported BEEP features are available with the Junos DMI router driver, including stateless firewall filters, Cos and advanced services policies (stateful firewall and NAT). As with the current Junos (BEEP) router driver, script services that use the Junos XML management protocol command channel are also supported. All drivers configured within a single SRC host are connected to the same Junos Space cluster. The Junos DMI driver is independent of the BEEP driver. Both drivers can be active at the same time but cannot be connected to the same router running Junos OS.

To provide redundancy, you can configure multiple instances of the Junos DMI driver for the same router running Junos OS. Only one driver for a given device is active at the same time.

Like all SAE router drivers, the Junos DMI driver reacts to requests from the device that signals subscribers logging in and logging out. The driver publishes Interface Tracking events, performs interface classification to determine any default policies, and initiates SAE subscriber session login and logout processing. The driver can dynamically activate, modify, and deactivate policies for existing subscriber sessions, or terminate a subscriber session. The driver can synchronize the state of a single subscriber session or all sessions.

Configuration Overview

With the Junos (BEEP) driver, because the sdx daemon establishes the connection to the SRC software, you need to configure the SRC server on the device. You also need to create the sdx and sdx-sessions groups and add them to the apply-groups with the highest priority. However, the Junos DMI router driver initiates the connection to the Junos Space cluster and does not communicate with the router directly. As a result, no additional configuration is required on the Junos Space cluster, or on the router to specify the SRC server. For the groups and the apply-groups configuration, the Junos DMI router driver automatically configures the device.

The groups name under which you install the SRC policies is configurable. However, for backward compatibility with the Junos (BEEP) router driver, the default groups name is "sdx" and "sdx-sessions."

Redundancy

For redundancy, multiple SRC hosts can be configured in a community. The community manager appoints a master to become active. The active driver connects to the Junos Space cluster and manages the router. The standby driver does not connect to Junos Space, or send any configuration to the router unless it detects the failure of the master and switches over.

Selecting an active driver requires that the network be reachable between all drivers managing a particular router.

If a community member cannot reach its peers, it appoints the local driver as an isolated master. When connectivity is restored, multiple masters may be active. The following scheme is used to resolve this issue:

1. If a driver is appointed and it cannot connect to the Junos Space cluster that has active connections to its device, the driver shuts down.
2. If two masters are active at the same time, they send pings to each other. In this case, one of the masters will be demoted and the other performs a full synchronization.

The Juniper Networks database is used to look up the endpoint address of the peers, so the drivers must be configured to use a shared Juniper Networks database (for example, by configuring the local Juniper Networks database to participate in the same directory community).

Related Documentation

- [Summary of Tasks for Configuring the SRC Software to Manage DMI-Enabled Routers Running Junos OS \(SRC CLI\) on page 107](#)
- [Adding the Router Running Junos OS as a DMI Network Device \(SRC CLI\) on page 108](#)
- [Configuring the Junos DMI Driver \(SRC CLI\) on page 109](#)

Summary of Tasks for Configuring the SRC Software to Manage DMI-Enabled Routers Running Junos OS (SRC CLI)

To configure the SRC software to manage DMI-enabled routers running Junos OS through Junos Space:

1. Add the router running Junos OS as a DMI network device. See [“Adding the Router Running Junos OS as a DMI Network Device \(SRC CLI\)” on page 108](#).
2. Configure the SRC Junos DMI driver. See [“Configuring the Junos DMI Driver \(SRC CLI\)” on page 109](#).
3. Configure the session store feature for the Junos DMI driver. See [“Configuring the Session Store Feature \(SRC CLI\)” on page 31](#).

Related Documentation

- [Managing DMI Devices Using the SRC Software and Junos Space Overview on page 106](#)
- [Adding the Router Running Junos OS as a DMI Network Device \(SRC CLI\) on page 108](#)
- [Configuring the Junos DMI Driver \(SRC CLI\) on page 109](#)

Adding the Router Running Junos OS as a DMI Network Device (SRC CLI)

Use the following configuration statements to configure the SAE to manage the DMI-enabled router running Junos OS:

```
shared network device name {  
  description description;  
  management-address management-address;  
  device-type (junose | junos-ise | junos-ptsp | junos | junos-dmi | pcmm | third-party);  
  interface-classifier interface-classifier;  
}
```

To add the router running Junos OS as a DMI network device:

1. From configuration mode, access the configuration statements that configure network devices. You must specify the name of a device with lowercase characters. This procedure uses `r1-dmi` as the name of the router.

```
user@host# edit shared network device r1-dmi
```

2. (Optional) Add a description for the router.

```
[edit shared network device r1-dmi]  
user@host# set description description
```

3. (Optional) Add the IP address of the router.

```
[edit shared network device r1-dmi]  
user@host# set management-address management-address
```

4. (Optional) Specify the device type as DMI.

```
[edit shared network device r1-dmi]  
user@host# set device-type junos-dmi
```

5. Configure an interface classifier for the network device. For more information about interface classifiers, see the *SRC PE Subscribers and Subscriptions Guide*.

```
[edit shared network device r1-dmi]  
user@host# set interface-classifier interface-classifier
```

6. (Optional) Verify your configuration.

```
[edit shared network device r1-dmi]  
user@host# show  
description "Juniper Networks";  
management-address 10.10.8.27;  
device-type junos-dmi;  
interface-classifier {  
  rule rule-0 {  
    script #;  
  }  
}
```

- Related Documentation**
- [Managing DMI Devices Using the SRC Software and Junos Space Overview on page 106](#)
 - [Summary of Tasks for Configuring the SRC Software to Manage DMI-Enabled Routers Running Junos OS \(SRC CLI\) on page 107](#)
 - [Configuring the Junos DMI Driver \(SRC CLI\) on page 109](#)

Configuring the Junos DMI Driver (SRC CLI)

Use the following configuration statements to configure the SAE to manage DMI devices through Junos Space:

```
shared sae configuration driver junos-dmi {  
  junos-space-server-address junos-space-server-address;  
  junos-space-port-number junos-space-port-number;  
  junos-space-user-name junos-space-user-name;  
  junos-space-password junos-space-password;  
  junos-space-protocol (http | https);  
  apply-group-name apply-group-name;  
  sae-community-manager sae-community-manager;  
}
```

To configure the SAE to manage DMI devices through the Junos Space:

1. From configuration mode, access the configuration statements that configure the DMI device driver.

```
user@host# edit shared sae configuration driver junos-dmi
```

2. Specify the IP address of the Junos Space server that manages the routers.

```
[edit shared sae configuration driver junos-dmi]  
user@host# set junos-space-server-address junos-space-server-address
```

3. Specify the Junos Space port number.

```
[edit shared sae configuration driver junos-dmi]  
user@host# set junos-space-port-number junos-space-port-number
```

4. (Optional) Specify the protocol used to connect to Junos Space.

```
[edit shared sae configuration driver junos-dmi]  
user@host# set junos-space-protocol https
```

Where the protocol is one of the following:

- http
- https (default)

5. Specify the Junos Space username.

```
[edit shared sae configuration driver junos-dmi]  
user@host# set junos-space-user-name junos-space-user-name
```

6. Specify the password to authenticate with Junos Space.

```
[edit shared sae configuration driver junos-dmi]
user@host# set junos-space-password junos-space-password
```

7. Specify the name of the group on the router running Junos OS in which provisioning objects are stored. This name must match the name configured on the router.

```
[edit shared sae configuration driver junos-dmi]
user@host# set apply-group-name apply-group-name
```

8. Specify the name of the community manager that manages DMI driver communities. Active SAEs are selected from this community.

```
[edit shared sae configuration driver junos-dmi]
user@host# set sae-community-manager sae-community-manager
```

9. (Optional) Verify your configuration.

```
[edit shared sae configuration driver junos-dmi]
user@host# show
junos-space-password *****;
junos-space-port-number 8080;
junos-space-protocol https;
junos-space-server-address 10.1.2.3;
junos-space-user-name user1;
apply-group-name sdx;
sae-community-manager sae_mgr;
}
```

**Related
Documentation**

- [Managing DMI Devices Using the SRC Software and Junos Space Overview on page 106](#)
- [Summary of Tasks for Configuring the SRC Software to Manage DMI-Enabled Routers Running Junos OS \(SRC CLI\) on page 107](#)
- [Adding the Router Running Junos OS as a DMI Network Device \(SRC CLI\) on page 108](#)

Migrating from the Junos (BEEP) Driver to the Junos DMI Driver (SRC CLI)

Migrating active sessions is not supported when upgrading to SRC Release 4.2 from previous releases of SRC software. This applies when running BEEP, or when migrating from the BEEP driver to DMI driver.

**Related
Documentation**

- [Managing DMI Devices Using the SRC Software and Junos Space Overview on page 106](#)

PART 3

Using Network Devices in the SRC Network

- [Integrating Third-Party Network Devices into the SRC Network \(SRC CLI\) on page 113](#)

CHAPTER 9

Integrating Third-Party Network Devices into the SRC Network (SRC CLI)

- [Integrating Network Devices into the SRC Network Overview on page 113](#)
- [Logging In Subscribers and Creating Sessions on page 115](#)
- [Configuration Tasks for Integrating Third-Party Network Devices \(SRC CLI\) on page 118](#)
- [Setting Up Script Services on page 119](#)
- [Adding Objects for Network Devices \(SRC CLI\) on page 119](#)
- [Adding Virtual Router Objects \(SRC CLI\) on page 120](#)
- [Setting Up SAE Communities \(SRC CLI\) on page 122](#)
- [Configuring SAE Properties for the Event Notification API \(SRC CLI\) on page 124](#)
- [Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 125](#)
- [Copying Initialization Scripts to the C Series Controller on page 128](#)
- [Specifying Initialization Scripts on the SAE \(SRC CLI\) on page 128](#)
- [Using SNMP to Retrieve Information from Network Devices on page 129](#)
- [Configuring Global SNMP Communities in the SRC Software \(SRC CLI\) on page 129](#)
- [Using the NIC Resolver in Environments That Have Third-Party Devices \(SRC CLI\) on page 130](#)

Integrating Network Devices into the SRC Network Overview

You can integrate third-party routers and other network devices into your SRC network. The SAE provides a driver that you can use to integrate the SAE with a third-party device. This device driver uses the session store to store and replicate subscriber and service session data within a community of SAEs.

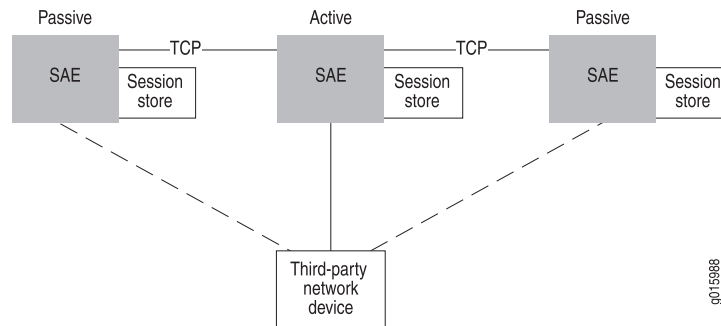
To log in subscribers to the SAE, you use assigned IP subscribers or event notification from an IP address manager.

To activate services and provision policies on the device, you use script services. You can also activate aggregate services for subscribers. However, you cannot activate normal services that require policies to be provisioned on the device.

SAE Communities

For SAE redundancy in an SRC network, you can have a community of two or more SAEs. SAEs in a community are given the role of either active SAE or passive SAE. The active SAE manages the connection to the network device and keeps session data up to date within the community. [Figure 7 on page 114](#) shows a typical SAE community.

Figure 7: SAE Community



When an SAE starts, it negotiates with other SAEs to determine which SAE controls the network device. The SAE community manager and members of the community select the active SAE.

A passive SAE needs to take over as active SAE in any of the following cases:

- The active SAE shuts down. In this case, the active SAE notifies the passive SAEs, and one of the passive SAEs takes over as active SAE.
- A passive SAE does not receive a keepalive message from the active SAE within the keepalive interval. In this case, the passive SAE attempts to become the active SAE.

Storing Session Data

To aid in recovering from an SAE failover, the SAE stores subscriber and service session data. When the SAE manages a network device, session data is stored in the SAE host's file system. The SRC component that controls the storage of session data on the SAE is called the session store. The session store queues data and then writes the data to session store files on the SAE host's disk. Once the data is written to disk, it can survive a server reboot.

For more information, see [“Storing Subscriber and Service Session Data” on page 29](#).

Using Script Services to Provision Third-Party Devices

You use script services to activate services and provision policies on third-party network devices. A script service is a service into which you can insert or reference a script. You write a script that will activate services and provision policies on the third-party device, and then you insert the script into the script service or reference the script in the service. When the SAE activates a service, it runs the script. The script provisions policies on the device using a means that the device supports. You can also include an interface in the

script that causes the SAE to send authentication and tracking events when it activates, modifies, or deactivates a script service session.

The SAE core API includes two interfaces for creating a script:

- **ScriptService**—Defines a service provider interface (SPI) that the script service must implement. The implementation of the ScriptService interface activates, modifies, or deactivates the service.
- **ServiceSessionInfo**—Provides a callback interface into the SAE and provides information about the service session to the script service.

For information about the ScriptService interface and the ServiceSessionInfo interface, see the script service documentation in the SAE core API documentation on the Juniper Networks Web site at

<http://www.juniper.net/techpubs/software/management/src/api-index.html>

You can write the script in Java or Jython.

**Related
Documentation**

- [Logging In Subscribers and Creating Sessions on page 115](#)
- [Configuration Tasks for Integrating Third-Party Network Devices \(SRC CLI\) on page 118](#)
- *Adding Objects for Network Devices (C-Web Interface)*
- *Setting Up SAE Communities (C-Web Interface)*
- *Configuring the SAE Community Manager*

Logging In Subscribers and Creating Sessions

You can use two mechanisms to obtain subscriber address requests and other information and to set up a pseudointerface on the network device. (You must choose one mechanism; you cannot mix them.)

1. **Assigned IP subscriber.** The SAE learns about a subscriber through subscriber-initiated activities, such as activating a service through the portal or through the SRC Web Services Gateway).

With this method, you use the assigned IP subscriber login type along with the network interface collector (NIC) to map IP addresses to the SAE.

2. **Event notification from an IP address manager.** The SAE learns about subscribers through notifications from an external IP address manager, such as a DHCP server or a RADIUS server.

With this method, you use the event notification application programming interface (API). The API provides an interface to the IP address manager, and lets the IP address manager notify the SAE of events such as IP address assignments.

Assigned IP Subscribers

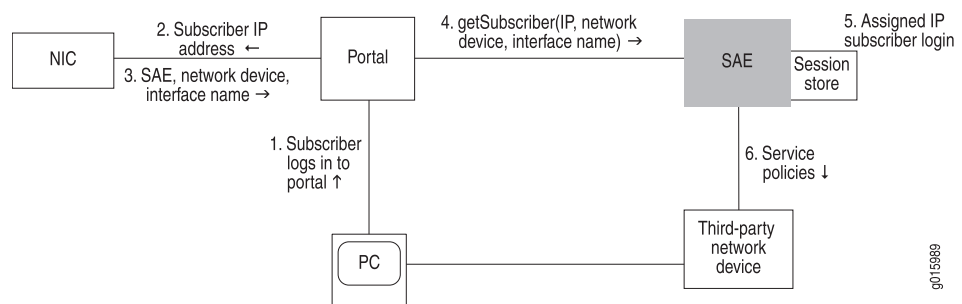
With the assigned IP subscriber method of logging in subscribers and creating sessions, the SRC module uses IP address pools along with network information collector (NIC)

resolvers to provide mapping of IP addresses to SAEs. You configure the static address pools or dynamically discovered address pools in the virtual router configuration for a network device. These pools are published in the NIC. The NIC maps subscriber IP addresses in requests received through the portal or SRC Web Services Gateway to the SAE that currently manages that network device.

Login Interactions with Assigned IP Subscribers

This section describes login interactions for assigned IP subscribers. In the example shown in [Figure 8 on page 116](#), the subscriber activates a service through a portal. You could also have the subscriber activate a service through the SRC Web Services Gateway.

Figure 8: Login Interactions with Assigned IP Subscribers



The sequence of events for logging in and creating sessions for assigned IP subscribers is:

1. The subscriber logs in to the portal.
2. The portal sends the subscriber's IP address to the NIC.
3. Based on the IP address, the NIC looks up the subscriber's SAE, network device, and interface name, and returns this information to the portal.
4. The portal sends a get Subscriber message to the SAE. The message includes the subscriber's IP address, network device, and interface name.
5. The SAE creates an assigned IP subscriber and performs a subscriber login. Specifically, it:
 - a. Runs the subscriber classification script with the IP address of the subscriber. (Use the ASSIGNEDIP login type in subscriber classification scripts.)
 - b. Loads the subscriber profile.
 - c. Runs the subscriber authorization plug-ins.
 - d. Runs the subscriber tracking plug-ins.
 - e. Creates a subscriber session and stores the session data in the session store file.
6. The SAE pushes service policies for the subscriber session to the network device.

Because the SAE is not notified when the subscriber logs out, the assigned IP idle timer begins when no service is active. The SAE removes the interface subscriber session when the timeout period ends.

Event Notification from an IP Address Manager

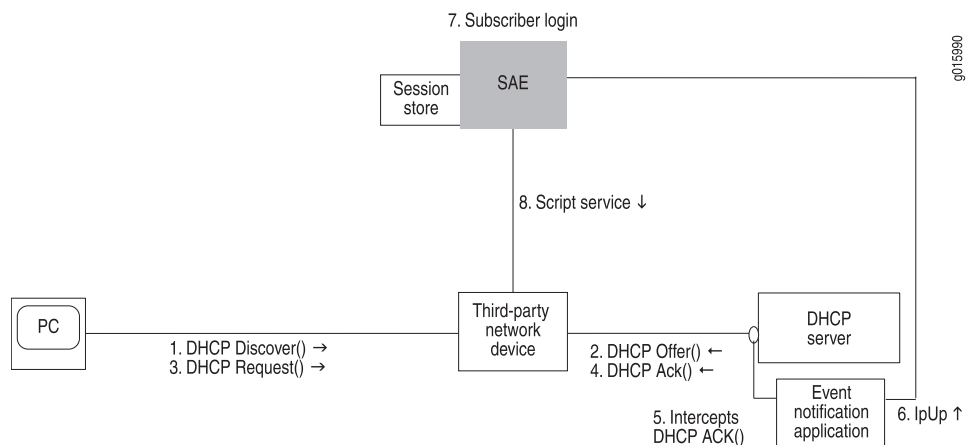
With the event notification method of logging in subscribers and creating subscriber sessions, the subscriber logs in to the network device and obtains an IP address through an address server, usually a DHCP server. The SAE receives notifications about the subscriber, such as the subscriber's IP address, from an event notification application that is installed on the DHCP server.

To use this method of logging in subscribers, you can use the event notification API to create the application that notifies the SAE when events occur between the DHCP server and the network device. You can also use Monitoring Agent, a sample application that was created with the event notification API and that monitors DHCP or RADIUS messages for DHCP or RADIUS servers. See the *SRC PE Sample Applications Guide*.

Login with Event Notification

This section describes login interactions by means of event notifications.

Figure 9: Login Interactions with Event Notification Application



The sequence of events for logging in subscribers and creating sessions is:

1. The DHCP client in the subscriber's computer sends a DHCP discover request to the DHCP server.
2. The DHCP server sends a DHCP offer to the subscriber's DHCP client.
3. The DHCP client sends a DHCP request to the DHCP server.
4. The DHCP server acknowledges the request by sending a DHCP Ack message to the DHCP client.
5. The event notification application that is running on the DHCP server intercepts the DHCP Ack message.
6. The event notification application sends an ipUp message to the SAE that notifies the SAE that an IP address is up.
7. The SAE performs a subscriber login. Specifically, it:

- a. Runs the subscriber classification script.
 - b. Loads the subscriber profile.
 - c. Runs the subscriber authorization plug-ins.
 - d. Runs the subscriber tracking plug-ins.
 - e. Creates a subscriber session and stores the session in the session store file.
8. The SAE can start script services.

The ipUp event should be sent with a timeout set to the DHCP lease time. The DHCP server sends an ipUp event for each Ack message sent to the client. The SAE restarts the timeout each time it receives an ipUp event.

If the client explicitly releases the DHCP address (that is, it sends a DHCP release event), the DHCP server sends an ipDown event. If the client does not renew the address, the lease expires on the DHCP server and the timeout expires on the SAE.

Related Documentation

- [Integrating Network Devices into the SRC Network Overview on page 113](#)
- [Using the NIC Resolver in Environments That Have Third-Party Devices \(C-Web Interface\)](#)
- [Configuration Tasks for Integrating Third-Party Network Devices \(SRC CLI\) on page 118](#)
- [Configuring SAE Properties for the Event Notification API \(SRC CLI\) on page 124](#)
- [Adding Objects for Network Devices \(SRC CLI\) on page 119](#)
- [Setting Up Script Services on page 119](#)

Configuration Tasks for Integrating Third-Party Network Devices (SRC CLI)

To integrate third-party devices into your SRC network, complete the following tasks:

- Write a script and add a script service that references the script.
See [“Setting Up Script Services” on page 119](#).
- Add objects for the devices.
See [“Adding Objects for Network Devices \(SRC CLI\)” on page 119](#).
- Configure an SAE community.
See [“Setting Up SAE Communities \(SRC CLI\)” on page 122](#).
- (Optional) Configure SAE properties for the Event Notification API if you are using the event notification method to log in subscribers.
See [“Configuring SAE Properties for the Event Notification API \(SRC CLI\)” on page 124](#).
- Configure the session store.
See [“Storing Subscriber and Service Session Data” on page 29](#).
- If you are using the event notification method to log in subscribers, integrate the SAE with an IP address manager. There are two ways to do so:

- Use the event notification API to create an application that notifies the SAE when events occur between the DHCP server and the network device.

See the event notification API documentation in the SAE CORBA remote API documentation on the Juniper Networks Web site at <http://www.juniper.net/techpubs/software/management/src/api-index.html>

- Use Monitoring Agent, a sample application that was created with the event notification API and that monitors DHCP or RADIUS messages for DHCP or RADIUS servers.

See the *SRC PE Sample Applications Guide*.

Related Documentation

- *Configuration Tasks for Integrating Third-Party Network Devices (C-Web Interface)*
- [Integrating Network Devices into the SRC Network Overview on page 113](#)
- [Logging In Subscribers and Creating Sessions on page 115](#)

Setting Up Script Services

To set up script services:

1. Write a script that implements the ScriptService interface, a service provider interface (SPI) for the SAE.

See *Customizing Service Implementations*.

See the script service documentation in the SAE core API documentation on the Juniper Networks Web site at

<http://www.juniper.net/techpubs/software/management/src/api-index.html>

2. Add a script service that references the script.

See *SRC Script Services Overview*.

Related Documentation

- [Configuration Tasks for Integrating Third-Party Network Devices \(SRC CLI\) on page 118](#)
- [Copying Initialization Scripts to the C Series Controller on page 128](#)
- [Integrating Network Devices into the SRC Network Overview on page 113](#)
- [Logging In Subscribers and Creating Sessions on page 115](#)
- *Setting Up SAE Communities (C-Web Interface)*

Adding Objects for Network Devices (SRC CLI)

For each network device that the SAE manages, add a router object and virtual router object.

Use the following configuration statements to add a router object:

```
shared network device name {
```

```
description description;  
management-address management-address;  
device-type (junose| junos| pcmm| third-party);  
qos-profile [qos-profile...];  
}
```

To add a router object:

1. From configuration mode, access the statements that configure network devices. You must specify the name of a device with lowercase characters. This sample procedure uses proxy_device as the name of the router.

```
user@host# edit shared network device proxy_device
```

2. (Optional) Add a description for the router object.

```
[edit shared network device proxy_device]  
user@host# set description description
```

3. (Optional) Add the IP address of the router object.

```
[edit shared network device proxy_device]  
user@host# set management-address management-address
```

4. Set the type of device that you are adding to third-party.

```
[edit shared network device proxy_device]  
user@host# set device-type third-party
```

5. (Optional) Verify your configuration.

```
[edit shared network device proxy_device]  
user@host# show  
description "Third-party router";  
management-address 192.168.9.25;  
device-type third-party;  
interface-classifier {  
  rule rule-0 {  
    script #;  
  }  
}
```

Related Documentation

- [Adding Objects for Network Devices \(C-Web Interface\)](#)
- [Configuration Tasks for Integrating Third-Party Network Devices \(SRC CLI\) on page 118](#)
- [Adding Virtual Router Objects \(SRC CLI\) on page 120](#)
- [Integrating Network Devices into the SRC Network Overview on page 113](#)
- [Logging In Subscribers and Creating Sessions on page 115](#)

Adding Virtual Router Objects (SRC CLI)

Use the following configuration statements to add a virtual router:


```
shared network device name virtual-router name {
  sae-connection [sae-connection ...];
  snmp-read-community snmp-read-community;
  snmp-write-community snmp-write-community;
  scope [scope...];
  tracking-plug-in [tracking-plug-in...];
}
```

To add a virtual router:

1. From configuration mode, access the statements for virtual routers. You must specify the name of a device with lowercase characters. This sample procedure uses `proxy_device` as the name of the router object. For third-party devices, use the name default for the virtual router.

```
user@host# edit shared network device proxy_device virtual-router default
```

2. Specify the addresses of SAEs that can manage this router. This step is required for the SAE to work with the router.

```
[edit shared network device proxy_device virtual-router default]
user@host# set sae-connection [sae-connection ...]
```

To specify the active SAE and the redundant SAE, enter an exclamation point (!) after the hostname or IP address of the connected SAE. For example:

```
[edit shared network device proxy_device virtual-router default]
user@host# set sae-connection [sae1! sae2!]
```

3. (Optional) Specify an SNMP community name for SNMP read-only operations for this virtual router.

```
[edit shared network device proxy_device virtual-router default]
user@host# set snmp-read-community snmp-read-community
```

4. (Optional) Specify an SNMP community name for SNMP write operations for this virtual router.

```
[edit shared network device proxy_device virtual-router default]
user@host# set snmp-write-community snmp-write-community
```

5. (Optional) Specify service scopes assigned to this virtual router. The scopes are available for subscribers connected to this virtual router for selecting customized versions of services.

```
[edit shared network device proxy_device virtual-router default]
user@host# set scope [scope ...]
```

6. (Optional) Specify the plug-ins that track interfaces that the SAE manages on this virtual router.

```
[edit shared network device proxy_device virtual-router default]
user@host# set tracking-plug-in [tracking-plug-in ...]
```

7. (Optional) Verify your configuration.

```
[edit shared network device proxy_device virtual-router default]
user@host# show
sae-connection 10.8.221.45;
snmp-read-community *****;
snmp-write-community *****;
scope POP-Toronto;
tracking-plugin flexRadius;
```

**Related
Documentation**

- [Adding Objects for Network Devices \(C-Web Interface\)](#)
- [Adding Objects for Network Devices \(SRC CLI\) on page 119](#)
- [Integrating Network Devices into the SRC Network Overview on page 113](#)

Setting Up SAE Communities (SRC CLI)

Tasks to configure SAE communities are:



NOTE: If there is a firewall in the network, configure the firewall to allow SAE messages through.

1. [Configuring the SAE Community Manager on page 122](#)
2. [Specifying the Community Manager in the SAE Device Driver on page 123](#)

Configuring the SAE Community Manager

Use the following configuration statements to configure the SAE community manager that manages third-party network device communities:

```
shared sae configuration external-interface-features name CommunityManager {
  keepalive-interval keepalive-interval ;
  threads threads ;
  acquire-timeout acquire-timeout ;
  blackout-time blackout-time ;
}
```

To configure the community manager:

1. From configuration mode, access the configuration statements for the community manager. In this sample procedure, sae_mgr is the name of the community manager.

```
user@host# edit shared sae configuration external-interface-features sae_mgr
CommunityManager
```

2. Specify the interval between keepalive messages sent from the active SAE to the passive members of the community.

```
[edit shared sae configuration external-interface-features sae_mgr CommunityManager]
user@host# set keepalive-interval keepalive-interval
```

3. Specify the number of threads that are allocated to manage the community. You generally do not need to change this value.

```
[edit shared sae configuration external-interface-features sae_mgr CommunityManager]
user@host# set threads threads
```

4. Specify the amount of time an SAE waits for a remote member of the community when it is acquiring a distributed lock. You generally do not need to change this value.

```
[edit shared sae configuration external-interface-features sae_mgr CommunityManager]
user@host# set acquire-timeout acquire-timeout
```

5. Specify the amount of time that an active SAE must wait after it shuts down before it can try to become the active SAE of the community again.

```
[edit shared sae configuration external-interface-features sae_mgr CommunityManager]
user@host# set blackout-time blackout-time
```

6. (Optional) Verify the configuration of the SAE community manager.

```
[edit shared sae configuration external-interface-features sae_mgr
CommunityManager]
user@host# show
CommunityManager {
  keepalive-interval 30;
  threads 5;
  acquire-timeout 15;
  blackout-time 30;
}
```

Specifying the Community Manager in the SAE Device Driver

Use the following configuration statements to specify the community manager in the SAE device driver.

```
shared sae configuration driver third-party {
  sae-community-manager sae-community-manager;
}
```

To specify the community manager:

1. From configuration mode, access the configuration statements for the third-party device driver.

```
user@host# edit shared sae configuration driver third-party
```

2. Specify the name of the community manager.

```
[edit shared sae configuration driver third-party]
user@host# set sae-community-manager sae-community-manager
```

3. (Optional) Verify the configuration of the third-party device driver.

```
[edit shared sae configuration driver third-party]
user@host# show
sae-community-manager sae_mgr;
```

- Related Documentation**
- [Setting Up SAE Communities \(C-Web Interface\)](#)
 - [Configuration Tasks for Integrating Third-Party Network Devices \(SRC CLI\) on page 118](#)
 - [Configuring the SAE Community Manager](#)
 - [Integrating Network Devices into the SRC Network Overview on page 113](#)
 - [Logging In Subscribers and Creating Sessions on page 115](#)

Configuring SAE Properties for the Event Notification API (SRC CLI)

Use the following configuration statements to configure properties for the Event Notification API:

```
shared sae configuration external-interface-features name EventAPI {  
    retry-time retry-time ;  
    retry-limit retry-limit ;  
    threads threads ;  
}
```

To configure properties for the Event Notification API:

1. From configuration mode, access the configuration statements for the Event Notification API. In this sample procedure, *west-region* is the name of the SAE group, and *event_api* is the name of the Event API configuration.

```
user@host# edit shared sae group west-region configuration  
external-interface-features event_api EventAPI
```

2. Specify the amount of time between attempts to send events that could not be delivered.

```
[edit shared sae group west-region configuration external-interface-features event_api  
EventAPI]  
user@host# set retry-time retry-time
```

3. Specify the number of times an event fails to be delivered before the event is discarded.

```
[edit shared sae group west-region configuration external-interface-features event_api  
EventAPI]  
user@host# set retry-limit retry-limit
```

4. Specify the number of threads allocated to process events.

```
[edit shared sae group west-region configuration external-interface-features event_api  
EventAPI]  
user@host# set threads threads
```

5. (Optional) Verify the configuration of the Event Notification API properties.

```
[edit shared sae group west-region configuration external-interface-features  
event_api EventAPI]  
user@host# show  
EventAPI {  
    retry-time 300;
```

```

    retry-limit 5;
    threads 5;
}

```

Related Documentation

- *Using the SAE in a PCMM Environment*
- *Configuring SAE Properties for the Event Notification API (C-Web Interface)*
- *Initially Configuring the SAE*
- *Configuring the SAE to Manage PCMM Devices (SRC CLI)*

Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers

When the SAE establishes a connection with a router or network device, it can run an initialization script to customize the setup of the connection. These initialization scripts are run when the connection between a router or network device and the SAE is established and again when the connection is dropped.

We provide the `iorPublisher` script in the `/opt/UMC/sae/lib` folder. The `iorPublisher` script publishes the interoperable object reference (IOR) of the SAE in the directory so that a NIC can associate a router with an SAE.

For JunosE VRs that supply IP addresses from a local pool, a router initialization script is provided that identifies which VR supplies each IP pool and writes the information to the configuration. The SAE runs the script only when a COPS connection is established to the JunosE router. Consequently, if you modify information about IP pools on a VR after the COPS connection is established, the SAE will not automatically register the changes, and you must update the configuration.

For Junos (only junos-ise device) virtual routers that supply IP addresses from a local pool, a router initialization script is provided to get the IP pool information from Junos router and update it in LDAP.

[Table 5 on page 63](#) describes the router initialization scripts that we provide with the SRC software in the `/opt/UMC/sae/lib` folder.

Table 7: Router Initialization Scripts

Script Name	Function	When to Use Script
<code>iorPublisher</code>	Publishes the IOR of the SAE into an internal part of the shared configuration so that a NIC can associate a router with an SAE.	<p>Use with JunosE routers that do not supply IP addresses from local pools, and with devices running Junos OS.</p> <p>Use with all devices running Junos OS.</p> <p>Use with third-party network devices.</p>

Table 7: Router Initialization Scripts (*continued*)

Script Name	Function	When to Use Script
poolPublisher	Publishes the IOR of the SAE and local IP address pools in the directory so that a NIC can associate a router with an SAE and resolve the IP-to-SAE mapping.	Use with JunosE virtual routers that supply IP addresses from local pools.
isePoolPublisher	Publishes the local IP address pools in the directory so that a NIC can associate a router with an SAE and resolve the IP-to-SAE mapping.	Use with Junos (only junos-ise device) virtual routers that supply IP addresses from local pools.

Interface Object Fields

Router initialization scripts are written in the Python programming language (www.python.org) and executed in the Jython environment (www.jython.org).

Router initialization scripts interact with the SAE through an interface object called Ssp. The SAE exports a number of fields through the interface object to the script and expects the script to provide the entry point to the SAE.

[Table 6 on page 64](#) describes the fields that the SAE exports.

Table 8: Exported Fields

Ssp Attribute	Description
Ssp.properties	System properties object (class: java.util.Properties)—The properties should be treated as read-only by the script.
Ssp.errorLog	Error logger—Use the Ssp.errorLog.println (message) to send error messages to the log.
Ssp.infoLog	Info logger—Use the Ssp.infoLog.println (message) to send informational messages to the log.
Ssp.debugLog	Debug logger—Use the Ssp.debugLog.println (message) to send debug messages to the log.

The router initialization script must set the field Ssp.routerInit to a factory function that instantiates a router initialization object:

- <VRName>—Name of the virtual router in which the COPS client has been configured, format: virtualRouterName@RouterName
- <virtualIp>—Virtual IP address of the SAE (string, dotted decimal; for example: 192.168.254.1)
- <realIp>—Real IP address of the SAE (string, dotted decimal; for example, 192.168.1.20)

- <VRip>—IP address of the virtual router (string, dotted decimal)
- <transportVR>—Name of the virtual router used for routing the COPS connection, or None, if the COPS client is directly connected

The factory function must implement the following interface:

```
Ssp.routerInit(VRName,
virtualIp,
realIp,
VRip,
transportVR)
```

The factory function returns an interface object that is used to set up and tear down a connection for a given COPS server. A common case of a factory function is the constructor of a class.

The factory function is called directly after a COPS server connection is established. In case of problems, an exception should be raised that leads to the termination of the COPS connection.

Required Methods

Instances of the interface object must implement the following methods:

- *setup()*—Is called when the COPS server connection is established and is operational. In case of problems, an exception should be raised that leads to the termination of the COPS connection.
- *shutdown()*—Is called when the COPS server connection to the virtual router is terminated. This method should not raise any exceptions in case of problems.

Example: Router Initialization Script

The following script defines a router initialization class named *SillyRouterInit*. The interface class does not implement any useful functionality. The interface class just writes messages to the infoLog when the router connection is created or terminated.

```
class SillyRouterInit:
    def __init__(self, vrName, virtualIp, realIp, vrIp, transportVr):
        """ initialize router initialization object """
        self.vrName = vrName
        Ssp.infoLog.println("SillyRouterInit created")

    def setup(self):
        """ initialize connection to router """
        Ssp.infoLog.println("Setup connection to VR %(vrName)s" %
            vars(self))

    def shutdown(self):
        """ shutdown connection to router """
        Ssp.infoLog.println("Shutdown connection to VR %(vrName)s" %
            vars(self))

#
# publish interface object to Ssp core
#
Ssp.routerInit = SillyRouterInit
```

Related Documentation

- [How SNMP Obtains Information from Routers for the SRC Software on page 62](#)
- [Specifying JunosE Router Initialization Scripts on the SAE \(SRC CLI\) on page 66](#)
- [Accessing the Router CLI on page 69](#)
- [Viewing Statistics for Specific JunosE Device Drivers \(SRC CLI\) on page 74](#)
- [Troubleshooting Problems with Managing JunosE Routers on page 71](#)
- [Updating Local IP Address Pools for Junos Virtual Routers \(SRC CLI\) on page 83](#)
- [Updating Local IP Address Pools for JunosE Virtual Routers \(SRC CLI\) on page 67](#)

Copying Initialization Scripts to the C Series Controller

If you use a script that is not provided with the SRC module, you need to use the **file copy** command to copy your script to the C Series Controller. For example:

```
user@host> file copy ftp://user@myserver/routerinit.py /opt/UMC/sae/lib
Password:
```

Related Documentation

- [Specifying Initialization Scripts on the SAE \(SRC CLI\) on page 128](#)
- [Setting Up Script Services on page 119](#)
- [Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 63](#)

Specifying Initialization Scripts on the SAE (SRC CLI)

Use the following configuration statements to specify initialization scripts for third-party devices:

```
shared sae configuration driver scripts {
    extension-path extension-path ;
    general general ;
}
```

To configure initialization scripts for third-party devices:

1. From configuration mode, access the configuration statements that configure initialization scripts.

```
user@host# edit shared sae configuration driver scripts
```

2. Specify the initialization script for third-party devices.

```
[edit shared sae configuration driver scripts]
user@host# set general general
```

3. Configure a path to scripts that are not in the default location, */opt/UMC/sae/lib*.

```
[edit shared sae configuration driver scripts]
user@host# set extension-path extension-path
```


4. (Optional) Verify your initialization script configuration.

```
[edit shared sae configuration driver scripts]
user@host# show
```

**Related
Documentation**

- [Specifying Initialization Scripts on the SAE \(C-Web Interface\)](#)
- [Copying Initialization Scripts to the C Series Controller on page 128](#)
- [Developing Router Initialization Scripts for Network Devices and Juniper Networks Routers on page 63](#)

Using SNMP to Retrieve Information from Network Devices

You can use SNMP to retrieve information from a network device. For example, if you create a script that uses SNMP, specify the SNMP communities that are on the network device.

To retrieve information:

- (Recommended) Specify SNMP communities for each virtual router object.
- Configure global default SNMP communities.

**Related
Documentation**

- [Adding Virtual Router Objects \(SRC CLI\) on page 120](#)
- [Adding Objects for Network Devices \(C-Web Interface\)](#)
- [Configuring Global SNMP Communities in the SRC Software \(SRC CLI\) on page 129](#)
- [Configuring Global SNMP Communities in the SRC Software \(C-Web Interface\)](#)

Configuring Global SNMP Communities in the SRC Software (SRC CLI)

You can configure global default SNMP communities that are used if a VR does not exist on the router or if the community strings have not been configured for the VR.

Use the following configuration statements to configure global default SNMP communities:

```
shared sae configuration driver snmp {
  read-only-community-string read-only-community-string;
  read-write-community-string read-write-community-string;
}
```

To configure global default SNMP communities:

1. From configuration mode, access the statements that configure default SNMP communities.


```
user@host# edit shared sae configuration driver snmp
```
2. Configure the default SNMP community string used for read access to the router.

```
[edit shared sae configuration driver snmp]
user@host# set read-only-community-string read-only-community-string
```

3. Configure the default SNMP community string used for write access to the router.

```
[edit shared sae configuration driver snmp]
user@host# set read-write-community-string read-write-community-string
```

4. (Optional) Verify your configuration.

```
[edit shared sae configuration driver snmp]
user@host# show
read-only-community-string *****;
read-write-community-string *****;
```

**Related
Documentation**

- [Using SNMP to Retrieve Information from JunosE Routers and Devices Running Junos OSs \(SRC CLI\) on page 93](#)
- [Configuring Global SNMP Communities in the SRC Software \(C-Web Interface\)](#)
- [Using SNMP to Retrieve Information from Network Devices on page 129](#)
- [How SNMP Obtains Information from Routers for the SRC Software on page 62](#)

Using the NIC Resolver in Environments That Have Third-Party Devices (SRC CLI)

If you are using the assigned IP subscriber method of logging in subscribers, and you are using the NIC to determine the subscriber's SAE, you need to configure a resolver on the NIC. The OnePopDynamicIp sample configuration data supports this scenario. The OnePopDynamicIp configuration supports one point of presence (POP) and provides no redundancy. The realm for this configuration accommodates the situation in which IP pools are configured locally on each virtual router object.

You can access the OnePopDynamicIp configuration in the SRC CLI.

**Related
Documentation**

- [Configuration Tasks for Integrating Third-Party Network Devices \(SRC CLI\) on page 118](#)
- [Integrating Network Devices into the SRC Network Overview on page 113](#)
- [Configuring the NIC \(SRC CLI\) on page 152](#)

PART 4

Locating Subscriber Management Information

- [Locating Subscriber Information with the NIC on page 133](#)
- [Configuring the NIC \(SRC CLI\) on page 149](#)
- [Obtaining Interface Configuration for OnePopStaticRouteIp or OnePopVrfIp on page 171](#)
- [Configuring Applications to Communicate with an SAE on page 185](#)
- [Configuring SRC Applications to Communicate with an SAE \(SRC CLI\) on page 187](#)
- [Developing Applications That Use NIC on page 195](#)
- [NIC Resolution Process on page 203](#)
- [NIC Configuration Scenarios on page 209](#)

CHAPTER 10

Locating Subscriber Information with the NIC

- [Locating Subscriber Management Information on page 133](#)
- [Mapping Subscribers to a Managing SAE on page 135](#)
- [High Availability for NIC on page 137](#)
- [Planning a NIC Implementation on page 139](#)
- [NIC Configuration Scenarios on page 140](#)
- [NIC Agents Used in the NIC Configuration Scenarios on page 144](#)
- [Router Initialization Scripts with NIC Configuration Scenarios on page 146](#)

Locating Subscriber Management Information

For services to be activated for a subscriber session, applications such as the SRC Volume-Tracking Application (SRC VTA), Dynamic Service Activator, Enterprise Manager Portal, or a residential portal need to locate the SAE that manages the subscriber.

The NIC is the component that locates which SAE manages a subscriber or an interface. The NIC uses information that identifies the subscriber or the interface to identify the managing SAE. A NIC is similar to a Domain Name System (DNS) in that a NIC processes resolution requests. Rather than translating hostnames to IP addresses and vice versa, the NIC resolves an identifier for a subscriber or an interface to a reference for the managing SAE.

The components that participate in this resolution are a NIC host and a NIC proxy, also called a NIC locator for particular applications. A NIC host processes resolution requests. A NIC proxy requests data resolution for an application. A NIC proxy is so-named because it requests information on behalf of an application. A NIC proxy and a NIC host communicate with each other through Common Object Request Broker Architecture (CORBA); NIC manages the CORBA interactions for you.

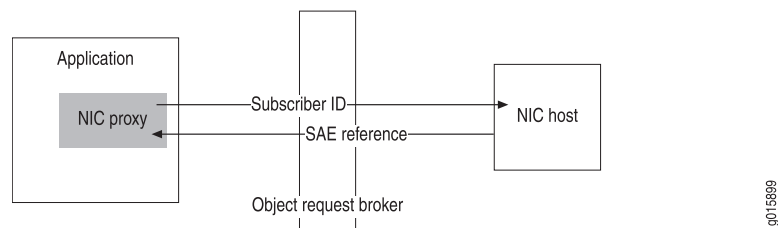
NIC can operate in a client/server mode or in a local host mode. In the client/server mode, a NIC host and NIC proxies can reside on different systems. In local host mode, a NIC host and NIC proxies reside in the same process on a machine.

NIC Client/Server Mode

In client/server mode, a NIC host is the server. A NIC proxy, which comprises libraries within an application that interacts with a NIC host, is the client.

Figure 10 on page 134 shows a NIC proxy running within an application and a NIC host running on a different machine. Both communicate through CORBA, with the NIC proxy providing an identifier for a subscriber and the NIC host returning a reference to the SAE that manages the subscriber.

Figure 10: Communication Between a NIC Proxy and a NIC Host in Client/Server Mode



NIC Local Host Mode

In local host mode, a Java application can include the libraries for a NIC host as well as NIC proxies. With this configuration, the NIC host and the NIC proxies communicate with each other within the same application. Because both components run within the same application, the application and the NIC host start and stop at the same time.

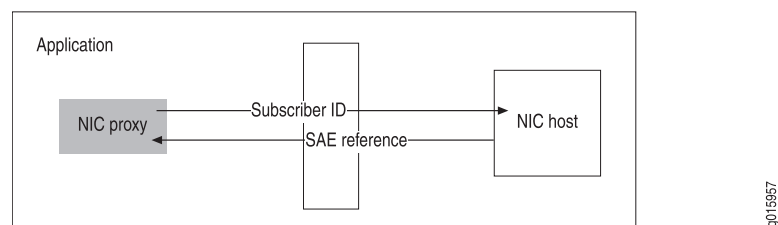
If an application uses a local NIC host, all NIC proxies for the application typically communicate with the local NIC host, but some of the NIC proxies can be configured to communicate with a NIC host that runs on another system.

When you use NIC in local host mode:

- You cannot use the C-Web interface to monitor or troubleshoot the local NIC host
- The NIC host runs all the resolvers and agents for the host on the local machine.
- Other NIC hosts cannot communicate with agents and resolvers that run in a local NIC host.

Figure 11 on page 134 shows a NIC proxy and a NIC host running within an application.

Figure 11: Communication Between a NIC Host and a NIC Proxy in Local Host Mode



- Related Documentation**
- [Mapping Subscribers to a Managing SAE on page 135](#)
 - [High Availability for NIC on page 137](#)
 - [NIC Proxy Configuration Overview on page 185](#)
 - [NIC Configuration Scenarios on page 140](#)
 - [NIC Agents Used in the NIC Configuration Scenarios on page 144](#)

Mapping Subscribers to a Managing SAE

A NIC collects information about the state of the network and can provide mapping from a specified type of network data, known as a *key*, to another type of network data, known as a *value*. Applications can use a NIC proxy to submit a key to a NIC host. The NIC host obtains a corresponding value from other components within NIC and returns it through the NIC proxy to the application. A typical use of a NIC is for a residential portal application to submit a subscriber's IP address and for the NIC to return the interoperable object reference (IOR) of the SAE managing that subscriber.

NIC Proxies and NIC Locators

Typically, an application supports one NIC proxy for each type of data request. A NIC proxy caches resolution results for a period of time so that it can resolve future requests without consulting the NIC host, thereby decreasing traffic between the NIC proxy and the NIC host. Applications that use NIC proxies communicate with the proxy to delete any invalid cache entries. Caching lets you optimize resolution performance for your network configuration and system resources.

You configure a NIC proxy when you configure that application. SRC applications such as the SRC VTA and Dynamic Service Activator contain NIC proxies. If you are writing an external application that will interact with a NIC, you must include NIC proxies in the application.

A NIC locator provides the same functionality as a NIC proxy; however, it runs as part of the NIC host. A NIC locator uses the NIC access interface module, a simple CORBA interface, to enable non-Java applications to interact with NIC. A NIC locator does not cache information.

For information about the NIC access interface module, see the API documentation on the Juniper Networks Web site at <http://www.juniper.net/techpubs/software/management/src/api-index.html>.

For more information about NIC proxies and NIC locators, see “[NIC Proxy Configuration Overview](#)” on page 185.

NIC Hosts

NIC hosts collect and store SRC information, and respond to requests from NIC proxies. The components in a NIC host that manage this process are:

- NIC agents—Collect data from SRC components, publish data, and make data available to NIC resolvers
- NIC resolvers—Process resolution requests

NIC Agents

NIC agents collect information about the state of the network from many data sources on the network. [Table 9 on page 136](#) describes the types of agents supplied with NIC.

Table 9: Types of NIC Agents

Type of Agent	Type of Information the Agent Makes Available
Consolidator agent	Summary information received from other agents.
Directory agent	Specified directory entries and changes to directory entries.
Properties agent	Information from a specified list of property file. Typically, you do not configure properties agents.
SAE client agent	SAEs managing a subscriber at resolution time.
SAE plug-in agent	Subscriber information and interface information for SAE-managed subscribers and interfaces.
SSR client agent	Subscriber information from the Session State Registrar.
XML agent	Information from a specified XML document. Typically, you do not configure XML agents.

NIC Resolvers

NIC resolvers manage information to resolve requests by:

- Receiving and storing information about the state of the network from components within NIC and other NIC resolvers
- Requesting information from NIC agents and other NIC resolvers
- Receiving requests from the NIC proxies or other NIC resolvers
- Processing requests and sending responses to the requesters

Related Documentation

- [Locating Subscriber Management Information on page 133](#)
- [Configuring a NIC Scenario \(SRC CLI\) on page 156](#)

High Availability for NIC

You can configure high availability for NIC when you use client/server mode with the NIC host and the NIC proxies running on different machines. NIC supports several mechanisms to maintain high availability. We recommend that you use NIC replication to keep a NIC configuration highly available. NIC replication uses groups of NIC hosts that share the same configuration for NIC resolutions to respond to resolution requests.

When you use NIC in local host mode, you do not need to configure redundancy for a NIC host, because the NIC host runs within the application.

High Availability in Existing NIC Configurations

If you have a previous NIC configuration, you may be using:

- NIC host redundancy, in which a set of NIC hosts provide redundancy

The SRC CLI does not support NIC host redundancy.

- Redundancy for SAE plug-in agents, in which a set of SAE plug-in agents provide redundancy

If you have an SAE plug-in agent that uses agent redundancy, enable state synchronization for the agent and use NIC replication. In SRC Release 1.0.0, configuration for SAE plug-in agent redundancy is discontinued.

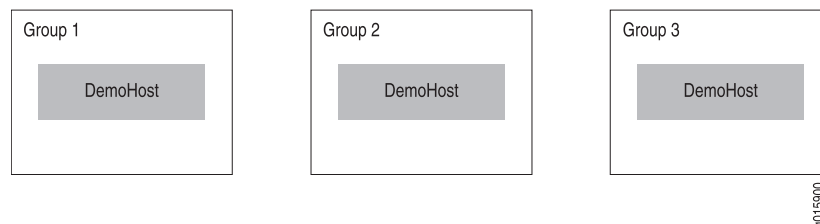
NIC Replication

NIC replication uses the concept of a group to identify a NIC host that has a particular configuration. A group contains one or more NIC hosts; each NIC host in a group is unique; for example, each NIC host could reside on a different system. A NIC proxy contacts specified groups that contain hosts with the same configuration to locate a managing SAE.

For example, a group might include the host DemoHost, but not two instances of DemoHost. Typically, each NIC host in a group is located in the same point of presence (POP). However, a machine can support only one NIC host. The SRC software stores groups in the directory in *ou=dynamicConfiguration*, *ou=Configuration*, *o=Management*, *o=umc*.

For example, [Figure 12 on page 137](#) shows three NIC groups with each group containing a NIC host that has the same configuration.

Figure 12: NIC Groups



Groups let you:

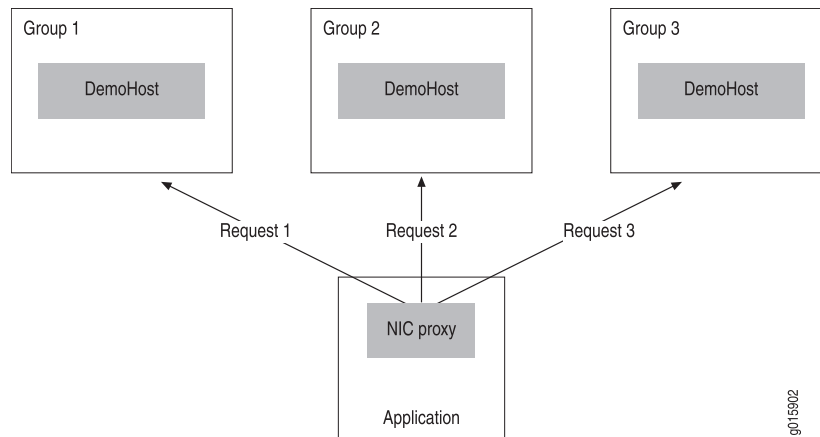
- Distribute network and processing load between two or more groups
- Provide failover protection if one group becomes unavailable

With NIC replication, a NIC proxy can contact multiple NIC hosts that are assigned to different groups. When a NIC proxy is configured to contact more than one group, the NIC configuration on a NIC host in each group should be equivalent—the NIC hosts should use the same configuration scenarios.

A NIC proxy selects a group by using the method specified in the configuration for the proxy; for example, the NIC proxy can randomly choose a group from a list. The NIC proxy then sends resolution requests to the corresponding host in that group. If a NIC proxy submits high numbers of resolution requests to the NIC host, you can configure the NIC proxy to randomly pick a NIC host or to pick a NIC host in a cyclic order to decrease the probability that one NIC host manages all the resolution requests.

[Figure 13 on page 138](#) shows resolution requests sent by means of a round-robin selection.

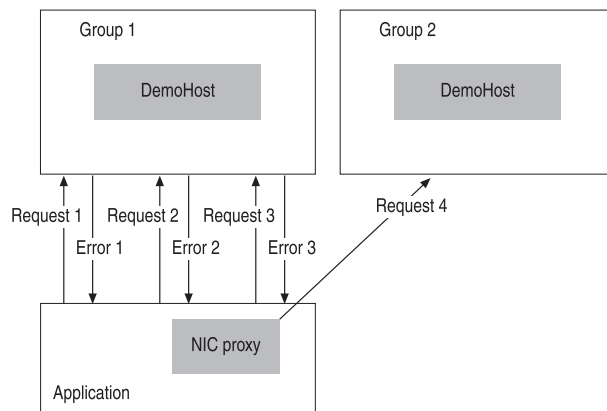
Figure 13: NIC Group Selection by Round-Robin



If the NIC host fails to respond to a specified number of resolution requests, the NIC proxy stops sending resolution requests to the unavailable NIC host and sends the resolution requests to another NIC host. The NIC proxy continues to poll the unavailable NIC host to determine its availability. When the NIC host becomes available, the NIC proxy can again send resolution requests to that host.

[Figure 14 on page 139](#) shows a NIC proxy that sends a resolution request to Group 1, receives an error message, then sends two more resolution requests before sending a request to Group 2 rather than Group 1. When Group 1 is available again, the NIC proxy will send the request to Group 1.

Figure 14: NIC Resolution Request



g015903

You configure NIC replication for hosts, then configure NIC proxies to use replication.

Although you can distribute agents and resolvers among different hosts, as shown in the configuration for the NIC hosts OnePopBO and OnePopH1 in the sample data, we recommend that you use the DemoHost configuration, which centralizes the configuration for agents and resolvers.

Related Documentation

- [Router Initialization Scripts with NIC Configuration Scenarios on page 146](#)
- [Planning a NIC Implementation on page 139](#)
- [NIC Configuration Scenarios on page 140](#)
- [NIC Agents Used in the NIC Configuration Scenarios on page 144](#)

Planning a NIC Implementation

The SRC software provides standard NIC configuration scenarios that you can modify to meet the requirements for your environment. Which scenarios you choose depends on the applications you use.

If the resolution scenarios do not provide the type of resolution needed, we recommend that you consult Juniper Professional Services.

To plan your NIC implementation:

1. Review the NIC configuration scenarios, and select the scenario that best fits the requirements for your application. In most cases, one of the basic configuration scenarios provides the type of resolution needed.

See [“NIC Configuration Scenarios” on page 140](#).
2. Determine the number of NIC proxies that you will need to access NIC hosts, and estimate the amount of traffic between the NIC proxies and the NIC hosts. If you expect heavy traffic between NIC proxies and NIC hosts, configure a number of NIC hosts to share the traffic load and processing.

3. Determine which NIC hosts to assign to a group to provide NIC replication; choose names for these groups.
4. If you have not done so already, determine which systems are to run NIC hosts.

Related Documentation

- [Testing a NIC Resolution \(SRC CLI\) on page 166](#)
- [Router Initialization Scripts with NIC Configuration Scenarios on page 146](#)
- [NIC Agents Used in the NIC Configuration Scenarios on page 144](#)
- [NIC Resolution Process Overview on page 203](#)

NIC Configuration Scenarios

Table 10 on page 140 lists the NIC configuration scenarios provided in the SRC software.

Table 10: NIC Configuration Scenarios

Configuration Scenario	Name of NIC Configuration Scenario to Use	Type of Resolution	Notes
Basic Configuration Scenarios			
For subscribers who have a tunnel ID associated with an L2TP interface use scenario Can be used in wholesaler – retailer ISP scenario, where wholesaler offers tiered services for retail customers through L2TP interfaces Sample use: Support for the L2TP interfaces	OnePopTunnel	Tunnel ID (Tunnel ID + Tunnel Session ID + LAC IP Address) of a subscriber to the SAE IOR	Tunnel ID is comprised of Tunnel ID, Tunnel Session ID, and the LAC IP address of the interface. Tunnel ID and Tunnel Session ID uniquely identify the tunnel session within the JunosE router. Combined with the LAC IP address, they uniquely identify the subscriber.
For JunosE local configuration for PPP and DHCP subscribers. Sample use: DSL providers for residential customers.	OnePop	Subscriber IP address to the SAE IOR	Simplest configuration. IP pools configured locally on each virtual router (VR) with IP addresses from a static pool of IP addresses configured on the virtual router.

Table 10: NIC Configuration Scenarios (*continued*)

Configuration Scenario	Name of NIC Configuration Scenario to Use	Type of Resolution	Notes
<p>For subscribers who have an accounting ID.</p> <p>Can be used for multiple subscribers who use the same accounting ID, in which case NIC returns all SAE IORs for mapped subscribers.</p> <p>Sample use:</p> <p>Support for the volume-tracking application.</p>	OnePopAcctId	Accounting ID of a subscriber to the SAE IOR and the IP address of a subscriber to accounting ID	<p>A subscriber's accounting ID can be specified at subscriber login from the SAE subscriber classification script. As a result, the accounting ID encapsulates other attributes of the subscriber session processed by the subscriber classification script. The OnePopAcctId configuration scenario can resolve the encapsulated attributes.</p> <p>For example, customers can assign a subscriber username (login id without domain name) to an accounting ID with the following subscriber classification.</p> <pre>[<-retailerDn- >?accountingUserId =<-userName->?sub?(uniqueID = <-userName->)]</pre>
<p>For subscribers who have assigned IP addresses (assigned external to the SAE).</p> <p>Sample use:</p> <p>In a PacketCable Multimedia Specification (PCMM) environment when the SAE acts as both a policy server and application manager.</p>	OnePopDynamicIp	Subscriber IP address to the SAE IOR	
<p>For resolution of a subscriber login name to an SAE IOR, and of a subscriber IP address to a subscriber login name.</p> <p>Sample use:</p> <p>Support for tracking subscriber bandwidth usage or for using a billing model. You can use the SRC VTA with this scenario.</p>	OnePopLogin	Subscriber login name to the SAE IOR and subscriber IP address to login name	Uses two resolvers. Use a separate NIC proxy for each resolution.

Table 10: NIC Configuration Scenarios (*continued*)

Configuration Scenario	Name of NIC Configuration Scenario to Use	Type of Resolution	Notes
For use with applications that need to support tracking a large number of subscribers.	OnePopLoginPull	Subscriber login name or a subscriber IP address to an SAE IOR	
<p>For subscribers who connect through a cable modem termination system (CMTS) device.</p> <p>Sample use:</p> <p>In a PCMM environment in which the policy server is separate from the application server. This scenario can be used when the configuration includes Juniper Policy Server or another policy server, and the SAE is an application manager.</p>	OnePopPcmm	Subscriber IP address to the SAE IOR	
<p>For use with applications that use the SAE programming interfaces and that identify subscribers by the primary username.</p> <p>Sample uses:</p> <ul style="list-style-type: none"> Aggregate services Dynamic service activator application 	OnePopPrimaryUser	Primary username of a subscriber to the SAE IOR	Similar to OnePopLogin
<p>For a router configuration in which VRs share IP pools.</p> <p>Sample use:</p> <ul style="list-style-type: none"> Services for enterprise subscribers. Support for two different proxies: Subscriber DN to the SAE IOR Subscriber IP address to the SAE IOR 	OnePopDnSharedIp	Subscriber distinguished name (DN) or subscriber IP address to the SAE IOR	Includes resolution available in OnPopSharedIp and adds resolution from a subscriber DN.

Table 10: NIC Configuration Scenarios (*continued*)

Configuration Scenario	Name of NIC Configuration Scenario to Use	Type of Resolution	Notes
<p>For a router configuration in which pools can be shared among routers. Pools can be assigned by RADIUS or by a DHCP server.</p> <p>Sample use:</p> <p>Support for DHCP and PPP connections for residential subscribers.</p>	OnePopSharedIp	Subscriber IP address to the SAE IOR	
For scenarios in which subscribers have an assigned IP address and these IP addresses can be associated with interfaces on devices running Junos OS.	OnePopStaticRouteIp	Assigned subscriber IP address to the SAE IOR	Static route information for routers resides in an XML document in the directory under the router object.
<p>For scenarios in which subscribers have an assigned IP address.</p> <p>Sample use:</p> <ul style="list-style-type: none"> Applications that use an SAE to manage a provider edge router, not directly manage end subscribers, and not support individual subscriber sessions for these subscribers. 	OnePopVrfIp	Assigned subscriber IP address to the SAE IOR	<p>Similar to OnePopStaticRouteIp. Used to support multiple VPNs with overlapping IP pools.</p> <p>Static route information for routers resides in an XML document in the directory under the router object.</p>
<p>For scenarios in which subscribers are identified by a set of IPv6 prefixes defined by the device. These IPv6 prefixes are made available to the NIC through SAE IPv6 plug-in attributes.</p> <p>Sample use:</p> <ul style="list-style-type: none"> Applications can identify subscribers based on their IP addresses and get a reference to the SAE managing the subscribers. 	OnePopPrefixIp	The OnePopPrefixIp scenario is identical to the OnePop scenario but the IP pool information is provided by the SAE (through NIC SAE Plug-in agents) instead of being read from the directory.	
For enterprise customers.	OnePopAllRealms	Subscriber IP address or subscriber DN to the SAE IOR	The scenario combines the OnePop and OnePopSharedIp scenarios and adds resolution from a subscriber DN.
Advanced Configuration Scenario			

Table 10: NIC Configuration Scenarios (*continued*)

Configuration Scenario	Name of NIC Configuration Scenario to Use	Type of Resolution	Notes
For two POPs that share a back office. Sample use: Support for a deployment that has a back office that connects to NIC hosts at other sites.	MultiPop	Subscriber IP address to the SAE IOR	You can deploy this scenario in an environment that has a number of POPs; for example, a configuration in which there are two POPs with NIC proxy communication to a back office, which in turn communicates with the POP hosts. The POP hosts each support parallel hosts and agents and manage resolutions in the same way. You can add POPs by copying the configuration for one POP and modifying the configuration to suit your environment.

- Related Documentation**
- [Configuration Statements for the NIC on page 149](#)
 - [Router Initialization Scripts with NIC Configuration Scenarios on page 146](#)
 - [NIC Agents Used in the NIC Configuration Scenarios on page 144](#)

NIC Agents Used in the NIC Configuration Scenarios

When you configure a NIC configuration scenario, you use the basic configuration for each NIC agent in the scenario, but modify properties such as directory properties to make the agent configuration compatible with your SRC configuration. The NIC configuration scenario that you use determines which agents appear in your configuration.

[Table 11 on page 144](#) lists all agents that are available in the various configuration scenarios.

Table 11: NIC Agents

Agent Name	Type of Agent	Type of Information
AcctIdIp	SAE plug-in	Mappings of accounting IDs of a subscribers to the SAE IOR and subscriber IP addresses to accounting ID(s).
DnVr	SAE plug-in	Mappings of enterprise access DNs to VRs.
Enterprise	Directory	List of enterprise names.
IpAcctId	SAE plug-in	Mappings of subscriber IP addresses to accounting IDs.
IpLoginName	SAE plug-in	Mappings of IP addresses to login names.

Table 11: NIC Agents (*continued*)

Agent Name	Type of Agent	Type of Information
IpSaeld	SAE client	Mappings of IP addresses to SAEs managing a subscriber. Uses the SAE remote interface to determine which SAEs are managing a subscriber at resolution time.
IpVr	SAE plug-in	Mappings of IP addresses to VRs.
LoginNameVr	SAE plug-in	Mappings of login names to VRs.
LoginSaeld	SAE client	Mappings of login names to SAEs. Uses the SAE remote interface to determine which SAEs are managing a subscriber at resolution time.
PoolInterface	Directory	Mappings of IP pools to an interface. Note: Reads a JunosE routing table and extracts the VR name to perform the mapping.
PoolVr	Directory	Mappings of IP pools to VRs.
TunnelIdVr	SAE plug-in	Mapping of Tunnel IDs (Tunnel ID + Tunnel/Session ID + LAC IP Address) to VRs.
UserNameVr	SAE plug-in	Mappings of subscriber IP addresses to accounting IDs.
VrSaeld	Directory	Reads information about virtual routers and the mappings between virtual routers and SAEs.

Table 12 on page 145 shows the types of agents that each configuration scenario uses.

Table 12: Agents in Configuration Scenarios

NIC Configuration Scenario	Directory Agents	SAE Plug-In Agents	SAE Client Agents	SSR Client Agents
OnePop	PoolVr, VrSaeld			
OnePopAcctId	PoolVr, VrSaeld	AcctIdIp, IpAcctId		
OnePopDnSharedIp	PoolVr, VrSaeld, Enterprise	DnVr		
OnePopDynamicIp	PoolVr, VrSaeld			
OnePopLogin	Pool, VrSaeld	IpLoginName, LoginNameVr		
OnePopLoginPull			IpSaeld, LoginSaeld	
OnePopPcmm	PoolVr, VrSaeld			
OnePopSharedIp	PoolVr, VrSaeld	IpVr		

Table 12: Agents in Configuration Scenarios (*continued*)

NIC Configuration Scenario	Directory Agents	SAE Plug-In Agents	SAE Client Agents	SSR Client Agents
MultiPop	PoolVr, VrSaeld, site-specific versions of PoolVr and VrSaeld	IpVr		
OnePopAllRealms	PoolVr, VrSaeld, Enterprise	IpVr		
OnePopPrimaryUser	VrSaeld	UserNameVr		
OnePopStaticRouteIp	VrSaeld, PoolInterface			
OnePopTunnel	VrSaeld	TunnelIdVr		
OnePopVrflip	VrSaeld, PoolInterface			

Related Documentation

- [Mapping Subscribers to a Managing SAE on page 135](#)
- [Router Initialization Scripts with NIC Configuration Scenarios on page 146](#)
- [Configuring a NIC Scenario \(SRC CLI\) on page 156](#)
- [NIC Configuration Scenarios on page 140](#)

Router Initialization Scripts with NIC Configuration Scenarios

The NIC resolutions map VRs to SAEs. For these resolutions, use a router initialization script that associates each VR with the SAE that manages it. Which router initialization script you use depends on whether the SAE obtains IP pools from JunosE VRs:

- **poolPublisher** router initialization script—Use when the SAE obtains local IP pools locally from JunosE VRs.
- **iorPublisher** router initialization script—Use when the router is one of the following:

- JunosE routers that do not supply IP addresses from local pools
- devices running Junos OS
- CMTS devices

These devices do not supply IP addresses from local pools in your network.

Table 13 on page 147 lists which type of initialization script should be used with the various NIC configuration scenarios. The OnePopLoginPull scenario does not require an initialization script.

Table 13: Type of Router Initialization Script to Use for NIC Configuration Scenarios

poolPublisher	iorPublisher	poolPublisher or iorPublisher
One Pop	OnePopDnSharedIp	OnePopAcctId
	OneLoginPull	OnePopAllReams
	OnePopPcmm	OnePopDynamicIp
	OnePopPrimaryUser	OnePopLogin
	OnePopSharedIp	MultiPop
	OnePopStaticRouteIp	
	OnePopVrflp	



NOTE: If you modify information about IP pools on a VR after the COPS connection is established, the SAE does not automatically register the changes, and you must update the directory.

For more information about router initialization scripts for JunosE routers, including how to update the directory, see “[Configuring the SAE to Manage JunosE Routers \(SRC CLI\)](#)” on page 59.

For more information about router initialization scripts for devices running Junos OS, see “[Configuring the SAE to Manage Devices Running Junos OS \(SRC CLI\)](#)” on page 86.

Related Documentation

- [High Availability for NIC on page 137](#)
- [Planning a NIC Implementation on page 139](#)
- [NIC Configuration Scenarios on page 140](#)
- [NIC Agents Used in the NIC Configuration Scenarios on page 144](#)

CHAPTER 11

Configuring the NIC (SRC CLI)

- Configuration Statements for the NIC on page 149
- Before You Configure the NIC on page 151
- Configuring the NIC (SRC CLI) on page 152
- Reviewing and Changing Operating Properties for the NIC (SRC CLI) on page 153
- Configuring NIC Replication (SRC CLI) on page 155
- Configuring a NIC Scenario (SRC CLI) on page 156
- Configuring Advanced NIC Features on page 165
- Verifying Configuration for the NIC (SRC CLI) on page 165
- Starting the NIC (SRC CLI) on page 166
- Testing a NIC Resolution (SRC CLI) on page 166
- Stopping a NIC Host on a C Series Controller (SRC CLI) on page 167
- Restarting the NIC (SRC CLI) on page 167
- Restarting a NIC Agent (SRC CLI) on page 168
- Restarting a NIC Resolver (SRC CLI) on page 168
- Changing NIC Configurations (SRC CLI) on page 169

Configuration Statements for the NIC

The SRC CLI provides the following groups of configuration statements for the NIC:

- Configuration statements for NIC operating properties
- Configuration statements for NIC scenarios
- Configuration statements for NIC logging



NOTE: We recommend that you change only those statements visible at the basic editing level. Contact Juniper Professional Services or Juniper Customer Support before you change any of the NIC statements and options not visible at the basic editing level.

Configuration Statements for NIC Operating Properties

Use the following configuration statements to configure the NIC operating properties at the **[edit]** hierarchy level. These statements are visible at the CLI basic editing level.

```
slot number nic {
  base-dn base-dn;
  java-garbage-collection-options java-garbage-collection-options;
  java-heap-size java-heap-size;
  scenario-name scenario-name;
  snmp-agent;
  hostname hostname;
  runtime-group runtime-group;
}
slot number nic initial {
  static-dn static-dn;
  dynamic-dn dynamic-dn;
}
slot number nic initial directory-connection {
  url url;
  backup-urls [ backup-urls...];
  principal principal;
  credentials credentials;
  protocol (ldaps);
  timeout timeout;
  check-interval check-interval;
  blacklist;
  snmp-agent;
}
slot number nic initial directory-eventing {
  eventing;
  signature-dn signature-dn;
  polling-interval polling-interval;
  event-base-dn event-base-dn;
  dispatcher-pool-size dispatcher-pool-size;
}
```

Configuration Statements for NIC Scenarios

Use the following configuration statements to configure the NIC at the **[edit]** hierarchy level. These statements are visible at the CLI basic editing level.

Which agents you configure depends on the NIC configuration scenario that you use.



NOTE: Although the CLI provides configuration statements for SSR Client agents, you typically do not need to change the basic configuration for these agents. Changes can be made at the expert editing level.

The CLI also provides configuration statements for consolidator agents, properties agents, and XML agents. At this time, none of the NIC configuration scenarios uses these agents. The following list does not include the configuration statements for these agents.

```

shared nic scenario name
shared nic scenario name agents name
shared nic scenario name agents name configuration directory {
  search-base search-base ;
  search-filter search-filter ;
  search-scope (0 | 1 | 2);
  server-url server-url ;
  directory-backup--urls directory-backup-urls ;
  principal principal ;
  credentials credentials ;
}
shared nic scenario name agents name configuration sae-client {
  principal principal;
  credentials credentials;
  subscriber-id (user-ip-address | dn | login-name | interface-name | primary-user-name);
  search-base search-base;
  search-filter search-filter;
  search-scope (object | one-level | sub-tree);
  server-url server-url;
  directory-backup-urls directory-backup-urls ;
}
shared nic scenario name agents agent configuration sae-plug-in {
  event-filter event-filter ;
  number-of-events number-of-events ;
}

```

Configuration Statements for NIC Logging

Use the following configuration statements to configure logging for the NIC at the [edit] hierarchy level.

```

shared nic scenario name hosts name configuration logger name syslog {
  filter filter ;
  host host ;
  facility facility ;
  format format ;
}
shared nic scenario name hosts name configuration logger name file {
  filter filter ;
  filename filename;
  rollover-filename rollover-filename ;
  maximum-file-size maximum-file-size ;
}

```

Related Documentation

- [Before You Configure the NIC on page 151](#)
- [Configuring the NIC \(SRC CLI\) on page 152](#)
- For detailed information about each configuration statement, see the *SRC PE CLI Command Reference*.

Before You Configure the NIC

When you use NIC in a client/server configuration, you configure the NIC scenario before you configure the NIC proxies.

Before you configure NIC hosts from the CLI:

- Plan your NIC implementation:
- Choose the NIC configuration scenario to use.

The default scenario is OnePop.

For information about NIC configuration scenarios and NIC agents, see [“Locating Subscriber Management Information” on page 133](#).

- Ensure that the appropriate type of router initialization script is configured for the router or network device.

See [“Locating Subscriber Management Information” on page 133](#).

Set the editing level for the configuration application you are using, the SRC CLI or the C-Web interface to basic. This ensures that only the statements that you need to configure are visible.

To set the editing level for the C-Web interface to basic:

- Click **Preferences>Level Basic**.

Related Documentation

- [Configuring the NIC \(SRC CLI\) on page 152](#)
- [Configuring the NIC \(C-Web Interface\)](#)
- [Starting the NIC \(SRC CLI\) on page 166](#)
- [NIC Agents Used in the NIC Configuration Scenarios on page 144](#)
- [Router Initialization Scripts with NIC Configuration Scenarios on page 146](#)
- [Verifying Configuration for the NIC \(SRC CLI\) on page 165](#)

Configuring the NIC (SRC CLI)

Before you configure the NIC, complete the prerequisite tasks.

See [“Before You Configure the NIC” on page 151](#).

To configure the NIC:

1. Configure NIC operating properties.
See [“Reviewing and Changing Operating Properties for the NIC \(SRC CLI\)” on page 153](#).
2. Configure NIC replication.
See [“Reviewing and Changing Operating Properties for the NIC \(SRC CLI\)” on page 153](#).
3. (Optional) If you plan to use a configuration scenario other than OnePop (the default), delete any data for the OnePop scenario and configure the scenario name to specify the configuration scenario.
See [“Changing NIC Configurations \(SRC CLI\)” on page 169](#).
4. Configure a NIC scenario.

See [“Configuring a NIC Scenario \(SRC CLI\)” on page 156.](#)

5. Verify the NIC configuration.

See [“Verifying Configuration for the NIC \(SRC CLI\)” on page 165.](#)

6. Start the NIC component.

See [“Starting the NIC \(SRC CLI\)” on page 166.](#)

Related Documentation

- [Testing a NIC Resolution \(SRC CLI\) on page 166](#)
- [Configuration Statements for the NIC on page 149](#)

Reviewing and Changing Operating Properties for the NIC (SRC CLI)

Before you configure a NIC configuration scenario, review the default operating properties and change values as needed. Operating properties are configured for a slot.

The following topics provide procedures for reviewing and changing operating properties for NIC with the SRC CLI:

1. [Reviewing the Default NIC Operating Properties on page 153](#)
2. [Changing NIC Operating Properties on page 154](#)

Reviewing the Default NIC Operating Properties

To review the default NIC operating properties:

1. From configuration mode, access the configuration statement that specifies the configuration for the NIC on a slot.

```
[edit]
user@host# edit slot number nic
```

For example:

```
[edit]
user@host# edit slot 0 nic
```

2. Run the **show** command.

```
[edit slot 0 nic]
user@host# show
base-dn o=umc;
java-runtime-environment ../jre/bin/java;
java-heap-size 128m;
snmp-agent;
hostname DemoHost;
initial {
  dynamic-dn "ou=dynamicConfiguration, ou=Configuration, o=Management,<base>";

  directory-connection {
    url ldap://127.0.0.1:389/;
    backup-urls ;
    principal cn=nic,ou=Components,o=Operators,<base>;
```

```
        credentials *****;  
        timeout 10;  
        check-interval 60;  
    }  
    directory-eventing {  
        eventing;  
        signature-dn <base>;  
        polling-interval 15;  
        event-base-dn <base>;  
        dispatcher-pool-size 1;  
    }  
    static-dn "l=OnePop,l=NIC, ou=StaticConfiguration, ou=Configuration, o=Management,<base>";  
}
```

Changing NIC Operating Properties

In most cases you can use the default NIC operating properties. Change the default properties if needed for your environment.

To change NIC operating properties:

1. From configuration mode, access the configuration statement that specifies the configuration for the NIC on a slot.

```
[edit]  
user@host# edit slot number nic
```

For example:

```
[edit]  
user@host# edit slot 0 nic
```

2. (Optional) If you store data in the directory in a location other than the default, *o=umc*, change this value.

```
[edit slot 0 nic]  
user@host# set base-dn base-dn
```

3. (Optional) Configure the garbage collection functionality of the Java Virtual Machine.

```
[edit slot 0 nic]  
user@host# set java-garbage-collection-options java-garbage-collection-options
```

4. (Optional) If you determine that additional memory is needed, change the maximum memory size available to the (Java Runtime Environment) JRE.

```
[edit slot 0 nic]  
user@host# set java-heap-size java-heap-size
```

By default, the JRE can allocate 128 MB. Set to a value lower than the available physical memory to avoid low performance because of disk swapping.

If you use an SAE plug-in agent, we recommend that you increase the JVM max heap to a value in the range 400–500 MB.

If you need help to determine the amount of memory needed, contact Juniper Networks Customer Services and Support.

5. (Optional) Specify the name of the NIC scenario that you want to configure. The default scenario is OnePop.

```
[edit slot 0 nic]
user@host# set scenario-name scenario-name
```

6. (Optional) Enable viewing of SNMP counters through an SNMP browser.

```
[edit slot 0 nic]
user@host# set snmp-agent
```

7. (Optional) Change the name of the NIC host. Use the default name of the NIC host configured for a NIC scenario. In most cases, the NIC host name is DemoHost.

```
[edit slot 0 nic]
user@host# set hostname hostname
```

8. (Optional) Change the initial properties.

See *Configuring Basic Local Properties*.

Related Documentation

- [Reviewing and Changing Operating Properties for NIC \(C-Web Interface\)](#)
- [Configuring the NIC \(SRC CLI\) on page 152](#)
- [Configuration Statements for the NIC on page 149](#)
- [Changing NIC Configurations \(SRC CLI\) on page 169](#)
- [Verifying Configuration for the NIC \(SRC CLI\) on page 165](#)

Configuring NIC Replication (SRC CLI)

You configure NIC replication to keep the NIC configuration highly available.

Before you configure NIC replication:

- Make sure that you understand how NIC groups are used.
See [“Locating Subscriber Management Information” on page 133](#).
- Identify which NIC hosts are to provide redundancy for each other.
- Select a name for a group for each of these hosts.

To configure NIC replication:

1. From configuration mode, access the configuration statement that specifies the configuration for the agent.

```
[edit]
user@host# slot number nic
```

For example:

```
[edit]
```

```
user@host# slot 0 nic
```

2. Configure the runtime group for the NIC host.

```
[edit slot 0 nic]  
user@host# runtime-group runtime-group
```

For example:

```
[edit slot 0 nic]  
user@host# runtime-group group1
```

Related Documentation

- [Configuring the NIC \(SRC CLI\) on page 152](#)

Configuring a NIC Scenario (SRC CLI)

The following topics provide procedures for configuring a NIC scenario with the SRC CLI:

- [Defining the NIC Configuration to Use on page 156](#)
- [Configuring Directory Agents on page 159](#)
- [Configuring SAE Client Agents on page 161](#)
- [Configuring SAE Plug-In Agents on page 162](#)
- [Configuring the SAE to Communicate with SAE Plug-In Agents When You Use NIC Replication on page 163](#)

Defining the NIC Configuration to Use

The OnePop configuration scenario is the default configuration for NIC. If you want to use another configuration scenario, you first clear data for the configuration scenario and change the scenario name that identifies the scenario, see “[Changing NIC Configurations \(SRC CLI\)](#)” on page 169.

When you select a NIC configuration scenario, the software adds the default configuration for most properties. You can modify the NIC properties, including those for agents.



CAUTION: We recommend that you change only those statements visible at the basic editing level. Contact Juniper Professional Services or Juniper Customer Support before you change any of the NIC statements not visible at the basic editing level.

To specify a NIC configuration scenario for NIC to use:

1. Make sure that the NIC component is running.

```
user@host> show component  
Installed Components  
Name   Version      Status  
...
```

- [edit]

For example:

[edit]

3. View the default configuration for the configuration scenario. For example:

```
[edit shared nic scenario OnePopLogin]
```

```
' ' ' ' ' ' ' ' 'credentials *****;
```

```

        sae-plugin {
            event-filter "(&(! (PA_USER_TYPE=INTF)) (! (PA_LOGIN_NAME=[None])))";
            number-of-events-sent-in-a-synchronization-call 50;
        }
    }
}
IpLoginName {
    configuration {
        sae-plugin {
            number-of-events-sent-in-a-synchronization-call 50;
        }
    }
}
PoolVr {
    configuration {
        directory {
            search-base o=Network,<base>;
            search-filter (objectclass=umcVirtualRouter);
            search-scope 2;
            server-url ldap://127.0.0.1:389/;
            backup-servers-url ;
            ' ' ' ' ' ' ' ' ' ' 'principal cn=nic,ou=Components,o=Operators,<base>;
            ' ' ' ' ' ' ' ' ' ' 'credentials *****;
        }
    }
}
}

```

4. (Optional) Update logging configuration.

See *Logging for SRC Components Overview*.

By default, NIC has the following logging enabled for a NIC host:

```

logger file-1 {
    file {
        filter !ConfigMgr,!DES,/debug-;
        filename var/log/nicdebug.log;
        rollover-filename var/log/nicdebug.alt;
        maximum-file-size 10000000;
    }
}
logger file-2 {
    file {
        filter /info-;
        filename var/log/nicinfo.log;
    }
}
logger file-3 {
    file {
        filter /error-;
        filename var/log/nicerror.log;
    }
}

```

5. For each agent that the NIC configuration scenario includes, if needed update NIC agent configuration to define properties specific to your environment, such as directory properties.

Each type of agent has different configuration properties. The output from the **show** command identifies the type of agent under the **agents** hierarchy. For example:

```
VrSaeId {
  configuration {
    directory {
LoginNameVr {
  configuration {
    sae-plugin {
```

Configuring Directory Agents

Use the following configuration statements to configure NIC directory agents:

```
shared nic scenario name agents agent configuration directory {
  search-base search-base ;
  search-filter search-filter ;
  search-scope (0 | 1 | 2);
  server-url server-url ;
  backup-servers-url backup-servers-url ;
  principal principal ;
  credentials credentials ;
}
```

To configure a directory agent:

1. From configuration mode, access the statement that specifies the configuration for the agent.

```
[edit]
user@host# edit shared nic scenario name agents agent configuration directory
```

For example:

```
[edit]
user@host# edit shared nic scenario OnePopLogin agents VrSaeld configuration
directory
```

2. Review the default configuration for the agent. For example:

```
[edit shared nic scenario OnePopLogin agents VrSaeId configuration directory]
user@host# show
search-base o=Network,<base>;
search-filter (objectclass=umcVirtualRouter);
search-scope 2;
server-url ldap://127.0.0.1:389/;
directory-backup-urls ;
principal cn=nic,ou=Components,o=Operators,<base>;
credentials *****;
```

3. (Optional) Change the distinguished name (DN) of the location in the directory from which the agent should read information.

```
[edit shared nic scenario name agents name configuration directory]
user@host# set search-base search-base
```

For example:

```
[edit shared nic scenario OnePop agents PoolVr configuration directory]
user@host# set search-base o=myNetwork,<base>
```

You can use <base> in the DN to refer to the globally configured base DN.

4. (Optional) Change the directory search filter that the agent should use.

```
[edit shared nic scenario name agents name configuration directory]
user@host# set search-filter search-filter
```

For example:

```
[edit shared nic scenario OnePop agents PoolVr configuration directory]
user@host# set search-filter objectclass=umcVirtualRouter
```

5. (Optional) Change the location in the directory relative to the base DN from which the NIC agent can retrieve information.

```
[edit shared nic scenario name agents name configuration directory]
user@host# set search-scope (0 | 1 | 2)
```

where:

- 0—Entry specified in the **search-base** statement
 - 1—Entry specified in the **search-base** statement and objects that are subordinate by one level
 - 2—Subtree of entry specified in the **search-base** statement
6. For an installation on a Solaris platform, specify the location of the directory in URL string format.

```
[edit shared nic scenario name agents name configuration directory]
user@host# set server-url ldap://host:portNumber
```

For example, to specify the directory on a C Series Controller:

```
[edit shared nic scenario OnePop agents PoolVr configuration directory]
user@host# set server-url ldap://127.0.0.1:389/
```

7. List the URLs of redundant directories. Separate URLs with semicolons.

```
[edit shared nic scenario name agents name configuration directory]
user@host# set directory-backup-urls backup-servers-urls
```

8. Specify the DN that contains the username that the directory server uses to authenticate the NIC agent.

```
[edit shared nic scenario name agents name configuration directory]
user@host# set principal principal
```

For example:

```
[edit shared nic scenario OnePop agents PoolVr configuration directory]
user@host# set principal cn=nic,ou=Components,o=Operators,<base>
```

9. Specify the password that the directory server uses to authenticate the NIC agent.


```
[edit shared nic scenario name agents name configuration directory]
user@host# set credentials credentials
```

- Restart the NIC agent.

```
user@host>request nic restart agent name name
```

Configuring SAE Client Agents

Use the following configuration statements to configure NIC SAE client agents:

```
shared nic scenario nameagents nameconfiguration sae-client {
  principal principal;
  credentials credentials;
  subscriber-id (user-ip-address | dn | login-name | interface-name | primary-user-name);
  search-base search-base;
  search-filter search-filter;
  search-scope (object | one-level | sub-tree);
  server-url server-url;
  directory-backup-urlsdirectory-backup-urls ;
}
```

To configure an SAE client agent:

- From configuration mode, access the statement that specifies the configuration for the agent.

```
[edit]
user@host# edit shared nic scenario name agents agent configuration sae-client
```

For example:

```
[edit]
user@host# edit shared nic scenario OnePopLoginPull agents IpSaeId configuration
sae-client
```

- Review the default configuration for the agent. For example:

```
[edit shared nic scenario OnePopLoginPull agents IpSaeId configuration
sae-client]
user@host# show
principal cn=umcadmin,<base>;
credentials *****;
subscriber-id user-ip-address;
search-base ou=sspadmurl,s,o=Servers;;
search-filter (objectclass=corbaObjectReference);
search-scope sub-tree;
server-url ldap://127.0.0.1:389/; directory-backup-urls "";
```

- (Optional) Change the authentication DN.

For example:

```
[edit edit shared nic scenario OnePopLoginPull agents IpSaeId configuration sae-client
]
user@host# set principal cn=umcadmin, <base>
```

4. (Optional) Change the password that the NIC uses to access the directory. For example:

```
[edit edit shared nic scenario OnePopLoginPull agents IpSaeld configuration sae-client
]
user@host# set credentials —
```

5. Specify the part of the directory that you want the network publisher to search.

```
[edit edit shared nic scenario OnePopLoginPull agents IpSaeld configuration sae-client
]
user@host# set search-base search-base
```

6. (Optional) Change the URL that identifies the primary Juniper Networks database to which the NIC agent connects.

```
[edit edit shared nic scenario OnePopLoginPull agents IpSaeld configuration sae-client
]
user@host# set server-url server-url
```

7. Specify the type of subscriber ID that the agent uses to identify the subscriber. The type can be **user-ip-address**, **dn**, **login-name**, or **interface-name**. For example, to specify an IP address:

```
[edit edit shared nic scenario OnePopLoginPull agents IpSaeld configuration sae-client
]
user@host# set subscriber-id use-ip-address
```

Configuring SAE Plug-In Agents

By default, the CORBA naming server on a C Series Controller uses port 2809. The NIC host is configured to communicate with this naming server; you do not need to change JacORB properties.

Use the following configuration statements to configure NIC SAE plug-in agents:

```
shared nic scenario name agents agent configuration sae-plug-in{
  event-filter event-filter ;
  number-of-events number-of-events ;
}
```

If you plan to change the event filter for the agent, make sure that you are familiar with:

- Plug-in attributes and values

See *Types of Tracking Plug-Ins* .

- Filter syntax

See the documentation for the SAE CORBA Remote API in the SAE Core API documentation on the Juniper Networks Web site at:

<http://www.juniper.net/techpubs/software/management/src/api-index.html>

To configure an SAE plug-in agent:

1. From configuration mode, access the statement that specifies the configuration for the agent.

```
[edit]
user@host# edit shared nic scenario name agents agent configuration sae-plugin
```

For example:

```
[edit]
user@host# edit shared nic scenario OnePopLogin agents LoginNameVr configuration
sae plugin
```

2. Review the default configuration for the agent. For example:

```
[edit shared nic scenario OnePopLogin agents LoginNameVr configuration
sae-plugin]
user@host# show
event-filter "(&(! (PA_USER_TYPE=INTF)) (! (PA_LOGIN_NAME=[None])))";
number-of-events-sent-in-a-synchronization-call 50;
```

3. (Optional) Change an LDAP filter that change the events that the agent collects.

```
[edit shared nic scenario name agents agent configuration sae-plugin]
user@host# set event-filter event-filter
```

Typically, you do not need to change this value. If you do want to filter other events, use the format ***pluginAttribute=attributeValue*** format for event filters, where:

- ***pluginAttribute*** —Plug-in attribute name
- ***attributeValue*** —Value of filter

For example:

```
[edit shared nic scenario name agents agent configuration sae-plugin]
user@host# set event-filter PA_USER_TYPE=INTF
```

4. Specify the number of events that the SAE sends to the agent at one time during state synchronization.

```
[edit shared nic scenario name agents agent configuration sae-plugin]
user@host# set number-of-events number-of-events
```

For example:

```
[edit shared nic scenario OnePopLogin agents LoginNameVr configuration sae plugin]
user@host# set number-of-events 50
```

Configuring the SAE to Communicate with SAE Plug-In Agents When You Use NIC Replication

For each NIC host that uses SAE plug-in agents, configure a corresponding external plug-in for the SAE. By default, the SAE plug-in agents share events with the single SAE plug-in. You must also configure the SAE to communicate with the SAE plug-in agent in each NIC host that you use in the NIC replication.

For information about configuring an external plug-in for the SAE, see *Configuring the SAE for External Plug-Ins (SRC CLI)*.

To configure an external plug-in:

1. From configuration mode, access the statement that specifies the configuration for an external plug-in for the SAE that communicates with the agent, and assign the plug-in a unique name.

```
[edit]
user@host# shared sae configuration plug-ins name name
```

2. Configure CORBA object reference for the plug-in.

```
[shared sae configuration plug-ins name name external]
user@host# corba-object-reference corba-object-reference
```

For the CORBA object reference, use the following syntax:

host : port-number /NameService# plugInName

where:

- ***host*** —IP address or name of the machine on which you installed the NIC host that supports the agent

For local host, use the IP address 127.0.0.1.

- ***port-number*** —Port on which the name server runs

The default port number is 2809.

- ***plugInName*** —Name under which the agent is registered in the naming service

Use the format ***nicssae_ groupname /saePort*** where ***groupname*** is the name of the replication group. (When replication is not used, the format is ***nicssae/saePort***.)

For example:

```
[shared sae configuration plug-ins name name external]
user@host# set corba-object-reference
corbaname::127.0.0.1:2809/NameService#nicssae/saePort
```

3. Configure attributes that are sent to the external plug-in for a NIC host. Because the SAE plug-in agents share the event by default, you configure only one for a NIC host.

```
[shared sae configuration plug-ins name name external]
user@host# set attr
[( router-name | user-dn | session-id | user-type | user-ip-address | login-name)]
```

Specify the plug-in options that the agent uses. You must specify the options **session-id** and **router-name**, and other options that you specified for the agent's network data types and the agent's event filter. Do not specify attributes options of the PAT_OPAQUE attribute type, such as the option **dhcp-packet**.



NOTE: Do not include attributes that are not needed.

4. Reference the NIC as a subscriber tracking plug-in.

```
[edit shared sae group name configuration plugins event-publishers]
user@host# set subscriber-tracking pool-name
```

For example, for a pool named nic:

```
[edit shared sae group name configuration plugins event-publishers]
user@host# set subscriber-tracking nic
```

Related Documentation

- [Configuring a NIC Scenario \(C-Web Interface\)](#)
- [Configuring the NIC \(SRC CLI\) on page 152](#)
- [Verifying Configuration for the NIC \(SRC CLI\) on page 165](#)
- [Configuration Statements for the NIC on page 149](#)
- [NIC Configuration Scenarios Overview on page 209](#)

Configuring Advanced NIC Features

If you want to configure NIC features not available at the basic editing level, set the editing level to advanced or expert and use the CLI Help to obtain information about statement options.

Related Documentation

- [Configuring the NIC \(SRC CLI\) on page 152](#)
- [Configuring a NIC Scenario \(SRC CLI\) on page 156](#)
- [Configuring NIC to Store Log Messages in a File \(C-Web Interface\)](#)

Verifying Configuration for the NIC (SRC CLI)

Purpose After you complete the NIC configuration, verify the local NIC configuration and the NIC configuration scenario information.

Action To verify NIC configuration:

1. In configuration mode, run the **show** command at the **[edit slot 0 nic]** hierarchy level.

```
[edit slot 0 nic]
user@host# show
```

2. In configuration mode, run the **show** command at the **[edit shared nic scenario *name*]** hierarchy level.

For example:

```
[edit shared nic scenario OnePop]
```

```
user@host# show
```

Related Documentation

- [Starting the NIC \(SRC CLI\) on page 166](#)
- [Configuring the NIC \(SRC CLI\) on page 152](#)
- [Testing a NIC Resolution \(SRC CLI\) on page 166](#)
- [Changing NIC Configurations \(SRC CLI\) on page 169](#)

Starting the NIC (SRC CLI)

Start the NIC component before you configure it. When you enable NIC for the first time, it creates the default operating properties for the component.

To start NIC:

- From operational mode, enable the NIC.

```
user@host> enable component nic
Starting NICHOST: may take a few minutes...
```

Related Documentation

- [Starting the NIC \(C-Web Interface\)](#)
- [Configuring the NIC \(SRC CLI\) on page 152](#)
- [Reviewing and Changing Operating Properties for the NIC \(SRC CLI\) on page 153](#)
- [Restarting the NIC \(SRC CLI\) on page 167](#)
- [Stopping a NIC Host on a C Series Controller \(SRC CLI\) on page 167](#)

Testing a NIC Resolution (SRC CLI)

To test a NIC resolution:

- Run the **test nic resolve** command.

```
user@host> test nic resolve <locator locator> <key key>
```

where:

- **locator** —Name of locator that requests information on behalf of an application
- **key** —Value to be resolved. This value must be of the same NIC data type configured in the NIC locator.

For example:

```
user@host> test nic resolve locator /nicLocators/ip key 10.10.10.10
```

Example: Testing a NIC Resolution

The following example shows a successful resolution for an IP key that has the value 192.168.8.2:

```
user@host> test nic resolve locator /nicLocators/ip key 192.168.8.2
IOR:
00000000000003549444CA738D67742E6A756E9706572E6E65742F7361652F536572769636541637469766174696F6E456E67696E653A312E3000
0000000000001000000000000068000102000000000F3137322F32382F3233302F3132300000022610000000000107372632D382F736165504F412F53
414500000002000000000000008000000004A41430000000010000001C0000000000100010000000105010001001090000000105010001
user@host>
```

The following example shows an unsuccessful resolution for an IP key that has the value 192.168.8.2:

```
user@host> test nic resolve locator /nicLocators/ip key 192.168.3.2
Failed to resolve key 192.168.3.2 for resolver /nicLocators/ip due to
net.juniper.smgmt.gateway.nic.protocol.NICExc
IDL:net/juniper/smgmt/gateway/nic/protocol/NICExc:1.0
user@host>
```

Related Documentation

- [Configuring NIC Test Data \(SRC CLI\) on page 193](#)
- [Testing a NIC Resolution \(C-Web Interface\)](#)
- [Stopping a NIC Host on a C Series Controller \(SRC CLI\) on page 167](#)

Stopping a NIC Host on a C Series Controller (SRC CLI)

If you run NIC in client/server mode, you can stop the NIC host independently of the NIC proxy.

To stop a NIC host:

- From operational mode, disable the NIC.

```
user@host> disable component nic
```

Related Documentation

- [Stopping a NIC Host on a C Series Controller \(C-Web Interface\)](#)
- [Restarting the NIC \(SRC CLI\) on page 167](#)
- [Restarting a NIC Agent \(SRC CLI\) on page 168](#)
- [Restarting a NIC Resolver \(SRC CLI\) on page 168](#)
- [Changing NIC Configurations \(SRC CLI\) on page 169](#)

Restarting the NIC (SRC CLI)

To restart a NIC host:

- From operational mode, restart the NIC.

```
user@host> request restart nic
```

You can also restart the NIC at the slot level.

**Related
Documentation**

- [Stopping a NIC Host on a C Series Controller \(SRC CLI\) on page 167](#)
- [Restarting a NIC Agent \(SRC CLI\) on page 168](#)
- [Restarting a NIC Resolver \(SRC CLI\) on page 168](#)
- [Changing NIC Configurations \(SRC CLI\) on page 169](#)
- [Restarting the NIC \(C-Web Interface\)](#)

Restarting a NIC Agent (SRC CLI)

You can restart a NIC agent to have the agent read all data in the directory again. Restart a NIC agent if the agent is not synchronized with the directory, or if you switch from one directory to another.

To restart a NIC agent:

- From operational mode, restart the agent.
`user@host>request nic restart agent name name`

You can restart all NIC agents by omitting an agent name for the **request nic restart agent** command.

You can also restart a NIC agent at the slot level.

**Related
Documentation**

- [Stopping a NIC Host on a C Series Controller \(SRC CLI\) on page 167](#)
- [Restarting the NIC \(SRC CLI\) on page 167](#)
- [Restarting a NIC Resolver \(SRC CLI\) on page 168](#)
- [Changing NIC Configurations \(SRC CLI\) on page 169](#)

Restarting a NIC Resolver (SRC CLI)

In rare instances, such as when you are troubleshooting a NIC configuration, you may want to restart a NIC resolver.

To restart a NIC resolver:

- From operational mode, restart a resolver.
`user@host>request nic restart resolver name name`

You can restart all NIC resolvers by omitting a resolver name for the **request nic restart resolver** command.

You can also restart a NIC resolver at the slot level.

- Related Documentation**
- [Stopping a NIC Host on a C Series Controller \(SRC CLI\) on page 167](#)
 - [Restarting the NIC \(SRC CLI\) on page 167](#)
 - [Restarting a NIC Agent \(SRC CLI\) on page 168](#)
 - [Changing NIC Configurations \(SRC CLI\) on page 169](#)

Changing NIC Configurations (SRC CLI)

If you change the type of NIC resolution that you use in your network (for example, from the OnePop configuration scenario to the OnePopAllRealms configuration scenario), delete any existing data and specify the scenario name for the new NIC configuration scenario; otherwise, the new NIC configuration may not perform resolutions correctly.

To change the type of NIC resolution that you use in your network:

1. Set the editing level for the CLI to expert.

```
user@host> set cli level expert
```
2. Disable the NIC:

```
user@host> disable component nic
```
3. Delete the NIC configuration data for the existing configuration scenario from the directory.

```
user@host> request nic clear scenario-data
```
4. Navigate to the **[edit slot 0 nic]** hierarchy level.
5. Change the value of **scenario-name** for the local configuration to identify the new configuration scenario. For example:

```
[edit slot 0 nic]
user@host# set scenario-name OnePopSharedIp
```
6. Return to operational mode, and restart the NIC host.

```
user@host> request nic slot number restart
```
7. Set the editing level for the CLI to basic.

```
user@host> set cli level basic
```
8. Configure the new NIC scenario.

- Related Documentation**
- [Configuring the NIC \(SRC CLI\) on page 152](#)
 - [Configuring Advanced NIC Features on page 165](#)
 - [NIC Configuration Scenarios on page 140](#)

- [Configuration Statements for the NIC on page 149](#)
- *Changing NIC Configurations (C-Web Interface)*

CHAPTER 12

Obtaining Interface Configuration for OnePopStaticRouteIp or OnePopVrfIp

- [Network Publisher Overview on page 171](#)
- [NIC Document That Maps Subscriber IP Addresses to a Junos OS Interface on page 172](#)
- [Configuration Statements for the Network Publisher on page 172](#)
- [Before You Configure and Run the Network Publisher on page 173](#)
- [Configuring the Network Publisher \(SRC CLI\) on page 174](#)
- [Running the Network Publisher \(SRC CLI\) on page 179](#)
- [Files Used to Test Network Publisher on page 180](#)
- [Configuring Information to Test the Network Publisher \(SRC CLI\) on page 180](#)
- [Troubleshooting Network Publisher Operations \(SRC CLI\) on page 181](#)
- [Reviewing the Information Collected from a Device Running Junos OS \(SRC CLI\) on page 182](#)

Network Publisher Overview

The network publisher is a NIC component that connects to devices running Junos OS and collects information, such as information about system interfaces and VPNs, from IPv4 and IPv6 routing tables. After collecting the information, the network publisher stores this information in the Juniper Networks database for access by the NIC.

Use the network publisher to collect information from Junos OS routing tables for the following configuration scenarios:

- **OnePopStaticRouteIp**—Resolves an IP address for a subscriber whose traffic enters the network through a Junos OS interface to a reference for the SAE that manages the interface.
- **OnePopVrfIp**—Resolves an IP address for a subscriber whose traffic enters the network through a VPN configured on a Junos OS interface. This scenario provides support for multiple VPNs that have overlapping IP pools.

You run the network publisher whenever you want to get routing table information from one or more routers; the network publisher does not automatically update configuration information in the directory.

- Related Documentation**
- [NIC Document That Maps Subscriber IP Addresses to a Junos OS Interface on page 172](#)
 - [Files Used to Test Network Publisher on page 180](#)
 - [Before You Configure and Run the Network Publisher on page 173](#)
 - [Configuration Statements for the Network Publisher on page 172](#)
 - [Configuring the Network Publisher \(C-Web Interface\)](#)

NIC Document That Maps Subscriber IP Addresses to a Junos OS Interface

NIC stores information about IP pools or networks that map to Junos OS interfaces using routing table information. These files comply with the syntax in the file `/opt/UMC/nic/etc/networkConfig.xsd`. A sample file `/opt/UMC/nic/networkConfig.xml` shows the type of information generated by the network publisher.

- Related Documentation**
- [Network Publisher Overview on page 171](#)
 - [Reviewing the Information Collected from a Device Running Junos OS \(SRC CLI\) on page 182](#)

Configuration Statements for the Network Publisher

Use the following configuration statements to configure the network publisher.

```
slot number network-publisher logger logger-name file {
    filter filter;
    filename filename;
    rollover-filename rollover-filename;
    maximum-file-size maximum-file-size;
}
slot number network-publisher logger logger-name syslog {
    filter filter;
    hostname hostname;
    facility facility;
    format format;
}
slot number network-publisher routers {
    router-release-number router-release-number;
    router-script-version router-script-version;
}
slot number network-publisher routers authentication {
    login-name login-name;
    credentials credentials;
    protocol protocol;
}
slot number network-publisher routers router router-name {
    router-address router-address;
    router-release-number router-release-number;
    router-script-version router-script-version;
}
```

```

slot number network-publisher routers router router-name authentication {
    login-name login-name;
    credentials credentials;
    protocol protocol;
}
slot number network-publisher select {
    route-table-filter route-table-filter;
    route-entry-filter route-entry-filter;
}
slot number network-publisher directory-connection {
    url url;
    principal principal;
    credentials credentials;
    base-dn base-dn;
}
slot number network-publisher routers test-mode {
    enable-file-input;
    input-location input-location;
    enable-file-output;
    output-location output-location;
}
slot number network-publisher routers router router-name test-mode {
    enable-file-input;
    input-location input-location;
    enable-file-output;
    output-location output-location;
}

```

Related Documentation

- For detailed information about each configuration statement, see the *SRC PE CLI Command Reference*.
- [Network Publisher Overview on page 171](#)
- [Before You Configure and Run the Network Publisher on page 173](#)
- [Configuring the Network Publisher \(SRC CLI\) on page 174](#)

Before You Configure and Run the Network Publisher

Before you configure and run the network publisher:

- Verify the version of the Junos OS that is running on each devices running Junos OS.
Typically, all the devices running Junos OS should run the same version of the Junos OS.
- Verify that the C Series Controller can connect to the SAE-managed devices running Junos OS.
- Make sure that an SSH (recommended) or a Telnet service is enabled on each router from which the network publisher is to collect interface information.

When you run the network publisher, it connects to a number of devices running Junos OS through the configured protocol.

- Identify the routing tables and elements in the routing tables from which you want the network publisher to collect information. Which tables and elements you select depends on the application to use the NIC OnePopStaticRouteIp or the OnePopVrflp configuration scenario.
- Before you run the network publisher, make sure that the NIC is enabled.

**Related
Documentation**

- [Configuring the Network Publisher \(SRC CLI\) on page 174](#)
- [Configuring the Network Publisher \(C-Web Interface\)](#)
- [Starting the NIC \(SRC CLI\) on page 166](#)
- [Network Publisher Overview on page 171](#)

Configuring the Network Publisher (SRC CLI)

To configure the network publisher, complete the following tasks:

1. [Configuring Local Configuration for the Network Publisher on page 174](#)
2. [Configuring Connections Between Devices Running Junos OS and the Network Publisher on page 175](#)
3. [Configuring Router Authentication for the Network Publisher on page 176](#)
4. [Configuring Routing Table Filters for the Network Publisher on page 177](#)
5. [Configuring the Connection Between the Network Publisher and the Juniper Networks Database on page 178](#)

Configuring Local Configuration for the Network Publisher

You configure the network publisher for a slot. There is no shared configuration for the network publisher.

Use the following configuration statements to configure the basic local configuration for the network publisher:

```
slot number network-publisher logger logger-name file {  
    filter filter;  
    filename filename;  
    rollover-filename rollover-filename;  
    maximum-file-size maximum-file-size;  
}  
slot number network-publisher logger logger-name syslog {  
    filter filter;  
    hostname hostname;  
    facility facility;  
    format format;  
}
```

To set up the basic configuration for the network publisher:

1. From configuration mode, access the configuration statement that specifies the configuration for the network publisher for a slot.

```
[edit]
user@host# edit slot 0 network-publisher
```

2. Configure logging for the network publisher as you do for other SRC components.

Configuring Connections Between Devices Running Junos OS and the Network Publisher

The network publisher connects to the Junos XML management protocol server on a device running Junos OS. You can configure connection information for a group of routers running Junos OS that use the same version of Junos XML management protocol, and configure information for devices running Junos OS that use a different version.

Use the following configuration statements to configure connection information to allow the network publisher to connect to devices running Junos OS:

```
slot number network-publisher routers {
  router-release-number router-release-number;
  router-script-version router-script-version;
}
slot number network-publisher routers router router-name {
  router-address router-address;
  router-release-number router-release-number;
  router-script-version router-script-version;
}
```

To configure Junos XML management protocol connection information for the network publisher to connect to devices running Junos OS:

1. From configuration mode, access the configuration statement that specifies the configuration for the network publisher for a slot.

```
[edit]
user@host# edit slot 0 network-publisher routers
```

2. Specify the release number of the Junos OS running on the devices.

```
[edit slot 0 network-publisher routers]
user@host# set router-release-number 8.5R1
```

3. (Optional) Specify the version of Junos XML management protocol running on the devices running Junos OS.

```
[edit slot 0 network-publisher routers]
user@host# set router-script-version 1.0
```

4. (Optional) Configure connection information for devices running Junos OS that use a different version of the Junos OS to the Junos XML management protocol software.
 - a. Specify the router name of the router that uses a different version of the software.

```
[edit slot 0 network-publisher routers]
user@host# set router my-router
```

- b. Configure the IP address of the router.

```
[edit slot 0 network-publisher routers router my-router]
```

```
user@host# set router address 10.10.4..4
```

- c. Specify the release number of the Junos OS running on the devices.

```
[edit slot 0 network-publisher routers router my-router]
```

```
user@host# set router-release-number 8.5R2
```

- d. Specify the version of Junos XML management protocol running on the devices running Junos OS.

```
[edit slot 0 network-publisher routers router my-router]
```

```
user@host# set router-script-version 1.0
```

Configuring Router Authentication for the Network Publisher

You can configure connection authentication information for a group of devices running Junos OS that use the same authentication information, and configure information for devices running Junos OS that use a different username and password.



NOTE: For the network publisher to access devices running Junos OS, configure authentication for all devices or each specific device.

Use the following configuration statements to configure connection authentication information to allow the network publisher to connect to devices running Junos OS:

```
slot number network-publisher routers authentication {  
  login-name login-name;  
  credentials credentials;  
  protocol protocol;  
}  
slot number network-publisher routers router router-name authentication {  
  login-name login-name;  
  credentials credentials;  
  protocol protocol;  
}
```

To configure authentication information for the network publisher to connect to devices running Junos OS:

1. From configuration mode, access the configuration statement that specifies the configuration for router authentication.

```
[edit]
```

```
user@host# edit slot 0 network-publisher routers authentication
```

2. Specify the release number of the Junos OS running on the devices.

```
[edit slot 0 network-publisher routers]
```

```
user@host# set router-release-number 8.5R1
```

3. Specify the protocol to connect to the device running Junos OS. We recommend that you use SSH.

```
[edit slot 0 network-publisher routers authentication]
```



```
user@host# set protocol ssh
```

4. Specify the username to log into the Junos OS.

```
[edit slot 0 network-publisher routers authentication]
user@host# set login-name Chris-Bee
```

5. Specify the password for the username.

```
[edit slot 0 network-publisher routers authentication]
user@host# set credentials credentials
```

6. (Optional) Configure authentication information for devices running Junos OS that use different authentication information.

- a. Specify the router name.

```
[edit slot 0 network-publisher routers]
user@host# edit router my-router authentication
```

- b. Specify the username to log into the Junos OS.

```
[edit slot 0 network-publisher routers router my-router authentication]
user@host# set login-name Bee-C
```

- c. Specify the password for the username.

```
[edit slot 0 network-publisher routers router my-router authentication]
user@host# set credentials credentials
```

Configuring Routing Table Filters for the Network Publisher

The network publisher can collect information from Junos IPv4 and IPv6 routing tables. Specify which routing tables the network publisher should include to meet the requirements of your application that uses the NIC OnePopStaticRouteIp or OnePopVrflp configuration scenario.

By default, the network publisher collects information from all IPv4 routing tables, including tables for VPNs, and entries for all protocols. Based on your network configuration, consider which protocols to exclude from the configuration for network publisher.

Use the following configuration statements to identify the routing tables and routing table elements from which to collect information for the network publisher:

```
slot number network-publisher select {
  route-table-filter route-table-filter ;
  route-entry-filter route-entry-filter ;
}
```

To specify the routing tables from which the network publisher collects information:

1. From configuration mode, access the configuration statement that specifies the configuration for the IPv4 and IPv6 routing tables from which the network publisher is to collect information.

```
[edit]
user@host# edit slot 0 network-publisher select
```

2. Specify the routing table from which the network publisher collects information:

```
[edit slot 0 network-publisher select]
user@host# set route-table-filter route-table-filter
```

For example, to select only IPv6 tables:

```
[edit slot 0 network-publisher select]
user@host# set route-table-filter "table-name=*inet6.0"
```

You can use regular expressions to identify routing tables.

3. Specify the element(s) in a routing table:

```
[edit slot 0 network-publisher select]
user@host# set route-entry-filter route-entry-filter
```

For example, to select only those entries that pertain to OSPF advertisements:

```
[edit slot 0 network-publisher select]
user@host# set route-entry-filter "protocol=OSPF"
```

Configuring the Connection Between the Network Publisher and the Juniper Networks Database

Configure the connection properties that the network publisher uses to connect to the Juniper Networks database. The network publisher can then store information about routing tables from devices running Junos OS in the Juniper Networks database.

Use the following configuration statements to configure the connection information that the network publisher uses to connect to the Juniper Networks database:

```
slot number network-publisher directory-connection {
  url url;
  principal principal;
  credentials credentials;
  base-dn base-dn;
}
```

To configure connection information for the Juniper Networks database:

1. From configuration mode, access the configuration statement that specifies the configuration for router authentication.

```
[edit]
user@host# edit slot 0 network-publisher directory-connection
```

2. Specify the URL of the primary Juniper Networks database.

```
[edit slot 0 network-publisher directory-connection]
user@host# set url url
```

3. Specify the distinguished name (DN) that defines the username with which the network publisher accesses the Juniper Networks database, for example `cn = umcadmin, o = umc`.

```
[edit slot 0 network-publisher directory-connection]
user@host# set principal cn=umcadmin,o=umc
```

4. Specify the password with which the network publisher accesses the Juniper Networks database; for example:

```
[edit slot 0 network-publisher directory-connection]
user@host# set credentials admin123
```

5. (Optional) Specify the DN of the subtree in the database that stores the router data; for example `o = Network, o = umc`:

```
[edit slot 0 network-publisher directory-connection]
user@host# set base-dn o=Network,o=umc
```

Related Documentation

- [Before You Configure and Run the Network Publisher on page 173](#)
- [Configuring System Logging \(SRC CLI\)](#)
- [Configuring an SRC Component to Store Log Messages in a File \(SRC CLI\)](#)
- [Running the Network Publisher \(SRC CLI\) on page 179](#)
- [Network Publisher Overview on page 171](#)

Running the Network Publisher (SRC CLI)

You run the network publisher each time you want to collect information from routing tables on devices running Junos OS.

Before you run the network publisher, make sure that:

- The network publisher is configured.
- The NIC is enabled.

To run the network publisher:

- From operational mode, run one of the following commands:

```
user@host> request network-publisher execute
```

```
user@host> request network-publisher slot 0 execute
```

Related Documentation

- [Before You Configure and Run the Network Publisher on page 173](#)
- [Configuring the Network Publisher \(SRC CLI\) on page 174](#)
- [Starting the NIC \(SRC CLI\) on page 166](#)
- [Network Publisher Overview on page 171](#)

- [Files Used to Test Network Publisher on page 180](#)

Files Used to Test Network Publisher

You can configure the network publisher to use files to test a configuration or to troubleshoot network publisher operation.

Network publisher supports the following types of files:

- Input files—Use to test a configuration before routes to the NIC are available or before VPNs are configured. You can also use input files to set up a test configuration for demonstration purposes.
- Output files—Use to view the information collected from the router to see whether the network publisher is collecting the information you expect.

You must enable the network publisher to use files. Although you can specify a directory location for these files at the advanced editing level, we recommend that you use the default filenames:

- Input file—`/opt/UMC/nic/var/sample/junos/rt/router--name_1.xml`
- Output file for a specific router—`/opt/UMC/nic/var/junos/rt/router--name_1.xml`

Related Documentation

- [Network Publisher Overview on page 171](#)
- [Configuring Information to Test the Network Publisher \(SRC CLI\) on page 180](#)
- [Reviewing the Information Collected from a Device Running Junos OS \(SRC CLI\) on page 182](#)

Configuring Information to Test the Network Publisher (SRC CLI)

You can use an input file to verify that the network publisher is collecting information as configured or to set up a demonstration for an application.

To configure the network publisher to use an input file:

1. Enable the network publisher to use an input file for all routers or for a specific router.

Sample syntax for all routers:

```
[edit slot 0 network-publisher routers test-mode]  
user@host# set enable-file-input
```

Sample syntax to collect information for a router named my-router:

```
[edit slot 0 network-publisher routers router my-router test-mode]  
user@host# set enable-file-input
```

2. Run the network publisher.

```
user@host> request network-publisher execute
```

- Related Documentation**
- [Configuring Information to Test the Network Publisher \(C-Web Interface\)](#)
 - [Network Publisher Overview on page 171](#)
 - [Files Used to Test Network Publisher on page 180](#)
 - [Troubleshooting Network Publisher Operations \(SRC CLI\) on page 181](#)

Troubleshooting Network Publisher Operations (SRC CLI)

Problem The network publisher is not collecting the expected data.

- Solution**
1. Make sure that the network publisher can connect to the configured routers.
 2. Make sure that authentication is configured correctly for the network publisher and on the router.
 3. Verify the configuration for the network publisher.

```
[edit slot 0 network-publisher]
user@host# show
directory-connection {
  url ldap://127.0.0.1:389;
  base-dn o=Network,o=UMC;
  principal cn=umcadmin,o=umc;
  credentials *****;
}
select {
}
logger log1 {
  file {
    filter /debug-;
    filename var/log/netpub_debug.log;
    rollover-filename var/log/netpub_debug.alt;
    maximum-file-size 2000000;
  }
}
logger log2 {
  file {
    filter /info-;
    filename var/log/netpub_info.log;
    rollover-filename var/log/netpub_info.alt;
    maximum-file-size 2000000;
  }
}
logger log3 {
  file {
    filter /error-;
    filename var/log/netpub_error.log;
    rollover-filename var/log/netpub_error.alt;
    maximum-file-size 2000000;
  }
}
routers {
  router-release-number 7.6R1;
  authentication {
```

```
login-name admin2;
credentials *****;
}
router elf {
  address 10.227.7.115;
}
router giant {
  address 10.227.7.124;
}
}
```

4. Configure the network publisher to use an input file to ensure that the network publisher is collecting information as configured. Modify the content of the input file to reflect the router information.

See [“Configuring Information to Test the Network Publisher \(SRC CLI\)” on page 180](#)

5. Configure the network publisher to use an output file, and review the file.

See [“Reviewing the Information Collected from a Device Running Junos OS \(SRC CLI\)” on page 182](#)

**Related
Documentation**

- [Before You Configure and Run the Network Publisher on page 173](#)
- [Configuring the Network Publisher \(SRC CLI\) on page 174](#)
- [Network Publisher Overview on page 171](#)

Reviewing the Information Collected from a Device Running Junos OS (SRC CLI)

Purpose Review information that the network publisher collects from a device running Junos OS.

- Action**
1. Enable an output file to collect information from all routers or for a specific router.

Sample syntax for all routers:

```
[edit slot 0 network-publisher routers test-mode]
user@host# set enable-file-output
```

Sample syntax to collect information for a router named my-router:

```
[edit slot 0 network-publisher routers router my-router test-mode]
user@host# set enable-file-output
```

2. Run the network publisher.

```
user@host> request network-publisher execute
```
3. Use FTP to transfer the file from the C Series Controller to another system; then open the file on the remote system and examine the file content.

**Related
Documentation**

- [Network Publisher Overview on page 171](#)
- [Files Used to Test Network Publisher on page 180](#)
- [Troubleshooting Network Publisher Operations \(SRC CLI\) on page 181](#)

- *Specifying Filenames and URLs*
- *Reviewing the Information Collected from a Device Running Junos OS (C-Web Interface)*

CHAPTER 13

Configuring Applications to Communicate with an SAE

- [NIC Proxy Configuration Overview on page 185](#)
- [Before You Configure a NIC Proxy on page 186](#)

NIC Proxy Configuration Overview

You configure applications to communicate with network information collector (NIC) hosts. A NIC host can be local within an application, or external to the application. For Java applications, you also configure NIC proxies as part of an application.

For a number of SRC components, such as the SRC Volume-Tracking Application (SRC VTA) and the Dynamic Service Activator, you can configure the NIC proxy for the application from the SRC CLI. For other applications, such as the sample residential portal, you configure the NIC proxy in a property file. If you configure a NIC proxy from a property file, the fields are the same as the fields that appear at the CLI. When you develop and test SRC components that use a NIC, you can configure a NIC proxy stub to take the place of the NIC host.

For more information about NIC proxies, see [“Locating Subscriber Management Information” on page 133](#).

Related Documentation

- [Configuring Resolution Information for a NIC Proxy \(SRC CLI\) on page 188](#)
- [Changing the Configuration for the NIC Proxy Cache \(SRC CLI\) on page 190](#)
- [Before You Configure a NIC Proxy on page 186](#)
- [Configuration Statements for NIC Proxies on page 187](#)
- [Configuring a NIC Proxy for NIC Replication \(SRC CLI\) on page 191](#)
- [Removing the NIC Proxies on page 202](#)

Before You Configure a NIC Proxy

Before you configure a NIC proxy, you should have a good understanding of:

- NIC resolution
- NIC data types
- How NIC proxies work

See [“Locating Subscriber Management Information” on page 133](#), [“NIC Resolution Process Overview” on page 203](#), and [“NIC Proxy Configuration Overview” on page 185](#).



NOTE: You cannot configure a local NIC host when the NIC is running on a C Series Controller.

The values that you configure for a NIC proxy depend on the particular application; for example, you must specify the type of data used for the key and the type of data used for the value for each application.

Before you configure a NIC proxy for an application, obtain the following information from the system manager who maintains the NIC configuration for NIC hosts:

- The name of the resolver that the application uses.
- The type of key the application will provide to the NIC host.
- The type of value the NIC host is to return.
- Whether or not the application will use a local NIC host.
- If the application does not use a local NIC host:
 - The size of the NIC proxy cache.
 - The groups to be listed for NIC host selection. These groups provide NIC replication.

Related Documentation

- [Configuring a NIC Proxy \(C-Web Interface\)](#)
- [Configuring Resolution Information for a NIC Proxy \(SRC CLI\) on page 188](#)
- [Changing the Configuration for the NIC Proxy Cache \(SRC CLI\) on page 190](#)
- [Instantiating a Configuration Manager on page 198](#)
- [Configuration Statements for NIC Proxies on page 187](#)

Configuring SRC Applications to Communicate with an SAE (SRC CLI)

- [Configuration Statements for NIC Proxies on page 187](#)
- [Configuring Resolution Information for a NIC Proxy \(SRC CLI\) on page 188](#)
- [Changing the Configuration for the NIC Proxy Cache \(SRC CLI\) on page 190](#)
- [Configuring a NIC Proxy for NIC Replication \(SRC CLI\) on page 191](#)
- [Configuring NIC Test Data \(SRC CLI\) on page 193](#)

Configuration Statements for NIC Proxies

Use the following configuration statements to configure a NIC proxy for SRC components. You access these statements from the hierarchy for a component, such as:

- **[edit shared acp configuration]**
- [edit shared vta nic-proxy *name*]
- **[edit shared sae configuration]**

```
nic-proxy-configuration name {  
}  
  
nic-proxy-configuration name resolution {  
  resolver-name resolver-name;  
  key-type key-type;  
  value-type value-type;  
  expect-multiple-values;  
  constraints constraints;  
}  
  
nic-proxy-configuration name cache {  
  cache-size cache-size;  
  cache-cleanup-interval cache-cleanup-interval;  
  cache-entry-age cache-entry-age;  
}  
  
nic-proxy-configuration name nic-host-selection {  
  groups groups;  
  selection-criteria (roundRobin | randomPick | priorityList);  
}  
  
nic-proxy-configuration name nic-host-selection blacklisting {
```

```
try-next-system-on-error;  
number-of-retries-before-blacklisting number-of-retries-before-blacklisting;  
blacklist-retry-interval blacklist-retry-interval;  
}
```

Use the following statements to configure a NIC proxy stub for SRC components. You access these statements from the hierarchy for a component, such as:

- [edit shared dsa configuration]
- [edit shared vta configuration]
- [edit shared sae configuration]

```
nic-proxy-configuration name test-nic-bindings {  
  use-test-bindings;  
}  
  
nic-proxy-configuration name test-nic-bindings key-values name {  
  value;  
}
```

Related Documentation

- [Before You Configure a NIC Proxy on page 186](#)
- For detailed information about each configuration statement, see *SRC PE CLI Command Reference*.
- [Configuring Resolution Information for a NIC Proxy \(SRC CLI\) on page 188](#)
- [Changing the Configuration for the NIC Proxy Cache \(SRC CLI\) on page 190](#)
- [Configuring a NIC Proxy for NIC Replication \(SRC CLI\) on page 191](#)

Configuring Resolution Information for a NIC Proxy (SRC CLI)

Use the following statements to configure resolution information for a NIC proxy:

```
nic-proxy-configuration name resolution {  
  resolver-name resolver-name;  
  key-type key-type;  
  value-type value-type;  
  expect-multiple-values;  
  constraints constraints;  
}
```

To configure resolution information for a NIC proxy:

1. From configuration mode, access the configuration statement that specifies the NIC proxy configuration.

```
[edit]  
user@host# component-hierarchy nic-proxy-configuration name resolution
```

For example:

```
[edit]  
user@host# edit shared sae configuration nic-proxy-configuration ip resolution
```

2. Specify the NIC resolver that this NIC proxy uses.

```
[edit shared sae configuration nic-proxy-configuration ip resolution]
user@host# set resolver-name resolver-name
```

This resolver must be the same as one that is configured on the NIC host. For example:

```
[edit shared sae configuration nic-proxy-configuration ip resolution]
user@host# set resolver-name /realms/ip/A1
```

3. Specify the NIC data type that the key provides for the NIC resolution.

```
[edit shared sae configuration nic-proxy-configuration ip resolution]
user@host# set key-type key-type
```

For example:

```
[edit shared sae configuration nic-proxy-configuration ip resolution]
user@host# set key-type ip
```

To qualify data types, enter a qualifier within parentheses after the data type; for example, to specify username as a qualifier for the key `LoginName`:

```
[edit shared sae configuration nic-proxy-configuration ip resolution]
user@host# set key-type LoginName (username)
```

4. Specify the type of value to be returned in the resolution for the application that uses the NIC proxy.

```
[edit shared sae configuration nic-proxy-configuration ip resolution]
user@host# set value-type value-type
```

For example:

```
[edit shared sae configuration nic-proxy-configuration ip resolution]
user@host# set value-type SaeId
```

5. (Optional) If the key can have more than one value, specify that the key can have multiple corresponding values.

```
[edit shared sae configuration nic-proxy-configuration ip resolution]
user@host# set expect-multiple-values
```

6. (Optional. Available at the Advanced editing level.) If the application provides a constraint in the resolution request, specify the data type for the constraint. The constraint represents a condition that must or may be satisfied before the next stage of the resolution process can proceed.

```
[edit shared sae configuration nic-proxy-configuration ip resolution]
user@host# set constraints constraints
```

Related Documentation

- [Before You Configure a NIC Proxy on page 186](#)
- [Changing the Configuration for the NIC Proxy Cache \(SRC CLI\) on page 190](#)

- [Configuring a NIC Proxy for NIC Replication \(SRC CLI\) on page 191](#)
- [Configuration Statements for NIC Proxies on page 187](#)
- [NIC Proxy Configuration Overview on page 185](#)

Changing the Configuration for the NIC Proxy Cache (SRC CLI)

You can modify cache properties for the NIC proxy to optimize the resolution performance for your network configuration and system resources. Typically, you can use the default settings for the cache properties. The configuration statements are available at the Advanced editing level.

Use the following configuration statements to change values for the NIC proxy cache:

```
nic-proxy-configuration name cache {  
  cache-size cache-size;  
  cache-cleanup-interval cache-cleanup-interval;  
  cache-entry-age cache-entry-age;  
}
```

To configure the cache for a NIC proxy:

1. From configuration mode, access the configuration statement that specifies the NIC proxy configuration.

```
[edit]  
user@host# component-hierarchy nic-proxy-configuration name cache
```

For example:

```
[edit]  
user@host# edit shared sae configuration nic-proxy-configuration ip cache
```

2. Specify the maximum number of keys for which the NIC proxy retains data.

```
[edit shared sae configuration nic-proxy-configuration ip cache]  
user@host# set cache-size cache-size
```

If you decrease the cache size or disable the cache while the NIC proxy is running, the NIC proxy removes entries in order of descending age until the cache size meets the new limit.

3. Specify the time interval at which the NIC proxy removes expired entries from its cache.

```
[edit shared sae configuration nic-proxy-configuration ip cache]  
user@host# set cache-cleanup-interval cache-cleanup-interval
```

4. Specify how long an entry remains in the cache.

```
[edit shared sae configuration nic-proxy-configuration ip cache]  
user@host# set cache-entry-age cache-entry-age
```

- Related Documentation**
- [Before You Configure a NIC Proxy on page 186](#)
 - [Configuring Resolution Information for a NIC Proxy \(SRC CLI\) on page 188](#)
 - [Configuring a NIC Proxy for NIC Replication \(SRC CLI\) on page 191](#)
 - [Configuration Statements for NIC Proxies on page 187](#)
 - [NIC Proxy Configuration Overview on page 185](#)

Configuring a NIC Proxy for NIC Replication (SRC CLI)

Typically, you configure NIC replication to keep the NIC highly available. You configure NIC host selection to specify the groups of NIC hosts to be contacted to resolve a request, and to define how the NIC proxy handles NIC hosts that the proxy is unable to contact. The configuration statements are available at the Normal editing level.

Use the following configuration statements to configure NIC host selection for a NIC proxy:

```

nic-proxy-configuration name nic-host-selection {
    groups groups;
    selection-criteria (roundRobin | randomPick | priorityList);
}

nic-proxy-configuration name nic-host-selection blacklisting {
    try-next-system-on-error;
    number-of-retries-before-blacklisting number-of-retries-before-blacklisting;
    blacklist-retry-interval blacklist-retry-interval;
}

```

To configure a NIC proxy to use NIC replication:

1. From configuration mode, access the configuration statement that specifies the NIC proxy configuration.

```

[edit]
user@host# component-hierarchy nic-proxy-configuration name nic-host-selection

```

For example:

```

[edit]
user@host# edit shared sae configuration nic-proxy-configuration ip nic-host-selection

```

2. Specify the list of groups of NIC hosts that the NIC proxy can contact for resolution requests. Use commas to separate the group names.

```

[edit shared sae configuration nic-proxy-configuration ip nic-host-selection]
user@host# set groups groups

```

For example

```

[edit shared sae configuration nic-proxy-configuration ip nic-host-selection]
user@host# set groups [group1 group2]

```

3. If you configure more than one group, specify the selection criteria that the NIC proxy uses to determine which NIC host to contact.

```
[edit shared sae configuration nic-proxy-configuration ip nic-host-selection]
user@host# set selection-criteria (roundRobin | randomPick | priorityList)
```

where:

- roundRobin—NIC proxy selects NIC hosts in a fixed, cyclic order. The NIC proxy always selects the next host in the list.
- randomPick—NIC proxy selects NIC hosts randomly from the list.
- priorityList—NIC proxy selects NIC hosts according to their assigned priorities in the list. If the host with the highest priority in the list is not available, the NIC proxy tries the host with the next-highest priority, and so on.

Priorities are defined by the order in which you specify the groups. You can change the order of NIC hosts in the list by using the **insert** command.

4. Access the configuration statement that specifies the NIC proxy configuration for blacklisting—the process of handling nonresponsive NIC hosts.

```
[edit shared sae configuration nic-proxy-configuration ip nic-host-selection]
user@host# edit blacklisting
[edit shared sae configuration nic-proxy-configuration ip nic-host-selection blacklisting]
```

5. Specify whether or not the NIC proxy should contact the next specified NIC host if a NIC host is determined to be unavailable.

```
[edit shared sae configuration nic-proxy-configuration ip nic-host-selection blacklisting]
user@host# set try-next-system-on-error
```

6. (Optional) Change the number of times the NIC proxy tries to communicate with a NIC host before the NIC proxy stops communicating with the NIC host for a period of time. The default is 3.

```
[edit shared sae configuration nic-proxy-configuration ip nic-host-selection blacklisting]
user@host# set number-of-retries-before-blacklisting
number-of-retries-before-blacklisting
```

7. (Optional) Change the interval at which the NIC proxy attempts to connect to an unavailable NIC host. The default is 15 seconds.

```
[edit shared sae configuration nic-proxy-configuration name nic-host-selection
blacklisting]
user@host# set blacklist-retry-interval blacklist-retry-interval
```

Related Documentation

- [Before You Configure a NIC Proxy on page 186](#)
- [Configuring Resolution Information for a NIC Proxy \(SRC CLI\) on page 188](#)
- [Changing the Configuration for the NIC Proxy Cache \(SRC CLI\) on page 190](#)
- [Configuration Statements for NIC Proxies on page 187](#)

- [NIC Proxy Configuration Overview on page 185](#)

Configuring NIC Test Data (SRC CLI)

To test a resolution without NIC, you can configure a NIC proxy stub to take the place of the NIC. The NIC proxy stub comprises a set of explicit mappings of data keys and values in the NIC proxy configuration. When the SAE (or another SRC component configured to use a NIC proxy stub) passes a specified key to the NIC proxy stub, the NIC proxy stub returns the corresponding value. When you use a NIC proxy stub, no NIC infrastructure is required.

For example, you can specify a subscriber's IP address that is associated with a particular SAE. When the SRC component passes this IP address to the NIC proxy stub, the NIC proxy stub returns the corresponding SAE.

To use the NIC proxy stub for the SAE:

1. In configuration mode, navigate to the NIC proxy configuration and specify the type of key you want to map to a value.

```
[edit shared sae configuration nic-proxy-configuration name]
user@host# set resolution key-type key-type
```

For example, to specify the key ip for the ip NIC proxy configuration:

```
[edit shared sae configuration nic-proxy-configuration ip]
user@host# set resolution key-type ip
```

2. Enable a NIC proxy stub for a resolution.

```
[edit shared sae configuration nic-proxy-configuration ip]
user@host# set test-nic-bindings user-test-bindings
```

3. Specify the values of the keys for testing. These statements are available at the Expert CLI editing level.

```
[edit shared sae configuration nic-proxy-configuration ip]
user@host# set test-nic-bindings key-values name value
```

where:

- ***name***—Indicates the NIC data value for the proxy.
- ***value***—Specifies a value for the NIC data type.

For example, to set up a login name to IP mapping for login name jane@virneo.com to the IP address 192.0.2.30:

```
[edit shared sae configuration nic-proxy-configuration ip]
user@host# set test-nic-bindings key-values jane@virneo.com 192.0.2.30
```

For example, to set up an IP to SAE ID mapping for IP address 190.0.2.30 to SAE ID identified by the URL for the CORBA IOR corbaloc::10.227.7.145:8801/SAE:

```
[edit shared sae configuration nic-proxy-configuration ip]
user@host# set test-nic-bindings key-values 192.0.2.30
corbaloc::10.20.7.145:8801/SAE
```



NOTE: The SAE writes the value of the CORBA IOR to the `var/run` directory. The IP address in the corbaloc URL can be adjusted to the IP address or DNS name of the SAE.

You can use the key **ANY_KEY** to match any key for any key type. For example, if you want all IP addresses to resolve to the same SAE:

```
[edit shared sae configuration nic-proxy-configuration ip]
user@host# set test-nic-bindings key-values ANY_KEY corbaloc::10.20.7.145:8801/SAE
```

**Related
Documentation**

- [Planning a NIC Implementation on page 139](#)
- [NIC Configuration Scenarios on page 140](#)
- [High Availability for NIC on page 137](#)
- [Configuring NIC Test Data \(C-Web Interface\)](#)

Developing Applications That Use NIC

- [External Application Requirements for NIC on page 195](#)
- [External Non-Java Applications That Use NIC on page 195](#)
- [Creating a NIC Locator to Include with a Non-Java Application on page 196](#)
- [External Java Applications That Use NIC on page 197](#)
- [Developing a Java Application to Communicate with a NIC Proxy on page 198](#)
- [Updating Information About Address Pools on page 202](#)

External Application Requirements for NIC

If you write an external application to use NIC to perform a resolution, you can include NIC functionality in one of the following ways:

- For non-Java applications, use the interface module `NicAccess`, an IDL file that provides access to the NIC locator feature. The NIC locator can resolve the value of one or more keys.
- For Java applications, include the NIC proxy client libraries to use NIC in client/server mode.
- For Java applications, include the NIC proxy client libraries and the NIC host client libraries to use NIC in local host mode.

Related Documentation

- [External Non-Java Applications That Use NIC on page 195](#)
- [External Java Applications That Use NIC on page 197](#)
- [Creating a NIC Locator to Include with a Non-Java Application on page 196](#)

External Non-Java Applications That Use NIC

If you write an application in a language other than Java, you can use the NIC access interface module, a simplified CORBA interface, to perform one or more resolutions. By using this interface you can access through CORBA NIC locators, NIC proxies that run within the NIC host. The configuration properties for NIC locators are similar to those for NIC proxies in applications such as aggregate services and the sample residential portal.

Related Documentation

- For information about the NIC access interface module, see the API documentation on the Juniper Networks Web site at <http://www.juniper.net/techpubs/software/management/src/api-index.html>.
- [External Application Requirements for NIC on page 195](#)
- [External Java Applications That Use NIC on page 197](#)
- [Creating a NIC Locator to Include with a Non-Java Application on page 196](#)

Creating a NIC Locator to Include with a Non-Java Application

A NIC locator provides the same functionality as a NIC proxy, but is designed to work with non-Java applications.

You use the NIC access interface module to include NIC locators with your application by compiling the IDL file with your application files.

To use the NIC access interface module to create NIC locators:

1. Connect to the directory.
2. Obtain a CORBA reference to the NIC access interface from one of the following:
 - The access IOR provided in the directory in the dynamic configuration DN under the hostname—typically, *host/demohost*.
 - A corbaloc URL in the format:

`corbaloc::<host>:8810/Access`

3. From the NIC access interface module, obtain a NIC locator, as identified by `NicFeature`. For example:

```
feature = access.getLocatorFeature(nicNameSpace); //nicNameSpace example "/nicLocators/ip"
```

In the NIC configuration scenarios, the syntax for a NIC locator is `/nicLocators/<NIC key type>` where.

- **nicLocators**— Specifies all of the NIC locators in a NIC host.
 - **<NIC key type>**— Specifies the type of data that the key provides for the NIC resolution, such as ip, login, DN.
4. Search for the key. For example:

```
feature.lookupSingle(NicLocatorKey key) //NicLocatorKey is coming from the IDL
```

Related Documentation

- For information about the NIC access interface module, see the API documentation on the Juniper Networks Web site at <http://www.juniper.net/techpubs/software/management/src/api-index.html>.
- [External Java Applications That Use NIC on page 197](#)

- [External Application Requirements for NIC on page 195](#)
- [External Non-Java Applications That Use NIC on page 195](#)

External Java Applications That Use NIC

If you write an external Java application that interacts with a NIC, include NIC libraries in the application. These libraries are for NIC proxies and local NIC hosts. These libraries are located in the **SDK+AppSupport+Demos+Samples.tar.gz** on the Juniper Networks Web site at: <https://www.juniper.net/support/products/src/index.html>. You can locate the files in the *SDK/lib/nic* directory.

Typically, each NIC resolution process requires one NIC proxy. For example, the OnePopLogin sample data includes two resolution processes:

- Mapping of a subscriber's IP address to the subscriber's login name
- Mapping of the subscriber's login name to the SAE reference

An application that uses both these resolution processes would require two NIC proxies.

The NIC proxy provides a simple Java interface, the NIC application programming interface (API). You configure the NIC proxy to communicate with one resolver. For efficiency if you use NIC in client/server mode, the NIC proxy caches the results of resolution requests so it can respond to future requests for the same key without contacting the resolver.

The SRC software includes a factory interface, the NIC factory, to allow applications to instantiate, access, and remove NIC proxies. It also includes JAR files for NIC client and NIC host libraries.

You must configure an application to communicate with a NIC proxy.

If you are using Java Runtime Environment (JRE) 1.3 or higher, you must include in your application the Java archive (JAR) files, available in the **SDK+AppSupport+Demos+Samples.tar.gz** file on the Juniper Networks Web site at: <https://www.juniper.net/support/products/src/index.html>. The files are located in the */SDK/lib/* directory.

Related Documentation

- For more information about the API calls, see the online documentation on the Juniper Networks Web site at <http://www.juniper.net/techpubs/software/management/src/api-index.html>
- [External Application Requirements for NIC on page 195](#)
- [External Non-Java Applications That Use NIC on page 195](#)
- [Creating a NIC Locator to Include with a Non-Java Application on page 196](#)

Developing a Java Application to Communicate with a NIC Proxy

Configuration tasks that use the API calls to communicate with the NIC proxy are:

- [Instantiating a Configuration Manager on page 198](#)
- [Passing a Reference to the Configuration Manager to the NIC Factory on page 198](#)
- [Instantiating the NIC Factory Class on page 198](#)
- [Initializing Logging on page 199](#)
- [Instantiating the NIC Proxy on page 199](#)
- [Managing a Resolution Request on page 200](#)
- [Deleting Invalid Results from the NIC Proxy's Cache on page 201](#)
- [Removing the NIC Proxies on page 202](#)

Instantiating a Configuration Manager

The application must instantiate a configuration manager.

To enable the application to instantiate a configuration manager to obtain a NIC instance from the NIC factory:

- Call one of the following methods:
 - For some applications (other than Web applications), in which you must define the system property `-DConfig.bootstrapFilename`, you can call the following method:

```
ConfigMgr configMgr = ConfigMgrFactory.getConfigMgr();
```
 - For Web applications, you can instantiate the configuration manager as follows:

```
ConfigMgr configMgr = ConfigMgrFactory.getConfigMgr(properties);
```

 - `properties`—`java.util.Properties` object, typically the bootstrap file, which contains all the configuration properties for the NIC proxy.

Passing a Reference to the Configuration Manager to the NIC Factory

To pass a reference to the configuration manager to the NIC factory class:

- Call the following method in the application:

```
NicFactory.setConfigManager(configMgr);
```

Instantiating the NIC Factory Class

The way you instantiate the NIC factory depends on the object request broker (ORB) configuration:

- If the NIC proxy uses the default ORB, call the following method in the application:

```
NicFactory nicFactory = NicFactory.getInstance();
```

This code instantiates a new NIC factory. Unless the `NicFactory.destroy` method has been called, subsequent calls to this method will return the instantiated NIC factory.

- If the NIC proxy does not use the default ORB, call the following method:

```
NicFactory.initialize(props);
NicFactory nicFactory = NicFactory.getInstance();
```

- `props`—`java.util.Properties` object, which contains the ORB properties for the NIC proxy. For example, if the NIC proxy uses JacORB but JacORB is not the default ORB, the ORB properties are:

```
org.omg.CORBA.ORBClass=org.jacorb.orb.ORB
org.omg.CORBA.ORBSingletonClass=org.jacorb.orb.ORBSingleton
```

This code will instantiate a new NIC factory using the specified ORB. Unless the application has called the `NicFactory.destroy` method, subsequent calls to the `getInstance()` method will return the instantiated NIC factory. However, if the application has called the `destroy()` method, it must recall the `initialize()` method before it can call the `getInstance()` method.

For information about the `NicFactory.destroy` method, see [“Removing the NIC Proxies” on page 202](#).

Initializing Logging

You must initialize logging only if you want to view the logging information produced by the NIC proxy.

To enable the application to initialize logging:

- Call the following method:

```
Log.init(configMgr, configNameSpace);
```

- `configMgr`—Instance of the configuration manager, the value returned from the `getConfigMgr()` method
- `configNameSpace`—String that specifies the configuration namespace where you defined the logging properties
 - If you define the logging properties in the bootstrap file, specify the root namespace, `“/”`.

```
Log.init(configMgr, "/");
```

- If you define the logging properties in the directory, specify the namespace relative to the property `Config.net.juniper.smgmt.lib.config.staticConfigDN`, which you configure in the bootstrap file.

```
Log.init(configMgr, "/Applications/Quota");
```

Instantiating the NIC Proxy

To enable the application to instantiate a NIC proxy:

- Call the following method:

```
NIC nicProxy = nicFactory.getNicComponent(nicNameSpace, configMgr)
```

Alternatively, if the expected data value (specified for the property `nic.value` in the NIC proxy configuration) is an SAE reference, you can call the following method:

```
SaeLocator nicProxy = nicFactory.getSaeLocator(nicNameSpace, configMgr);
```

- `nicFactory`—Instance of the NIC factory
- `nicNameSpace`—String that specifies the configuration namespace where you defined the properties for the NIC proxy
 - If you define the NIC properties in the bootstrap file, specify the root namespace, `"/"`.

```
NIC nicProxy = nicFactory.getNicComponent("/", configMgr)
```

- If you define the properties in the directory, specify the namespace relative to the property `Config.net.juniper.sgmt.lib.config.staticConfigDN`, which you specified in the bootstrap file.

```
NIC nicProxy = nicFactory.getNicComponent("/Applications/Quota", configMgr)
```

- `configMgr`—Instance of the configuration manager, the value returned from the `getConfigMgr()` method

Managing a Resolution Request

To enable the application to submit a resolution request and obtain the associated values:

1. Construct a `NicKey` object to enable the application to pass the data key to the NIC proxy:

```
NicKey nicKey = new NicKey(stringKey);
```

- `stringKey`—Data key for which you want to find corresponding values.

For the syntax of allowed data types, see [“NIC Resolution Process Overview” on page 203](#).

2. If the resolution process specifies constraints that you wish to provide in the resolution request, add them to the `NicKey` object:

```
NicKey.addConstraint(constName, constValue);
```

- `constName`—Name of the constraint.

For the allowed data types and their syntax, see [“NIC Resolution Process Overview” on page 203](#).

- `constValue`—Specific value of the constraint.

For the allowed syntax for the data types, see [“NIC Resolution Process Overview” on page 203](#).

3. Call a method that starts the resolution process.

For example, you can call a method specified in the NIC interface:

```
NicValue val = nicProxy.lookupSingle(nicKey);
```

Alternatively, if the expected data value is an SAE reference, you can call the following method:

```
Saeld saeld = nicProxy.lookupSae(nicKey);
```

4. Call the `getValue` method to access the string representation of the data value obtained by the NIC proxy.

```
String val=val.getValue();
```

Alternatively, if the expected data value is an SAE reference:

```
String val=saeld.getValue();
```

5. (Optional) Call a method to get intermediate values obtained during a resolution.
 - Call the `getIntermediateValue` method if the application expects only one value. This method takes the name of a data type and returns as a string the first value it finds.

```
String getIntermediateValue(String dataTypeName){};
}
```

For information about data types, see [“NIC Resolution Process Overview” on page 203](#).

- Call the `getIntermediateValues` or `getAllIntermediateValues` method if the application expects multiple values. These methods take the name of a data type and return values as follows:
 - The `getIntermediateValues` method returns a list of values as a string array.

```
String[] getIntermediateValues(String dataTypeName){};
```

- For information about data types, see [“NIC Resolution Process Overview” on page 203](#)
- The `getAllIntermediateValues` method returns a map of all intermediate values for the request. The key for the map is the name of the network data type, and the value of the map is a string array of the intermediate values.

```
Map getAllIntermediateValues();
```

Deleting Invalid Results from the NIC Proxy's Cache

If the application receives an exception when using values that the NIC proxy returned for a specific key, it must inform the NIC proxy to delete this entry from its cache.

To enable the application to inform the NIC proxy to delete an entry from its cache:

- Call the following method:


```
nicProxy.invalidateLookup(nicKey, nicValue);
```

 - `nicKey`—Data key that you want to remove from the cache

- `nicValue`—Optional data value that corresponds to this key

If the application passes a null data value to the NIC proxy, the NIC proxy removes all the values associated with the data key from its cache.

Removing the NIC Proxies

Make sure that before your application shuts down, it removes the NIC proxy instances to release resources for other software processes.

To remove one NIC proxy instance:

- Call the following method:

```
NicProxy.destroy();
```

To remove all NIC proxy instances, call the following method:

```
NicFactory.destroy();
```

Related Documentation

- [NIC Configuration Scenarios Overview on page 209](#)
- [Configuring NIC to Store Log Messages in a File \(C-Web Interface\)](#)
- [Constraints as NIC Data Types on page 206](#)
- [NIC Data Types on page 204](#)
- [External Application Requirements for NIC on page 195](#)
- [Configuring NIC to Store Log Messages in a File \(C-Web Interface\)](#)

Updating Information About Address Pools

If you associate an existing address pool with an interface and you do not want to wait for this new information to be propagated based on the Cache Entry Age property of the NIC proxy or the Event Life Expectancy property of the agents, then you must manually clear the NIC proxy cache.

To clear the NIC proxy cache when an application is deployed in a J2EE container that supports Java Management Extension (JMX) software, do one of the following:

- Use the `NicProxyMgmt` MBean.
- Restart the application.
- Restart the application server.

Related Documentation

- [Deleting Invalid Results from the NIC Proxy's Cache on page 201](#)
- [Removing the NIC Proxies on page 202](#)
- [Passing a Reference to the Configuration Manager to the NIC Factory on page 198](#)
- [External Non-Java Applications That Use NIC on page 195](#)

CHAPTER 16

NIC Resolution Process

- [NIC Resolution Process Overview on page 203](#)
- [NIC Data Types on page 204](#)
- [Constraints as NIC Data Types on page 206](#)

NIC Resolution Process Overview

Because NIC can process all types of network data, you must use different resolution processes for different types of data mappings to maximize the performance of the NIC configuration. Resolving data requests consumes significant resources.

[Table 14 on page 203](#) shows the resolutions that the components in the NIC configuration scenarios perform. For customized types of resolutions, contact Juniper Networks Professional Services.

Table 14: Available NIC Resolutions

Key	Value
Subscriber's IP address (device running Junos OS)	SAE reference
Subscriber's IP address	Subscriber's login name
Subscriber's IP address	SAE reference
Subscriber's login name	SAE reference
Subscriber's username	SAE reference
Access DN	SAE reference

NIC Realms

Each resolution process and the resolvers that perform that process are defined by a *realm*—a group of resolvers that perform a series of resolution tasks to provide a mapping from a specified key to a specified data type. For example, the sample data provided for the NIC includes a realm called `dn` in which the resolution process takes an `access`

subscriber's distinguished name (DN) as the key and returns a reference to the SAE managing this subscriber as the value.

A set of hosts in a NIC can support multiple realms. Similarly, the agents in a NIC can support more than one realm. However, you can assign a resolver only to one realm.

A NIC host can support NIC resolvers for multiple realms. Consequently, you can simplify the NIC configuration and minimize the use of network resources by limiting the number of NIC hosts in your NIC configuration. NIC hosts can also handle multiple NIC resolvers in the same realm. In this case, when a NIC host receives a request, it chooses a NIC resolver as follows:

1. It identifies the NIC resolvers that are available to process the request.
2. If multiple NIC resolvers are available, it obtains a cost value associated with the resolution process from each resolver and selects the resolver that has the lowest cost value.

Key to Value Resolution

A resolution process typically defines several transitions or *roles*, with each transition resolving a NIC key to a value. For example, the resolution process to identify the SAE that manages a particular subscriber based on that subscriber's IP address involves the following roles:

1. Given the IP address, determine the IP address pool.
2. From the IP address pool, determine the VR.
3. From the VR, determine the SAE that manages that VR.

A role specifies the types of data with which it works. NIC supports a number of data types, including one that lets you add an identifier to other data types to let you specify different values for one data type.

For information about NIC data types, see [“NIC Data Types” on page 204](#) and [“Constraints as NIC Data Types” on page 206](#).

Related Documentation

- [Managing a Resolution Request on page 200](#)

NIC Data Types

The NIC supports the data types that appear in the following list. You can qualify these data types by adding an identifier to:

- Distinguish between different instances of a data type in a resolution scenario.
- Provide information about a data type to clarify the use of that data type in a resolution.

AnyString

- Generic data type to represent the information that you want to collect.
- Value—Alphanumeric characters
- Guidelines—You can qualify this data type with an identifier to provide information about the type of data that AnyString represents.
- Example—My(IP), My(Vr)

Dn

- DN of an access.
- Value—DN
- Example—*accessName=PrimaryAccess, enterpriseName=juniper, ou=Sunnyvale, retailerName=VPNprovider, o=Users, o=umc*

Domain

- Domain name.
- Value—Name of a domain
- Example—Example.net

Enterprise

- DN of an enterprise.
- Value—DN
- Example—*enterpriseName=juniper, ou=Sunnyvale, retailerName=VPNprovider, o=Users, o=umc*

Router

- Name of router.
- Value—Text string
- Example—router1

Interface

- Name of a router's interface. Can include a virtual routing forwarding identifier Vrfid). If a Vrfid is present, the DSA passes it to the SAE in an assignedIp request. The SAE uses the Vrfid to support IP addresses that may be the same across different VRFs.
- Value—`<interfaceName>/<ID>@<vrName>@ <routerName>`
`<interfaceName>#<vrId>@vrName@routerName`
- Example—FastEthernet4/1.0/4@boston@router1
fastEthernet4/1.0#vpn_a@boston@router1

InterfaceId

- Identifier of an interface.
- Value—<intfIndex>@<routerName>
- Example—4@router1

Ip

- Subscriber's IP address.
- Value—IP address
- Example—192.0.2.10

IpPool

- IP address pool.
- Value—Range of IP addresses enclosed in square brackets and parentheses
- Guidelines—If you enter an IP address that includes a value greater than 255 in one octet of the address, that part of the address is masked to fit the eight bits.
- Example—([192.0.2.0 192.0.2.255])

SaeId

- SAE reference.
- Value—CORBA interoperable object reference (IOR) for SAE
- Example—IOR:0000000000000002438444C3A736...

Vr

- Name of the virtual router.
- Value—<vrName>@<routerName>
- Example—vr1@router1

Constraints as NIC Data Types

Constraints are data types that a resolver uses when it executes a role. You can define:

- Multiple constraints for a role—Software performs an OR operation to determine whether the constraint is met.
- Multiple data types in a constraint—Software performs an AND operation to determine whether the multiple constraints are met.

Constraints can be either mandatory or optional. If a constraint is mandatory and the resolver for the role does not receive an appropriate value in the data request, the resolver must obtain the constraint value from other NIC resolvers. However, if a constraint is

optional and the resolver for the role does not receive an appropriate value in the data request, the resolver can execute its role without the constraint value. In this case, the resolver may obtain multiple values for the data key, and the NIC host responds to the NIC proxy as follows:

- If the request is for multiple results, the host provides all the results.
- If the request is for one result and the resolution process returns different results, the host returns an error message.
- If the resolution process returns multiple instances of the same result, the resolver provides only one result.

For example, if you want to obtain an SAE reference for a subscriber's IP address, you could define the following roles:

1. From the IP address, determine the VR (mandatory constraint IpPool).
2. From the VR, determine the SAE that manages that VR.

Because the first step has a mandatory constraint, the resolver for this role must use the IP pool supplied in the request, or obtain the IP pool from another resolver that determines IP pools from IP addresses. So you must define an extra step at the start of the resolution process:

1. From the IP address, determine the IP pool.
2. From the IP address, determine the VR (mandatory constraint IpPool).
3. From the VR, determine the SAE that manages that VR.

**Related
Documentation**

- [NIC Resolution Process Overview on page 203](#)
- [NIC Data Types on page 204](#)
- [Managing a Resolution Request on page 200](#)

CHAPTER 17

NIC Configuration Scenarios

- [NIC Configuration Scenarios Overview on page 209](#)
- [OnePop Scenario on page 210](#)
- [OnePopPcmm Scenario on page 212](#)
- [OnePopDynamicIp Scenario on page 214](#)
- [OnePopSharedIp Scenario on page 216](#)
- [OnePopStaticRouteIp Scenario on page 218](#)
- [OnePopVrfIp Scenario on page 221](#)
- [OnePopAcctId Scenario on page 223](#)
- [OnePopLogin Scenario on page 225](#)
- [OnePopLoginPull Scenario on page 227](#)
- [OnePopPrimaryUser on page 228](#)
- [OnePopDnSharedIp Scenario on page 230](#)
- [OnePopAllRealms Scenario on page 234](#)
- [OnePopTunnel Scenario on page 238](#)
- [OnePopPrefixIp Scenario on page 239](#)
- [MultiPop Scenario on page 240](#)

NIC Configuration Scenarios Overview

The NIC configuration scenarios in the sample data provide resolutions for a variety of network configurations.

Each NIC scenario includes two types of configuration:

- **Centralized**—A single host configuration for use with NIC replication. In a centralized configuration all agents and resolvers reside on one host. The name of this host is DemoHost.
- **Distributed**—A multiple host configuration in which agents and resolvers are distributed among more than one host. This type of configuration is designed for use with NIC host redundancy. In most cases, the hosts are named OnePopH1 (a host in a pop) and OnePopBO (a host in a back office).

The best way to view the sample data is with the NIC Web Admin tool.

For a summary of the NIC configuration scenarios included in the sample data, see [“NIC Configuration Scenarios” on page 140](#).

Related Documentation

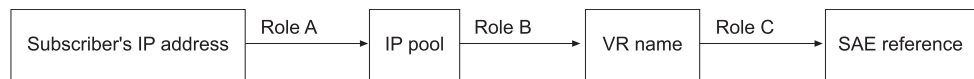
- [NIC Proxy Configuration Overview on page 185](#)
- [Before You Configure the NIC on page 151](#)
- [Configuration Statements for the NIC on page 149](#)
- [Configuring the NIC \(SRC CLI\) on page 152](#)

OnePop Scenario

The OnePop scenario illustrates a configuration that supports one POP. The realm for this configuration accommodates the situation in which IP address pools are configured locally on each VR. The resolution process takes a subscriber's IP address as the key and returns a reference to the SAE managing this subscriber as the value.

[Figure 15 on page 210](#) shows the resolution graph for this realm.

Figure 15: Resolution Process for ip Realm



g014923

The following agents collect information for resolvers in this realm:

- Directory agent PoolVr collects and publishes information about the mappings of IP address pools to VRs.
- Directory agent VrSaeld collects and publishes information about the mappings of VRs to SAEs.

The OnePop sample provides two host configurations: a centralized configuration and a distributed configuration. The OnePop Centralized configuration also provides an example of NIC host redundancy.

Centralized Configuration

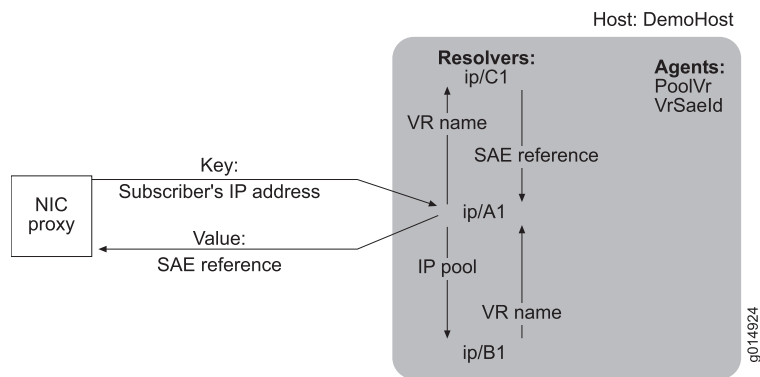
In this configuration, single host DemoHost supports all agents and resolvers. When the NIC proxy sends a subscriber's IP address to host DemoHost, the following sequence of actions occurs:

1. The host passes the IP address to resolver A1.
2. Resolver A1 obtains an IP pool for the IP address and forwards the request to resolver B1.
3. Resolver B1 obtains a VR name for the IP pool and returns the VR name to resolver A1.

4. Resolver A1 forwards the VR name to resolver C1.
5. Resolver C1 obtains an SAE reference for the VR and returns the VR identity to resolver A1.
6. Resolver A1 passes the SAE reference to its host.
7. The host returns the SAE reference to the NIC proxy.

Figure 16 on page 211 shows the interactions of the NIC components for this realm.

Figure 16: OnePop Centralized Configuration

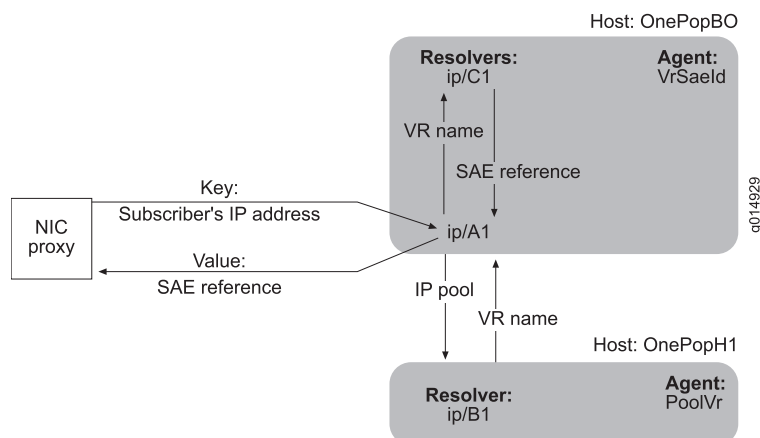


Distributed Configuration

In this configuration, the agents and resolvers are distributed among several hosts. When the NIC proxy sends a subscriber's IP address to host OnePopBO, the components execute the same actions as they do in the centralized configuration.

Figure 17 on page 211 illustrates the interactions of the NIC components for this realm.

Figure 17: OnePop Distributed Configuration

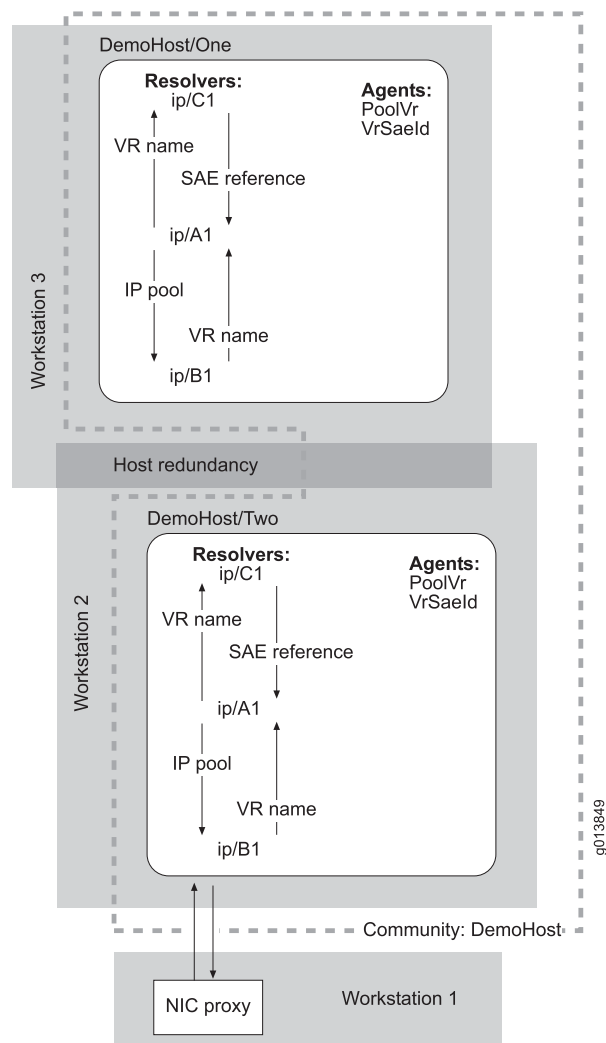


Redundancy

This sample data includes host redundancy for the centralized configuration. The hosts DemoHost/One and DemoHost/Two, which are installed on different machines, provide

host redundancy. These hosts form the community DemoHost, which does not include a monitor.

Figure 18: Redundancy for OnePop Centralized Configuration



- Related Documentation**
- [NIC Configuration Scenarios Overview on page 209](#)
 - [Configuring a NIC Scenario \(SRC CLI\) on page 156](#)

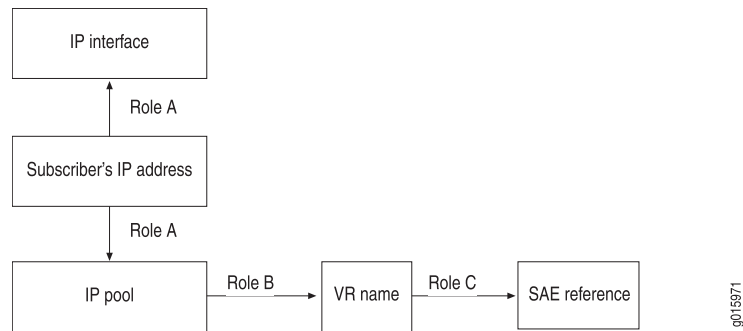
OnePopPcmm Scenario

This scenario is similar to the OnePop configuration scenario. It illustrates a configuration in which an assigned subscriber IP address managed by a network device such as a cable modem termination system (CMTS) device resolves to a reference to the SAE managing this subscriber. In this situation, the SAE acts as an application manager and interacts with the CMTS through a policy server.

The OnePopPcmm configuration scenario supports a PacketCable Multimedia Specification (PCMM) environment in which you use the assigned IP subscriber method to log in subscribers and in which you use the NIC to determine the subscriber's SAE. The realm for this configuration accommodates the situation in which IP pools are configured locally on each application manager group object. These IP pools represent an IP pools-managed policy decision point (PDP) group for one or more CMTS devices.

Figure 19 on page 213 shows the resolution graph for this realm.

Figure 19: Resolution Process for Pcmm_am Realm



This scenario uses the same agents as the OnePop scenario. For the OnePopPcmm configuration scenario, the agent collects information from the application manager object instead of the virtual router entry. A virtual router name is generated in the format "default"@<pdpGroup>.

The OnePopPcmm scenario provides two host configurations: a centralized configuration and a distributed configuration.

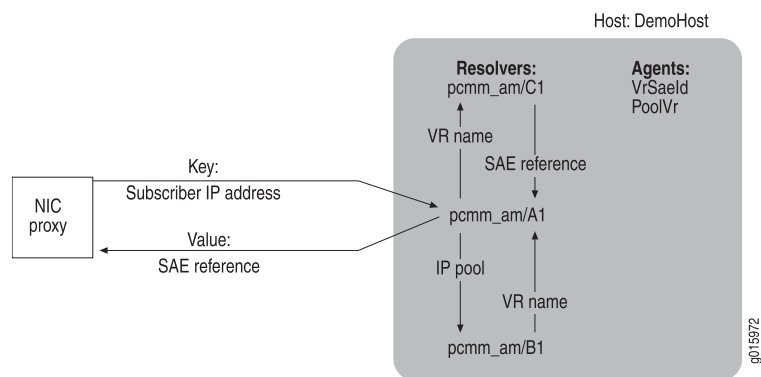
Centralized Configuration

In this configuration, the single host DemoHost supports all agents and resolvers. When a NIC proxy sends a subscriber's IP address to host DemoHost, the following sequence of actions occurs:

1. The host passes an assigned subscriber IP address resolver A1.
2. Resolver A1 obtains the IP pool name and the interface name, and forwards the request to resolver B1.
3. Resolver B1 obtains the VR name for the IP pool name and interface name, and returns the VR name to resolver A1.
4. Resolver A1 forwards the VR name to resolver C1.
5. Resolver C1 obtains an SAE reference for the VR and returns it to resolver A1.
6. Resolver A1 passes the SAE reference to its host.
7. The host returns the SAE reference to the NIC proxy.

Figure 20 on page 214 show the interactions of the NIC components for this realm.

Figure 20: OnePopPcmm Centralized Configuration

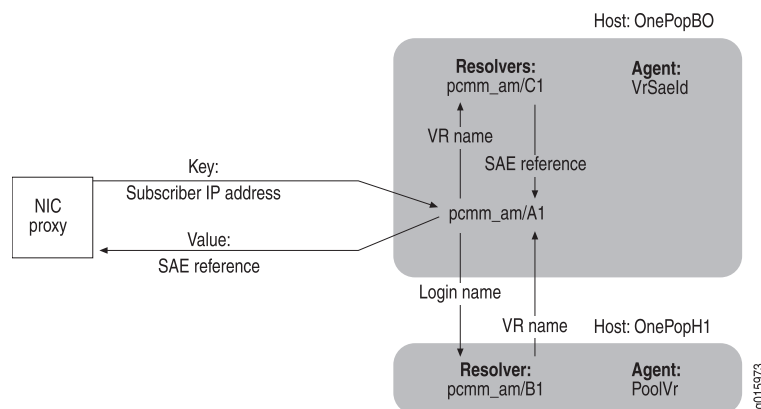


Distributed Configuration

In this configuration, the agents and resolvers are distributed among two hosts. When the NIC proxy sends a subscriber's IP address to host OnePopBO, the components execute the same actions as they do in the centralized configuration.

Figure 21 on page 214 illustrates the interactions of the NIC components for this realm.

Figure 21: OnePopPcmm Distributed Configuration



- Related Documentation**
- [NIC Configuration Scenarios Overview on page 209](#)
 - [Configuring a NIC Scenario \(SRC CLI\) on page 156](#)

OnePopDynamicIp Scenario

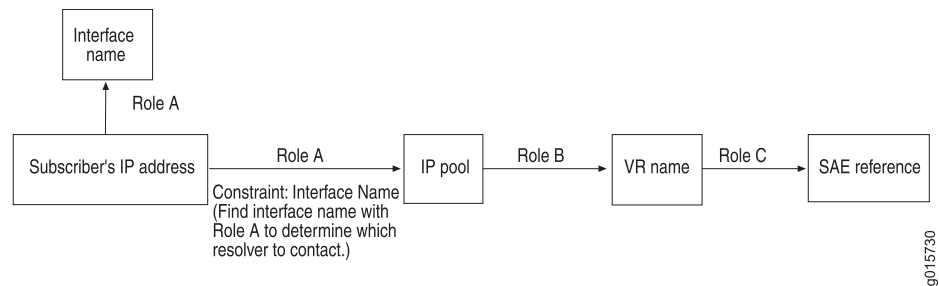
This scenario illustrates a configuration that is very similar to the OnePop scenario. The realm for this configuration accommodates the situation in which IP address pools are configured locally on each virtual router object. The resolution process takes a subscriber's IP address as the key and returns a reference to the SAE managing this subscriber as the value.

The scenario supports a configuration scenario for a PacketCable Multimedia Specification (PCMM) environment in which you use the assigned IP subscriber method to log in

subscribers, and use the NIC to determine the subscriber's SAE. In this scenario, the SAE acts as a combined application manager and policy server; it directly manages CMTS devices.

Figure 22 on page 215 shows the resolution graph for this realm.

Figure 22: Resolution Process for dynamicIp Realm



The following agents collect information for resolvers in this realm:

- Directory agent PoolVr collects and publishes information about the mappings of IP address pools to VRs.
- Directory agent VrSaeld collects and publishes information about the mappings of VRs to SAEs.

The OnePopDynamicIp scenario provides two host configurations: a centralized configuration and a distributed configuration.

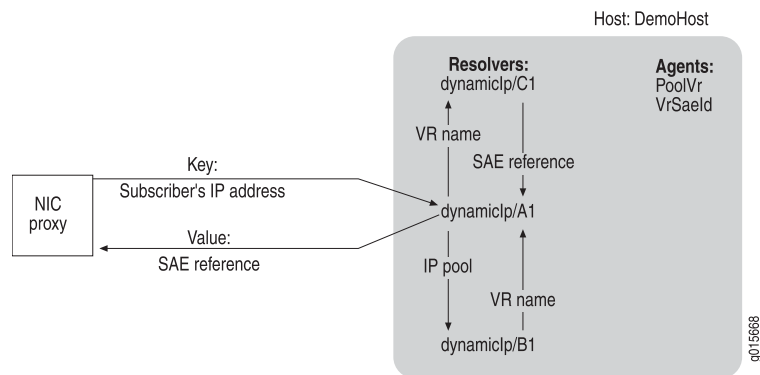
Centralized Configuration

In this configuration, single host DemoHost supports all agents and resolvers. When the NIC proxy sends a subscriber's IP address to host DemoHost, the following sequence of actions occurs:

1. The host passes the IP address to resolver A1.
2. Resolver A1 obtains an IP pool name and interface name for the IP address, and forwards the request to resolver B1.
3. Resolver B1 obtains a VR name for the IP pool name and interface name, and returns the VR name to resolver A1.
4. Resolver A1 forwards the VR name to resolver C1.
5. Resolver C1 obtains an SAE reference for the VR and returns the VR identity to resolver A1.
6. Resolver A1 passes the SAE reference to its host.
7. The host returns the SAE reference to the NIC proxy.

Figure 23 on page 216 illustrates the interactions of the NIC components for this realm.

Figure 23: OnePopDynamicIp Centralized Configuration

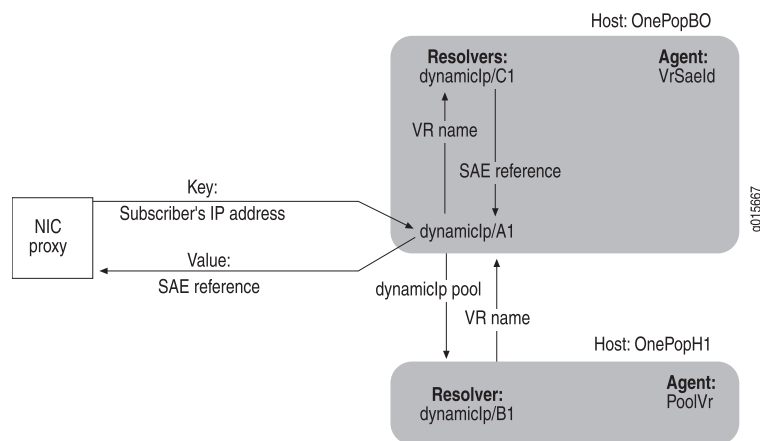


Distributed Configuration

In this configuration, the agents and resolvers are distributed among several hosts. When the NIC proxy sends a subscriber's IP address to host OnePopBO, the components execute the same actions as they do in the centralized configuration.

Figure 24 on page 216 illustrates the interactions of the NIC components for this realm.

Figure 24: OnePopDynamicIp Distributed Configuration



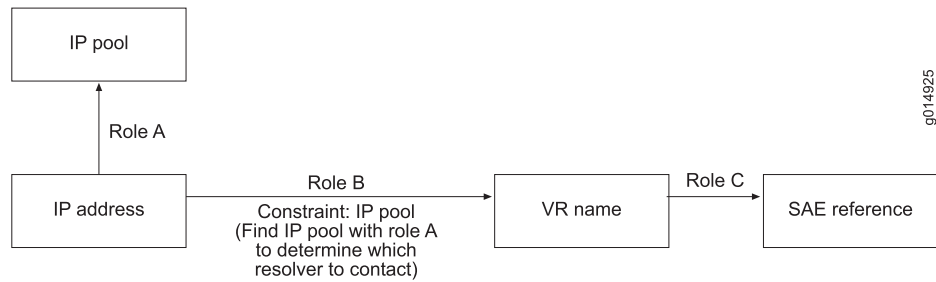
- Related Documentation**
- [NIC Configuration Scenarios Overview on page 209](#)
 - [Configuring a NIC Scenario \(SRC CLI\) on page 156](#)

OnePopSharedIp Scenario

This scenario illustrates a configuration that is very similar to the OnePop scenario. However, the realm for this configuration accommodates the situation in which IP address pools are shared by VRs in the same POP. The resolution process takes a subscriber's IP address as the key and returns a reference to the SAE managing this subscriber as the value.

Figure 25 on page 217 shows the resolution graph for this realm.

Figure 25: Resolution Process for sharedIp Realm



g014925

The following agents interact with resolvers in this realm:

- SAE plug-in agent IpVr collects and publishes information about the mappings of IP addresses to VRs.
- Directory agent PoolVr collects and publishes information about the IP address pools used by the VRs in a POP. Because the IP address pools are shared between VRs, this agent discards information about VRs.
- Directory agent VrSaeld collects and publishes information about the mappings of VRs to SAEs.

The OnePopSharedIP scenario provides two host configurations: a centralized configuration and a distributed configuration.

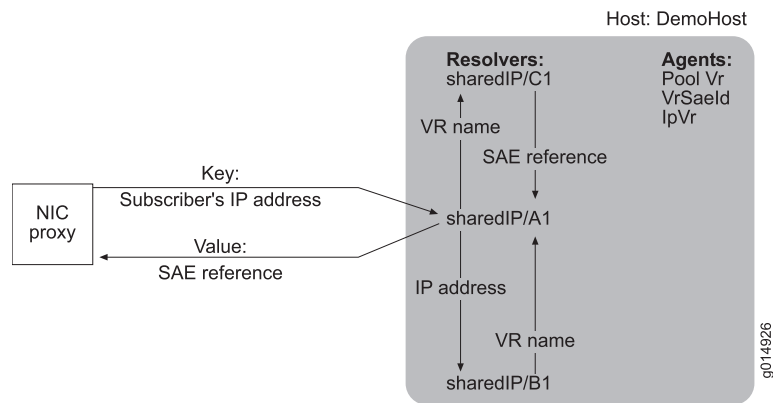
Centralized Configuration

In this configuration, single host DemoHost supports all agents and resolvers. When the NIC proxy sends a subscriber's IP address to host DemoHost, the following sequence of events occurs:

1. The host passes the IP address to resolver A1.
2. Resolver A1 obtains an IP pool for the IP address.
3. Resolver A1 forwards the IP address and the IP pool to resolver B1.
4. Resolver B1 obtains a VR name for the IP address and returns the VR name to resolver A1.
5. Resolver A1 forwards the VR name to resolver C1.
6. Resolver C1 obtains an SAE reference for the VR and returns the SAE reference to resolver A1.
7. Resolver A1 passes the SAE reference to its host.
8. The host returns the SAE reference to the NIC proxy.

Figure 26 on page 218 shows the interactions of the NIC components for this realm.

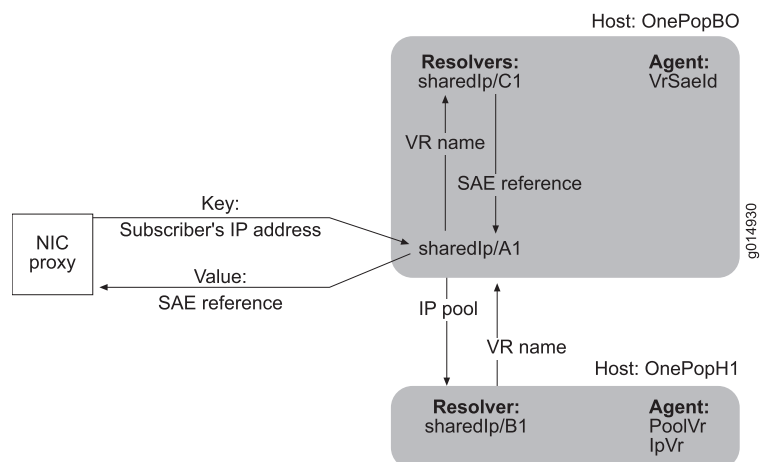
Figure 26: OnePopSharedIP Centralized Configuration



Distributed Configuration

In this configuration, the agents and resolvers are distributed among several hosts. When the NIC proxy sends a subscriber's IP address to the host OnePopBO, the resolvers execute the same actions as they do in the centralized configuration. [Figure 27 on page 218](#) illustrates the interactions of the NIC components for this realm.

Figure 27: OnePopSharedIP Distributed Configuration



- Related Documentation**
- [NIC Configuration Scenarios Overview on page 209](#)
 - [Configuring a NIC Scenario \(SRC CLI\) on page 156](#)

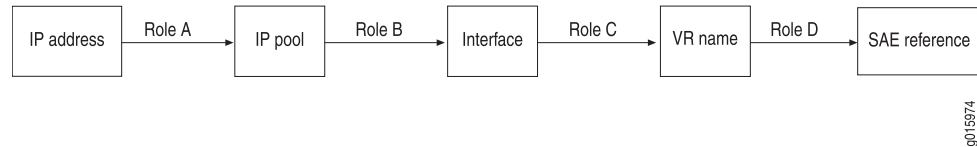
OnePopStaticRouteIp Scenario

The OnePopStaticRouteIp configuration scenario for NIC resolves an assigned IP address for a subscriber whose traffic enters the network through an interface on a device running Junos OS to a reference for the SAE that manages the interface. The realm for this configuration accommodates the situation in which the network publisher component gathers interface information for the devices running Junos OS. The resolution process

takes a subscriber's IP address as a key and returns a reference to the SAE that manages the interface.

Figure 28 on page 219 shows the resolution graph for this realm.

Figure 28: Resolution Process for the StaticRoutelp Realm



The following agents collect information for resolvers in this realm:

- Directory agent PoolInterface collects and publishes information about the mappings of IP address pools to interfaces.
- Directory agent VrSaeld collects and publishes information about the mappings of VRs to SAEs.

The agents obtain information from the interfaceConfiguration attribute of the EdgeRouter entry in the directory and read an XML document that conforms to the networkConfig.xsd schema. If this scenario is used with a different router type, you can edit the XML document.

For information about the XML document, see [“External Application Requirements for NIC” on page 195](#).

The OnePopStaticRoutelp scenario provides two host configurations: a centralized configuration and a distributed configuration.

Centralized Configuration

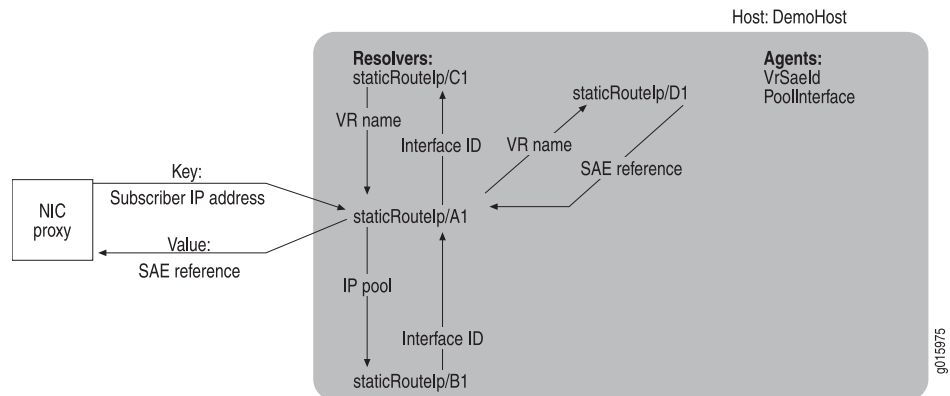
In this configuration, the single host DemoHost supports all agents and resolvers. When the NIC proxy sends a subscriber's IP address to host DemoHost, the following sequence of events occurs:

1. The host passes the subscriber's IP address to resolver A1.
2. Resolver A1 obtains an IP pool for the IP address.
3. Resolver A1 forwards the IP pool name to Resolver B1.
4. Resolver B1 obtains the interface ID for the IP pool and returns this value to resolver A1.
5. Resolver A1 forwards the interface ID to Resolver C1.
6. Resolver C1 resolves the interface ID to the VR name and returns the VR name to resolver A1.
7. Resolver A1 forwards the VR name to resolver D1.
8. Resolver D1 obtains a reference for the SAE managing the VR and returns the SAE reference to resolver A1.

9. Resolver A1 passes the SAE reference to its host.
10. The host returns the SAE reference to the NIC proxy.

Figure 29 on page 220 shows the interactions of the NIC components for this realm.

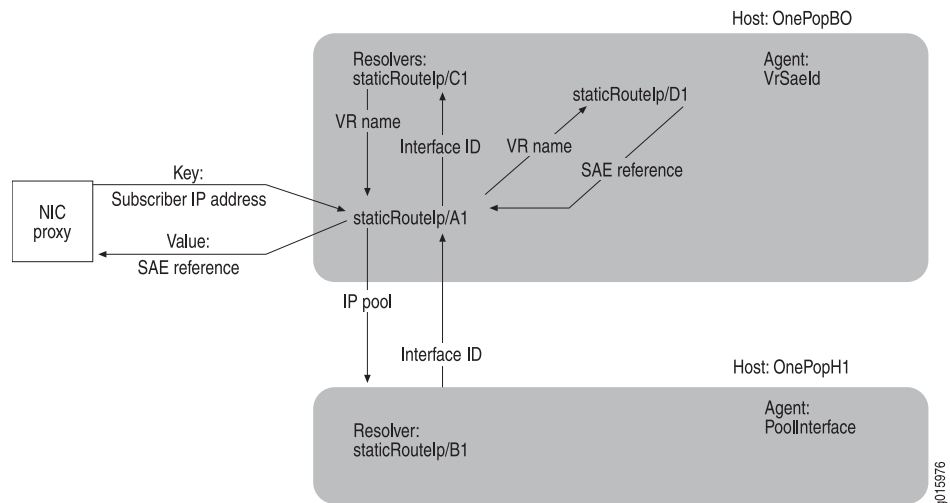
Figure 29: OnePopStaticRouteIp Centralized Configuration



Distributed Configuration

In this configuration, the agents and resolvers are distributed among two hosts. When a NIC proxy sends a subscriber IP address to host OnePopBO, the resolvers execute the same actions as they do in the centralized configuration. Figure 30 on page 220 illustrates the interactions of the NIC components for this realm.

Figure 30: OnePopStaticRouteIp Distributed Configuration



Related Documentation

- [NIC Configuration Scenarios Overview on page 209](#)
- [Configuring a NIC Scenario \(SRC CLI\) on page 156](#)

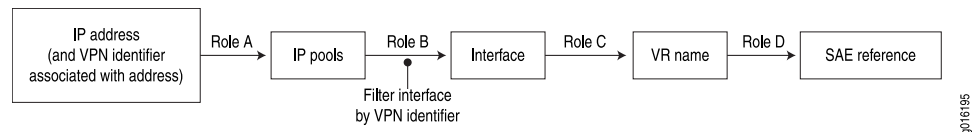
OnePopVrflp Scenario

The OnePopVrflp configuration scenario for NIC resolves an assigned IP address for a subscriber to IP pools or network whose traffic enters the network through an interface on a device running Junos OS to a reference for the SAE that manages the interface. The realm for this configuration utilizes routing information collected by the network publisher from particular devices running Junos OS. The resolution process takes a subscriber's IP address as a key and returns a reference to the SAE that manages the interface.

This configuration scenario is very similar to the OnePopStatic Routelp scenario. During resolution, the OnePopVrflp scenario filters interfaces the VPN identifier of the VPN that carries subscriber traffic.

Figure 31 on page 221 shows the resolution graph for this realm.

Figure 31: Resolution Process for the Vrflp Realm



The following agents collect information for resolvers in this realm:

- Directory agent PoolInterface collects and publishes information about the mappings of IP address pools to interfaces.
- Directory agent VrSaeld collects and publishes information about the mappings of VRs to SAEs.

The agents obtain information from the interfaceConfiguration attribute of the EdgeRouter entry in the directory and read an XML document that conforms to the networkConfig.xsd schema. If this scenario is used with a different router type, you can edit the XML document.

For information about the XML document, see [“Files Used to Test Network Publisher” on page 180](#).

The OnePopVrflp scenario provides two host configurations: a centralized configuration and a distributed configuration.

Centralized Configuration

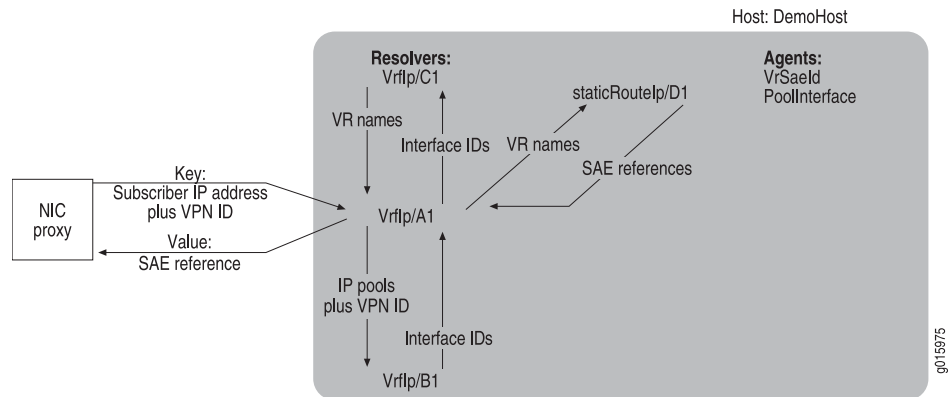
In this configuration, the single host DemoHost supports all agents and resolvers. When the NIC proxy sends a subscriber's IP address to host DemoHost, the following sequence of events occurs:

1. The host passes the subscriber's IP address and VPN ID to resolver A1.
2. Resolver A1 obtains all IP pools that match the IP address.
3. Resolver A1 forwards the IP pool names and VPN ID to Resolver B1.

4. Resolver B1 obtains the all interface IDs for the IP pools and filters all interfaces that match the VPN ID.
5. Resolver A1 forwards the interface IDs to Resolver C1.
6. Resolver C1 resolves the interface IDs to the VR name and returns the VR name to resolver A1.
7. Resolver A1 forwards the VR names to resolver D1.
8. Resolver D1 obtains references for the SAEs managing the VRs and returns the SAE reference to resolver A1.
9. Resolver A1 passes the SAE references to its host.
10. The host returns the SAE references to the NIC proxy.

Figure 32 on page 222 shows the interactions of the NIC components for this realm.

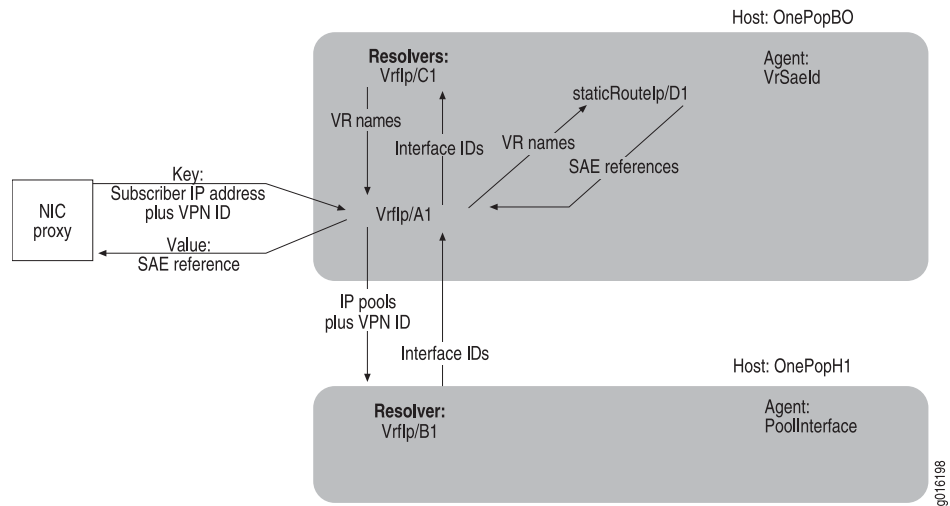
Figure 32: OnePopVrflp Centralized Configuration



Distributed Configuration

In this configuration, the agents and resolvers are distributed among two hosts. When a NIC proxy sends a subscriber IP address to host OnePopBO, the resolvers execute the same actions as they do in the centralized configuration. Figure 33 on page 223 illustrates the interactions of the NIC components for this realm.

Figure 33: OnePopStaticRouteIp Distributed Configuration

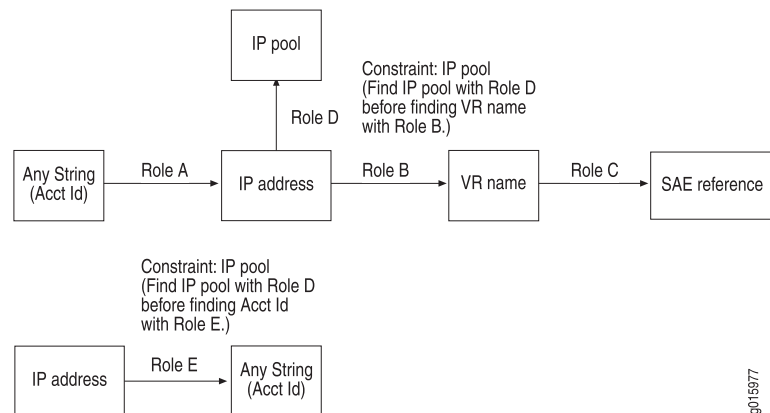


OnePopAcctId Scenario

This scenario illustrates a configuration in which subscribers have an accounting ID, as defined by the LDAP attribute `accountingUserId` or the plug-in attribute `PA_ACCOUNTING_ID`. The realms for this configuration accommodate two independent resolution processes, which can be used by the SRC Volume-Tracking Application (SRC VTA).

Figure 34 on page 223 shows the resolution graphs for this realm.

Figure 34: Resolution Process for acctId Realm



The following agents collect information for resolvers in this realm:

- Directory agent PoolVr collects and publishes information about the mappings of IP address pools to VRs.
- Directory agent VrSaeld collects and publishes information about the mappings of virtual routers and the mappings between virtual routers and SAEs.

- SAE plug-in agent AcctIdIp collects and publishes information about the mappings of accounting IDs of subscribers to subscriber IP addresses.
- SAE plug-in agent IpAcctId collects and publishes information about the mappings of subscriber IP addresses to accounting IDs.

The OnePopAcctId scenario provides one host for a centralized configuration. In this configuration the single host DemoHost supports all agents and resolvers. Two NIC proxies are associated with the configuration. One NIC proxy (called acct-sae in this description) submits accounting IDs, and another NIC proxy (called addr-acct in this description) submits subscribers' IP addresses.

When the NIC proxy sends an accounting ID to host DemoHost, the following sequence of events occurs:

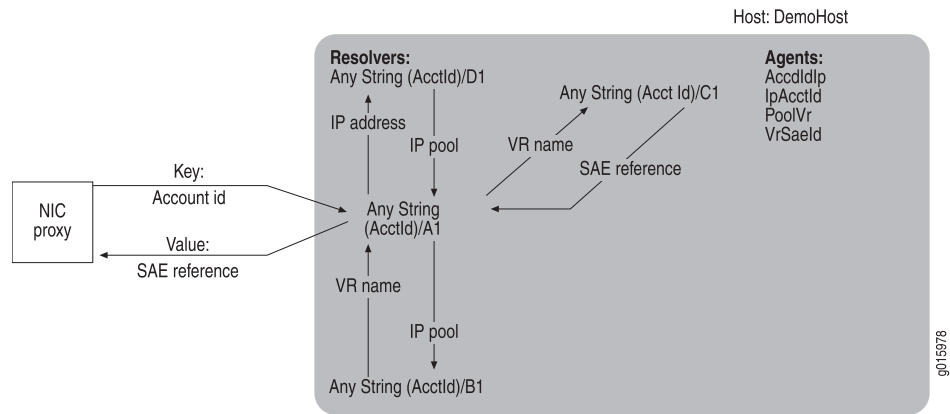
1. The host passes the subscriber's accounting ID to resolver A1.
2. Resolver A1 obtains an IP address for the account ID.
3. Resolver A1 forwards the IP address to Resolver D1.
4. Resolver D1 obtains the IP pool for the IP address and returns it to Resolver A1.
5. Resolver A1 forwards the IP address and IP pool to Resolver B1.
6. Resolver B1 obtains the VR name and return it to resolve A1.
7. Resolver A1 forwards the VR name to resolver C1.
8. Resolver C1 obtains the SAE reference for the VR name and returns it to resolver A1.
9. Resolver A1 passes the SAE reference to its host.
10. The host returns the SAE reference to the NIC proxy acct-sae.

When the NIC proxy sends an IP address to host DemoHost, the following sequence of events occurs:

1. The host passes the subscriber's IP address to resolver A1.
2. Resolver A1 forwards the IP address to resolver D1.
3. Resolver D1 obtains the IP pool for the IP address and returns it to resolver A1.
4. Resolver A1 forwards the IP address and IP pool to resolver C1.
5. Resolver C1 obtains the accounting ID for the IP address and associated IP pool and returns the accounting Id to resolver A1.
6. Resolver A1 passes the accounting ID to its host.
7. The host returns the accounting ID to the NIC proxy addr-acct.

[Figure 35 on page 225](#) illustrates the interactions of the NIC components for this realm.

Figure 35: OnePopAcctId Centralized Configuration



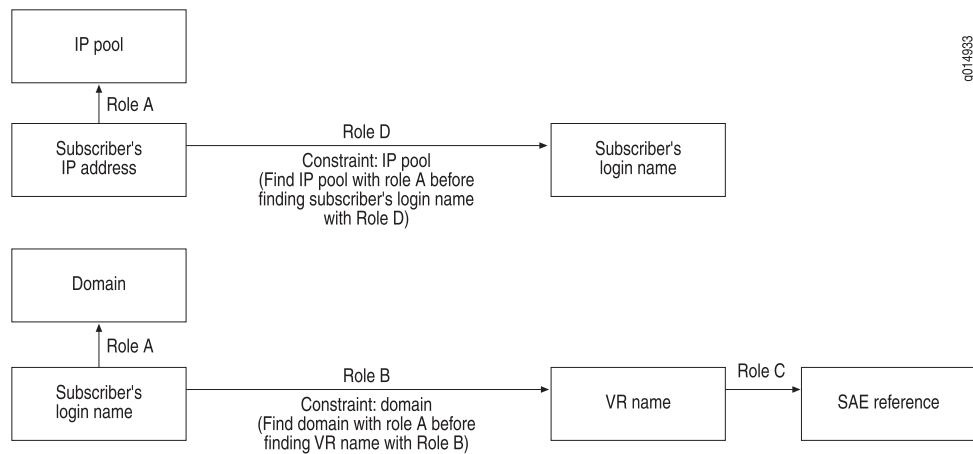
- Related Documentation**
- [NIC Configuration Scenarios Overview on page 209](#)
 - [Configuring a NIC Scenario \(SRC CLI\) on page 156](#)

OnePopLogin Scenario

This scenario illustrates a configuration that is very similar to the OnePop scenario. The realm for this configuration accommodates two independent resolution processes, which are used by the SRC Volume Tracking Applications (SRC VTAs) and may be used for other purposes.

Figure 36 on page 225 shows the resolution graphs for this realm.

Figure 36: Resolution Processes login Realm



The following agents interact with resolvers in this realm:

- SAE plug-in agent **IpLoginName** collects and publishes information about the mappings of IP addresses to login names.
- SAE plug-in agent **LoginNameVr** collects and publishes information about the mappings of login names to VRs.

- Directory agent Pool collects and publishes information about the IP address pools used by the VRs in a POP. The agent uses the information about the IP address pools to determine which resolver to communicate with, rather than communicating with all resolvers that are running role D.
- Directory agent VrSaeld collects and publishes information about the mappings of VRs to SAEs.

The OnePopLogin scenario provides two host configurations: a centralized configuration and a distributed configuration.

Centralized Configuration

In this configuration, single host DemoHost supports all agents and resolvers. Two NIC proxies are associated with this NIC configuration; one NIC proxy (called NIC proxy 1 in this documentation) submits subscribers' login names, and the other (called NIC proxy 2 in this documentation) submits subscribers' IP addresses.

When NIC proxy 1 sends a login name to the host DemoHost, the following sequence of events occurs:

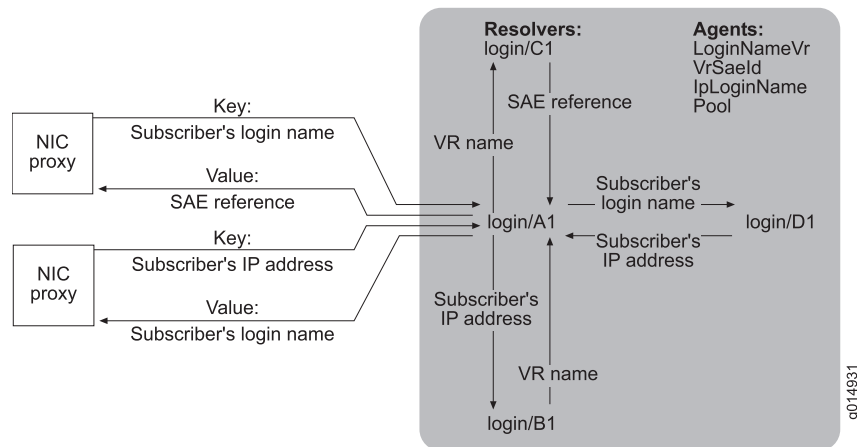
1. The host passes the login name to resolver A1.
2. Resolver A1 obtains a domain name for the login name.
3. Resolver A1 forwards the login name and the domain to resolver B1.
4. Resolver B1 obtains a VR name for the login name and returns the VR name to resolver A1.
5. Resolver A1 forwards the VR name to resolver C1.
6. Resolver C1 obtains an SAE reference for the VR and returns the SAE reference to resolver A1.
7. Resolver A1 returns the SAE reference to its host.
8. The host returns the SAE reference to the NIC proxy.

When NIC proxy 2 sends a subscriber's IP address to host DemoHost, the following sequence of events occurs.

1. The host passes the IP address to resolver A1.
2. Resolver A1 obtains an IP pool for the IP address.
3. Resolver A1 forwards the IP address and the IP pool to resolver D1.
4. Resolver D1 obtains a login name for the IP address and returns the login name to resolver A1.
5. Resolver A1 passes the login name to its host.
6. The host returns the login name to the NIC proxy.

[Figure 37 on page 227](#) illustrates the interactions of the NIC components for this realm.

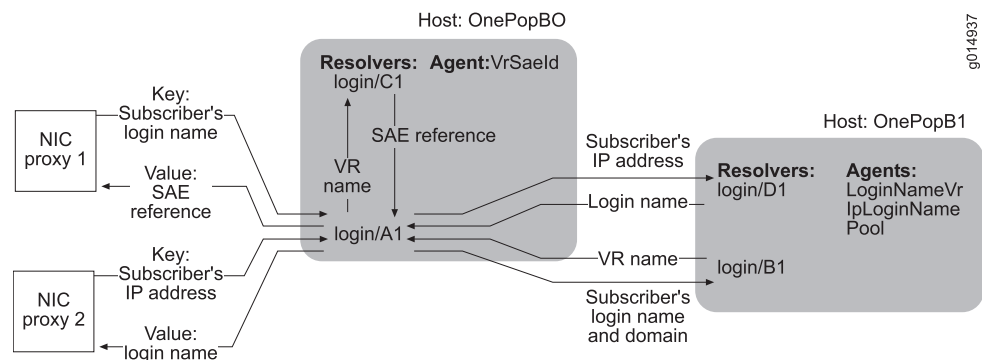
Figure 37: OnePopLogin Centralized Configuration



Distributed Configuration

In this configuration, the agents and resolvers are distributed among several hosts. When the NIC proxy sends a subscriber's IP address to the host OnePopBO, the resolvers execute the same actions as they do in the centralized configuration. [Figure 38 on page 227](#) illustrates the interactions of the NIC components for this realm.

Figure 38: OnePopLogin Distributed Configuration

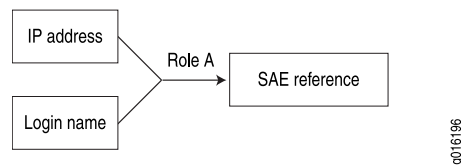


- Related Documentation**
- [NIC Configuration Scenarios Overview on page 209](#)
 - [Configuring a NIC Scenario \(SRC CLI\) on page 156](#)

OnePopLoginPull Scenario

The OnePopLoginPull configuration scenario provides a simple NIC resolution from a subscriber login name or IP address to an SAE reference.

[Figure 39 on page 228](#) shows the resolution graph for this scenario.

Figure 39: OnePopLoginPull Distributed Configuration

In the OnePopLoginPull scenario, SAE client agents read entries under *o=umc*, *o=servers*, *o=sspadminurls* in the Juniper Networks database to determine which SAEs are active. They also periodically check if other SAEs have become active. The SAE external interface for the active SAEs determines which SAE has a user session for the subscriber identified either by login identifier or IP identifier.

The OnePopLoginPull scenario includes the following SAE client agents:

- LoginSaeld
- IpSaeld

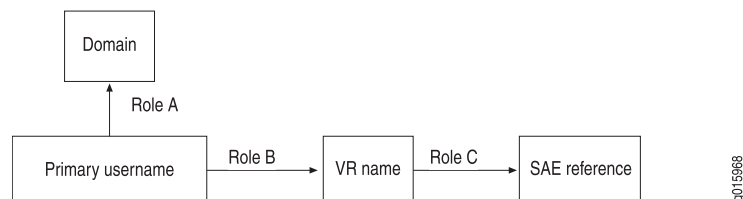
Related Documentation

- [NIC Configuration Scenarios Overview on page 209](#)
- [Configuring a NIC Scenario \(SRC CLI\) on page 156](#)

OnePopPrimaryUser

The OnePopPrimaryUser scenario is similar to one of the resolutions in the OnePopLogin scenario. In the OnePopPrimaryUser scenario, subscriber primary username, as identified by the PA_PRIMARY_USER_NAME attribute, is resolved to a reference for a managing SAE. The realm for this configuration accommodates a situation in which a NIC proxy provides a primary username.

[Figure 40 on page 228](#) show the resolution graph for this realm.

Figure 40: Resolution Processes for primary_user Realm

The following agents interact with resolvers in this realm:

- Directory agent VrSaeld collects and publishes information about virtual routers and the mappings between virtual routers and SAEs.
- SAE plug-in agent UserNameVr collects and publishes information about the mappings of subscriber primary usernames to VR names.

The OnePopPrimaryUser scenario provides two host configurations: a centralized configuration and a distributed configuration.

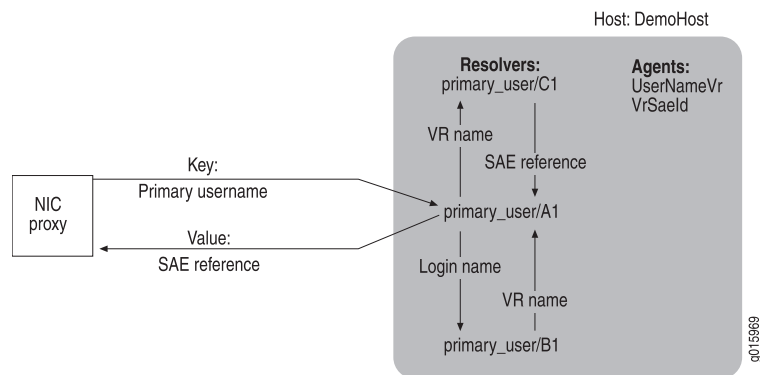
Centralized Configuration

In this configuration, a single host called DemoHost supports all agents and resolvers. When a NIC proxy send a subscriber's primary username to host Demo Host, the following sequence of events occurs:

1. The host passes the primary username to resolver A1.
2. (Optional) Resolver A1 resolves the primary username to its domain.
3. Resolver A1 forwards the primary username to resolver B1.
4. Resolver B1 obtains the name of the VR associated with the subscriber's primary username and returns the VR to resolver A1.
5. Resolver A1 forwards the VR to resolver C1.
6. Resolver C1 obtains the SAE reference for the SAE managing the VR and returns the SAE reference to resolver A1.
7. Resolver A1 returns the SAE reference to the host.
8. The host returns the SAE reference to the NIC proxy.

Figure 41 on page 229 illustrates the interactions of the NIC components for this realm.

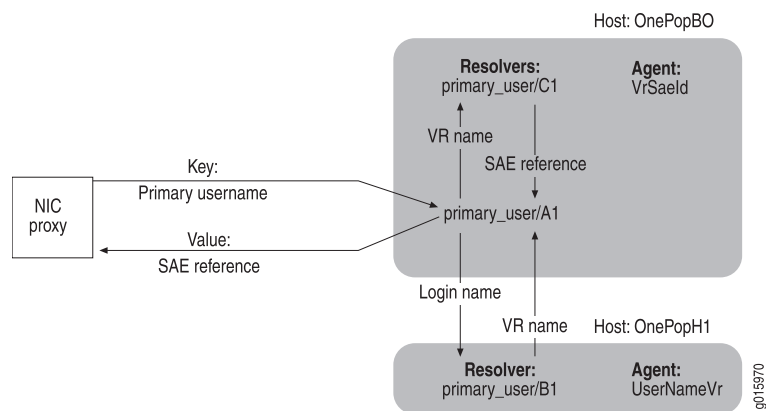
Figure 41: OnePopPrimaryUser Centralized Configuration



Distributed Configuration

In this configuration, the agents and resolvers are distributed among two hosts. When a NIC proxy sends a subscriber's primary username to the host OnePopBO, the resolvers execute the same actions as they do in the centralized configuration. Figure 42 on page 230 illustrates the interactions of the NIC components for this realm.

Figure 42: OnePopPrimaryUser Distributed Configuration



- Related Documentation**
- [NIC Configuration Scenarios Overview on page 209](#)
 - [Configuring a NIC Scenario \(SRC CLI\) on page 156](#)

OnePopDnSharedIp Scenario

The OnePopDnSharedIp scenario illustrates how to configure SAE plug-in agents that have state synchronization enabled to support an SAE plug-in that uses state synchronization. This scenario uses the same centralized and distributed configurations of hosts as the OnePop scenario.

Two realms are configured:

- Shared IP

The resolution process is identical to that for the OnePop scenario.

- DN realm

This realm uses essentially the same resolution process as the MultiPop DN realm. However, some of the constraints differ.

This realm also uses the same agents as the MultiPop DN realm. The names of agents and resolvers are essentially the same as those in the MultiPop configuration, although they do not include a POP identifier. [Figure 43 on page 231](#) illustrates the centralized configuration, and [Figure 44 on page 233](#) illustrates the distributed configuration for the DN realms.

The configuration for the two realms is similar to the configuration for the shared IP and DN realms in the OnePopAllRealms scenario. .

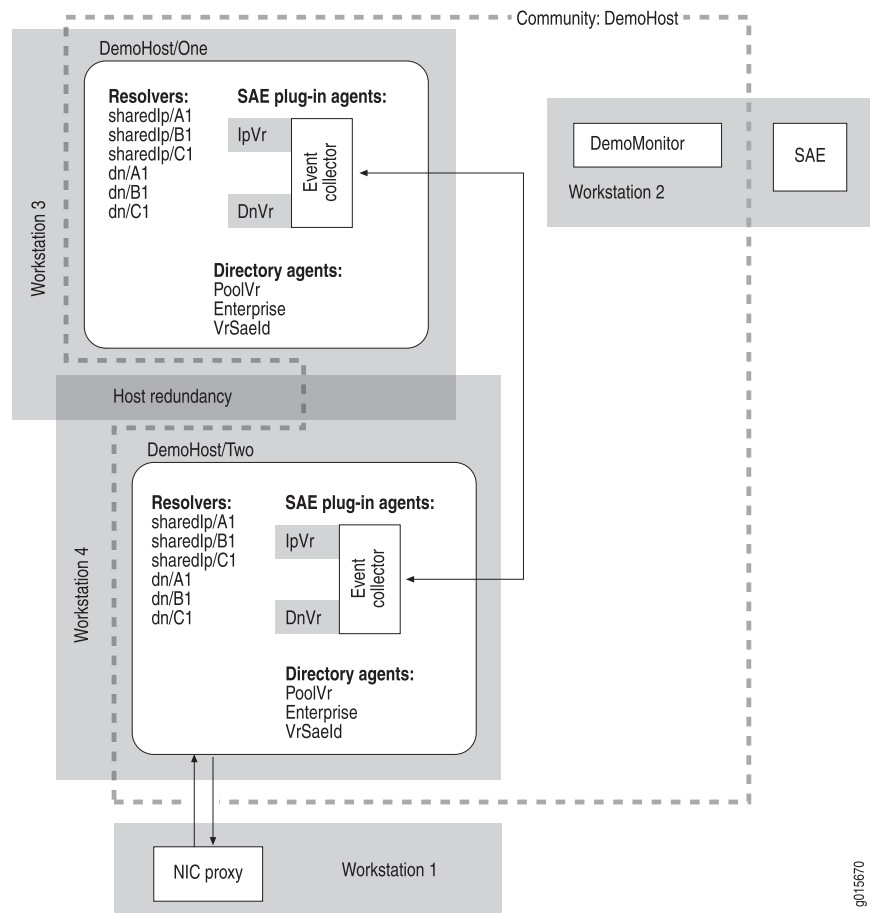
The OnePopAllRealms illustrates SAE plug-in agents configured to use SAE plug-in redundancy rather than SAE plug-in agents.

Centralized Configuration

Figure 43 on page 231 shows the centralized configuration for the scenario. Host DemoHost supports all resolvers and agents. The two SAE plug-in agents, IpVr and DnVr, share an event collector. Both plug-in agents have state synchronization enabled.

DemoHost is also configured for redundancy. Its redundant hosts (DemoHost/One and DemoHost/Two) perform the host function. The redundant hosts are on different machines, and both hosts support the resolvers and agents assigned to the parent host. The redundant hosts form a community called DemoHost with the monitor DemoMonitor, which tracks them.

Figure 43: OnePopDnSharedIp Realms Centralized Configuration



Distributed Configuration

Figure 44 on page 233 shows the distributed configuration from the scenario. Host OnePopBO supports two resolvers for each realm and a directory agent that is used by different realms. Host OnePopH1 supports one resolver for each realm and agents that are used by different realms.

Both hosts also have a redundant configuration. The redundant hosts for OnePopBO (OnePopBO/One and OnePopBO/Two) perform the host function. The redundant hosts are on different machines, and both hosts support the resolvers and agents assigned to the parent host.

The redundant hosts for OnePopH1 (OnePopH1/One and OnePopH1/Two) perform the host function. The redundant hosts are on different machines, and both hosts support the resolvers and agents assigned to the parent host.

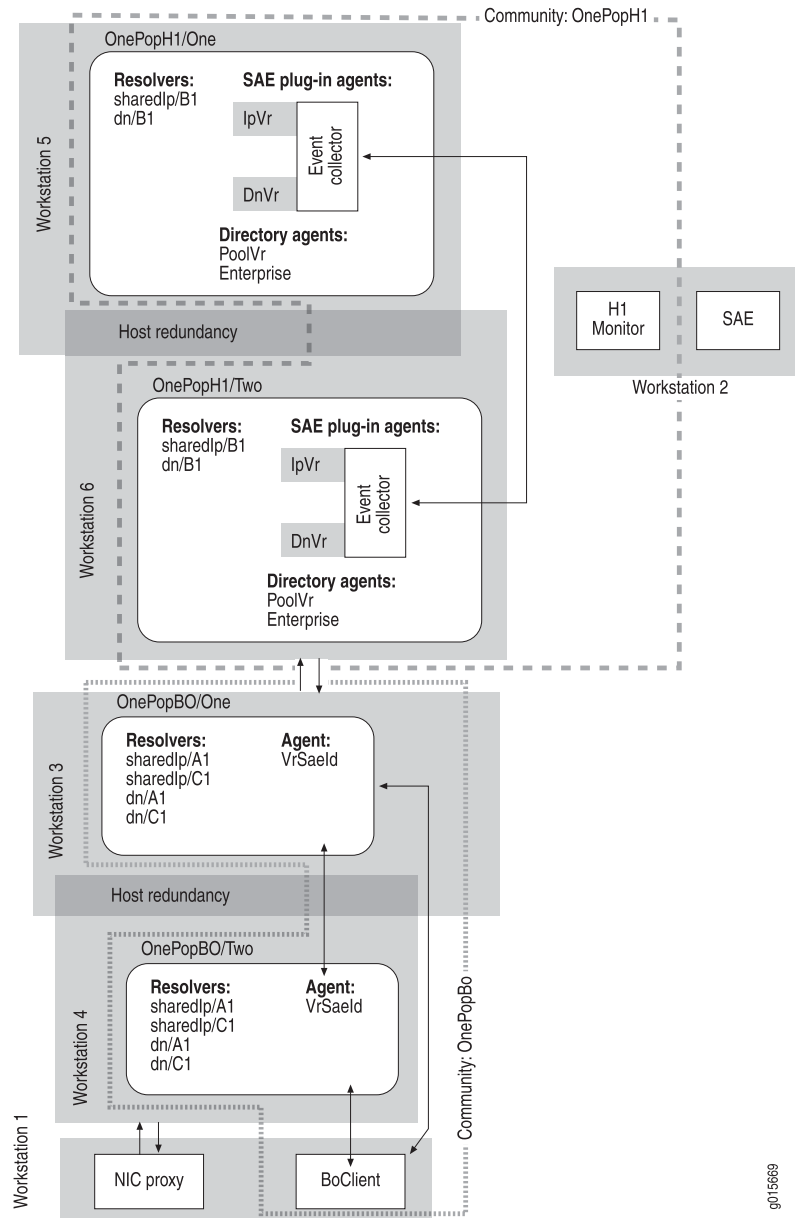
However, host OnePopH1 also supports two SAE plug-in agents, IpVr and DnVr, which share an event collector. These agents have state synchronization enabled.

The redundant hosts OnePopBO/One and OnePopBO/Two are members of a community called OnePopBO. This community supports the monitor, BoClient, which is installed on the machine that supports the NIC proxy. BoClient tracks the connections between the redundant hosts OnePopBO/One and OnePopBO/Two from the point of view of the NIC client (NIC proxy).

Similarly, the redundant hosts OnePopH1/One and OnePopH1/Two are members of a community called OnePopH1. This community has one monitor, H1Monitor, which is located on the same machine as the SAE and tracks the connections among the redundant hosts in the same community, their primary host, and the other hosts in the configuration.

H1Monitor comprises the monitor process OnePop, which is installed on the same machine as the SAE. BoClient comprises the monitor process OnePopClient, which is installed on the same machine as the NIC proxy.

Figure 44: OnePopDnSharedIp Realms Distributed Configuration



Related Documentation

- [NIC Configuration Scenarios Overview on page 209](#)
- [Configuring a NIC Scenario \(SRC CLI\) on page 156](#)
- [OnePop Scenario on page 210](#)
- [MultiPop Scenario on page 240](#)

OnePopAllRealms Scenario

The main purpose of the OnePopAllRealms scenario is to illustrate how to configure redundancy. This scenario uses the same centralized and distributed configurations of hosts as the OnePop scenario.

Three realms are configured:

- IP realm

This realm uses essentially the same resolution process as the IP realm for the OnePop scenario. However, some of the constraints differ.

- Shared IP

The resolution process is identical to that for the OnePopShared scenario.

- DN realm

This realm uses essentially the same resolution process as the MultiPop DN realm. However, some of the constraints differ.

This realm also uses the same agents as the MultiPop DN realm. The names of agents and resolvers are essentially the same as those in the MultiPop configuration, although they do not include a POP identifier. By reviewing the scenario, [Figure 45 on page 235](#) and [Figure 46 on page 237](#), you can determine exact pictures of the DN realms for the centralized and distributed configurations.

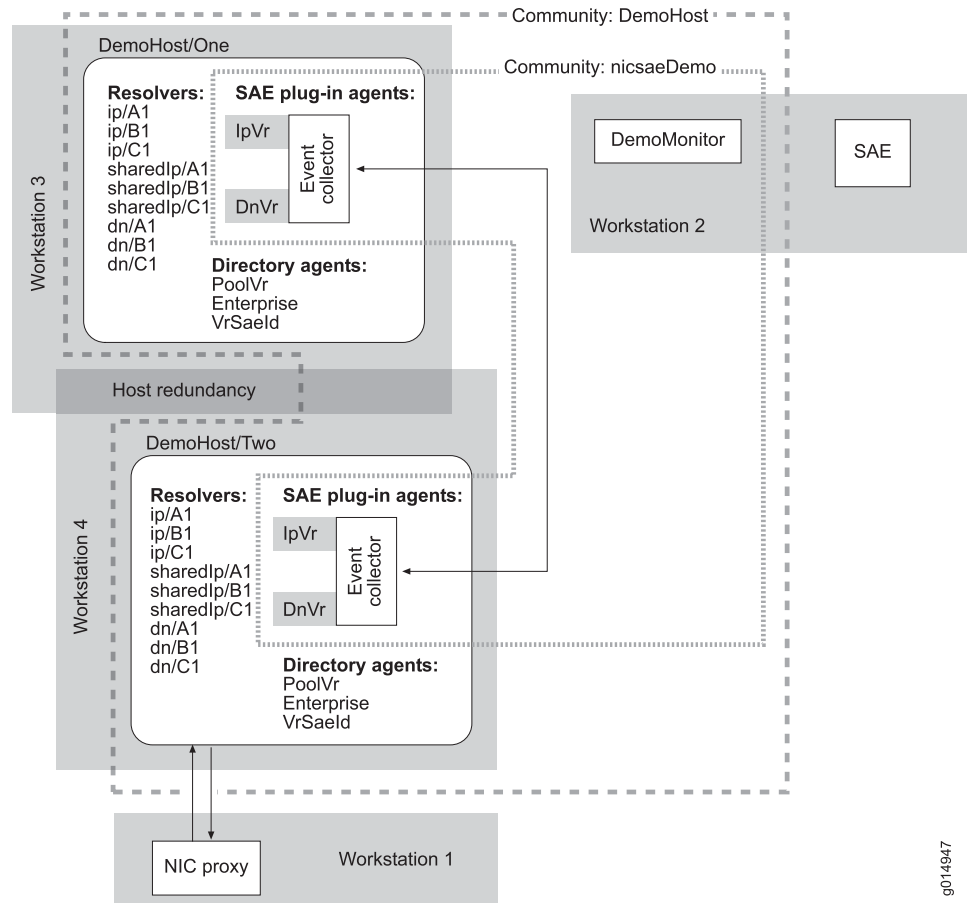
[Figure 45 on page 235](#) shows the centralized configuration for the scenario. Host DemoHost supports all resolvers and agents. However, because host DemoHost is configured for redundancy, its redundant hosts (DemoHost/One and DemoHost/Two) perform the host function. The redundant hosts are on different machines, and both hosts support the resolvers and agents assigned to the parent host.

The parent host DemoHost also supports two SAE plug-in agents, IpVr and DnVr, which share an event collector. Each SAE plug-in agent has a redundant agent called Demo; these redundant agents also share an event collector. The redundant agents and their shared event collector are assigned to both redundant hosts DemoHost/One and DemoHost/Two.

The redundant agents form a community called nicsaeDemo with the monitor DemoMonitor, which tracks them. The redundant agents are identified in the community by the names DemoHost/One and DemoHost/Two; these names specify their hosts and provide unique identifiers for the redundant agents.

The redundant hosts form a community called DemoHost with the monitor DemoMonitor, which tracks them.

Figure 45: OnePopAllRealms Centralized Configuration



g014947

Figure 46 on page 237 shows the distributed configuration for the scenario. Host OnePopBO supports two resolvers for each realm and a directory agent that is used by different realms. However, because host OnePopBO is configured for redundancy, its redundant hosts (OnePopBO/One and OnePopBO/Two) perform the host function. The redundant hosts are on different machines, and both hosts support the resolvers and agents assigned to the parent host.

Host OnePopH1 supports one resolver for each realm and agents that are used by different realms. Host OnePopH1 is also configured for redundancy, and its redundant hosts (OnePopH1/One and OnePopH1/Two) perform the host function. The redundant hosts are on different machines, and both hosts support the resolvers and agents assigned to the parent host.

However, host OnePopH1 also supports two SAE plug-in agents, IpVr and DnVr, which share an event collector. Each SAE plug-in agent has a redundant agent called onePop; these redundant agents also share an event collector. The redundant agents and their shared event collector are assigned to redundant hosts OnePopH1/One and OnePopH1/Two.

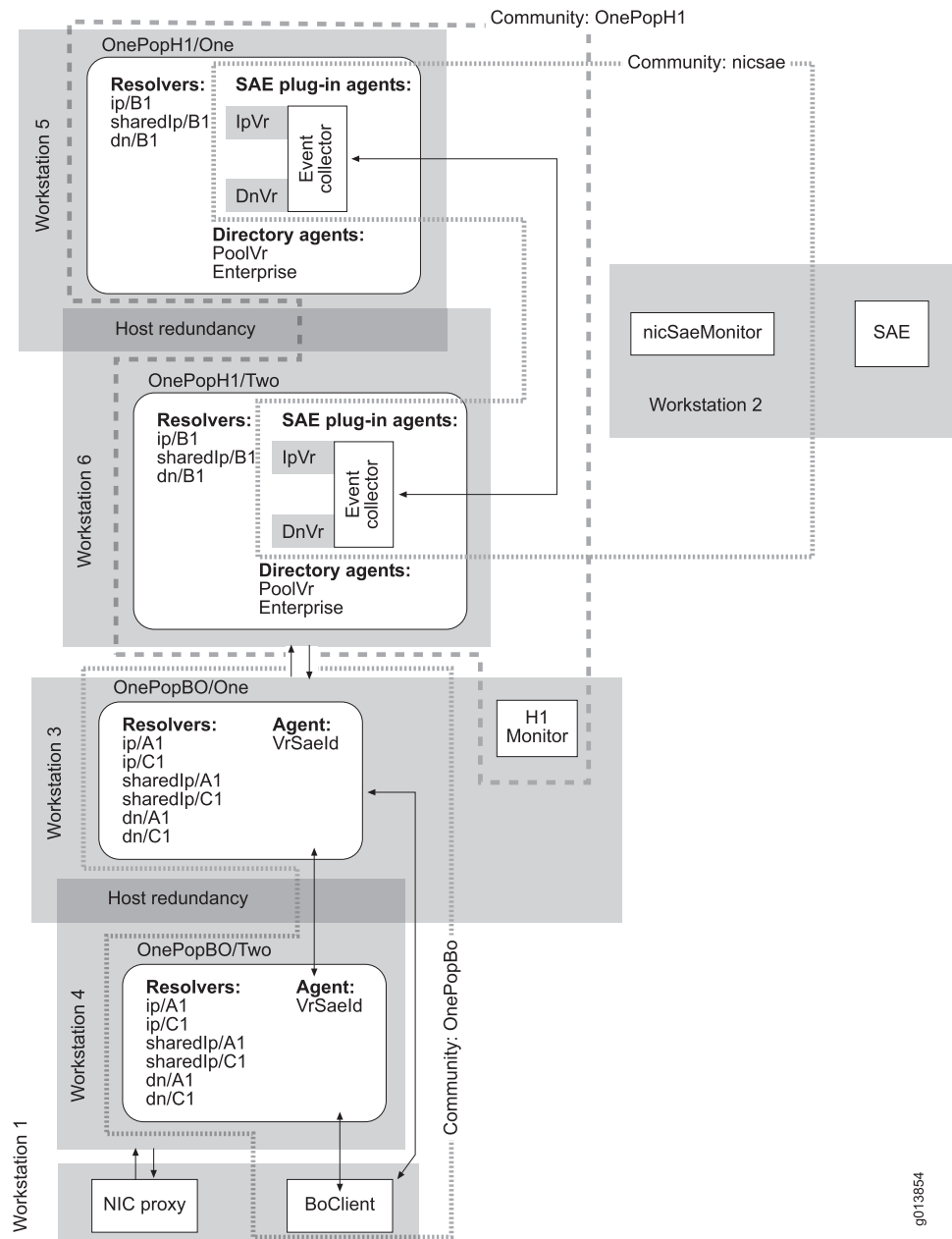
The redundant agents form a community called nicsae with monitor nicSaeMonitor, which tracks them. The redundant agents are identified in the community by the names OnePopH1/One and OnePopH1/Two; these names specify their hosts and provide unique identifiers for the redundant agents.

The redundant hosts OnePopBO/One and OnePopBO/Two are members of a community called OnePopBO. This community supports the monitor, BoClient, which is installed on the machine that supports the NIC proxy. BoClient tracks the connections between the redundant hosts OnePopBO/One and OnePopBO/Two from the point of view of the NIC client (NIC proxy).

Similarly, the redundant hosts OnePopH1/One and OnePopH1/Two are members of a community called OnePopH1. This community has one monitor, H1Monitor, which is located on the same machine as the SAE and tracks the connections among the redundant hosts in the same community, their primary host, and the other hosts in the configuration.

H1Monitor and nicSaeMonitor are part of the monitor process OnePop, which is also installed on the same machine as the SAE. BoClient is part of the monitor process OnePopClient, which is installed on the same machine as the NIC proxy.

Figure 46: OnePopAllRealms Distributed Configuration



g013854

Related Documentation

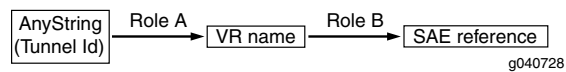
- [NIC Configuration Scenarios Overview on page 209](#)
- [Configuring a NIC Scenario \(SRC CLI\) on page 156](#)
- [OnePop Scenario on page 210](#)
- [OnePopSharedIp Scenario on page 216](#)
- [MultiPop Scenario on page 240](#)

OnePopTunnel Scenario

The OnePopTunnel scenario illustrates a configuration in which subscribers have a tunnel ID, as defined by the combination of the plug-in attributes PA_TUNNEL_ID, PA_TUNNEL_SESSION_ID, and PA_LAC_IP.

The resolution process takes a subscriber's Tunnel ID as the key and returns a reference to the SAE managing this subscriber as the value. [Figure 47 on page 238](#) depicts the resolution process for this scenario.

Figure 47: Resolution Process for Tunnel Realm



The following agents collect information for resolvers in this realm:

- The SAE plug-in agent TunnelIdVr collects and publishes information about the mapping of TunnelId (Tunnel ID + Tunnel Session ID + LAC IP Address) to VRs.
- The directory agent VrSaeld collects and publishes information about the mappings of VRs to SAEs.

The OnePopTunnel sample provides a centralized configuration.

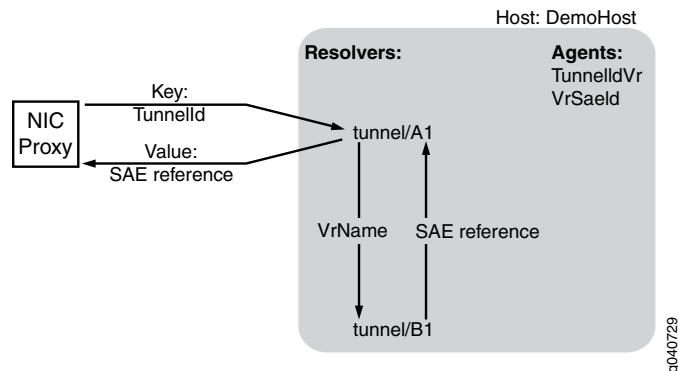
Centralized Configuration

In this configuration, a single host (DemoHost) supports all agents and resolvers. When the NIC proxy sends a subscriber's TunnelId to host DemoHost, the following sequence of actions occurs:

1. The host passes the tunnel ID to resolver A1.
2. Resolver A1 obtains a VR name for the tunnel ID.
3. Resolver A1 forwards the VR name to resolver B1.
4. Resolver B1 obtains an SAE reference for the VR and returns the VR identity to resolver A1.
5. Resolver A1 passes the SAE reference to its host.
6. The host returns the SAE reference to the NIC proxy.

[Figure 48 on page 239](#) illustrates the interactions of the NIC components for this realm.

Figure 48: OnePopTunnel Centralized Configuration



- Related Documentation**
- [NIC Configuration Scenarios Overview on page 209](#)
 - [Configuring a NIC Scenario \(SRC CLI\) on page 156](#)

OnePopPrefixIp Scenario

This section describes the NIC functionality required to support the IPv6 feature. Subscribers are identified by a set of IPv6 prefixes defined by the device. These IPv6 prefixes are made available to the NIC through SAE IPv6 plug-in attributes (`framedIpv6Prefix` and `delegatedIpv6Prefix`) and these attributes can be present for the same subscriber session. The SAE reports the IPv6 plug-in attributes to the NIC and any IP address starting with one of these IPv6 prefixes identifies the user session in the SAE.

The OnePopPrefixIp scenario supports the IPv6 feature. This scenario enables applications to identify subscribers based on their IP addresses and obtain a reference to the SAE managing these subscribers.

The OnePopPrefixIp configuration scenario is very similar to the OnePop scenario. In the OnePopPrefixIp scenario, the IP pool information is provided by the SAE (through NIC SAE plug-in agents) instead of being read from the directory.



NOTE: The OnePopPrefixIp scenario can be used in IPv4 or IPv6 dual-stack configuration. See [“SAE Support for Dual-Stack Configuration” on page 5](#) for more information.

The following agents collect information for resolvers:

- SAE plug-in agent `DelegatedIpVr` (using the `PA_DELEGATED_IPV6_PREFIX` attribute) collects information about the mappings of IP pools to VRs.
- SAE plug-in agent `FramedIpVr` (using the `PA_FRAMED_IPV6_PREFIX` attribute) collects information about the mappings of IP pools to VRs.
- SAE plug-in agent `InetAddressVr` (using the `PA_USER_INET_ADDRESS` and `PA_USER_IP_MASK` attributes) collects information about the mappings of IP pools to the VRs.

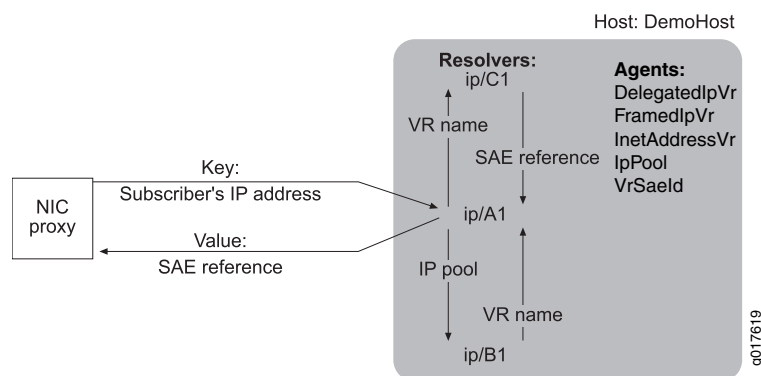
- Consolidator agent IpPool collects information from DelegatedIpVr, FramedIpVr, and InetAddressVr agents and publishes the list of known IP pools.
- Directory agent VrSaeld collects and publishes information about the mappings of VRs to the SAE.

In this configuration, the single host DemoHost supports all agents and resolvers. When the NIC proxy sends a subscriber's IP address to the host, the following sequence of events occurs:

1. The host passes the subscriber's IP address to resolver A1.
2. Resolver A1 returns the IP pool that best matches the IP address. This is synonymous with the longest or most specific match. For example, an IP address of 2001:db8:1:1:0:1:2:3 matches Pool1 (2001:db8:1:1::/64) and Pool2 (2001:db8:1:1::/32). In this case, the resolver returns Pool2 because it has the most specific match.
3. Resolver B1 obtains a VR name for the IP pool name and returns the VR name to resolver A1.
4. Resolver A1 forwards the VR name to resolver C1.
5. Resolver C1 obtains an SAE reference for the VR and returns the VR identity to resolver A1.
6. Resolver A1 passes the SAE reference to its host.
7. The host returns the SAE reference to the NIC proxy.

Figure 49 on page 240 shows the interactions of the NIC components for this realm.

Figure 49: OnePopPrefixIp Configuration



MultiPop Scenario

The MultiPop scenario illustrates a configuration that involves two POPs: Montreal and Ottawa. This configuration does not provide redundancy. The NIC proxy communicates with the back office host (BackOffice), which in turn communicates with the POP hosts (MontrealHost and OttawaHost). Hosts MontrealHost and OttawaHost support equivalent hosts and agents and manage resolutions in the same way.

When host BackOffice receives a data key from the NIC proxy, the following sequence of events occurs:

1. Host BackOffice forwards requests as follows:
 - If the request is for the Montreal POP, host BackOffice forwards the request to POP host MontrealHost.
 - If the request is for the Ottawa POP, host BackOffice forwards the request to POP host OttawaHost.
2. Delegating tasks to other resolvers as necessary, the resolvers in the POP obtain data values that correspond to the data key request, and return them.
3. The POP host returns the data values to host BackOffice, which returns the value to the NIC proxy.

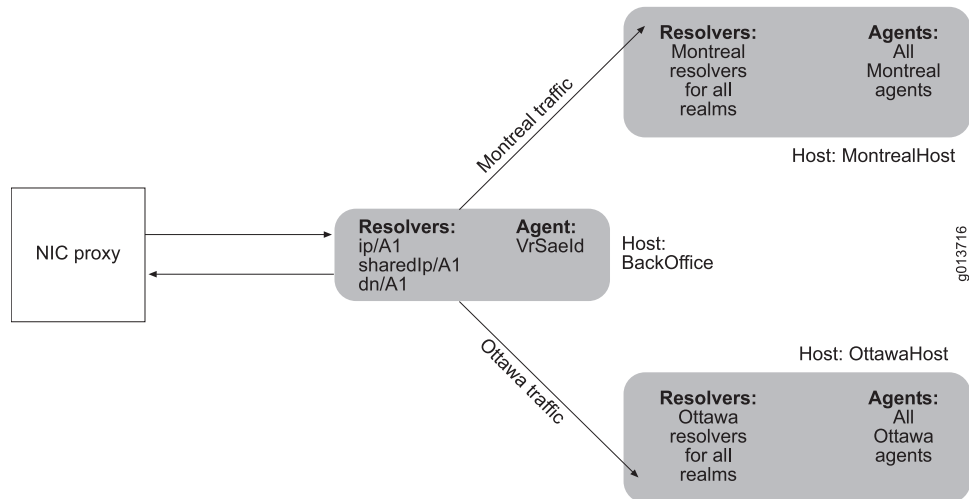
The scenario shows three realms for this configuration:

- IP
- Shared IP
- DN

Each realm provides a different type of resolution. The following sections provide information about these realms.

Figure 50 on page 241 illustrates this configuration.

Figure 50: MultiPop Configuration



IP Realm

This realm accommodates the situation in which IP address pools are configured locally on each VR. The resolution process takes a subscriber's IP address as the key and returns a reference to the SAE managing this subscriber as the value. This realm uses essentially

the same resolution process as the ip realm for the OnePop scenario (see [“OnePop Scenario” on page 210](#)). However, some of the constraints differ.

The following agents interact with the resolvers in this realm:

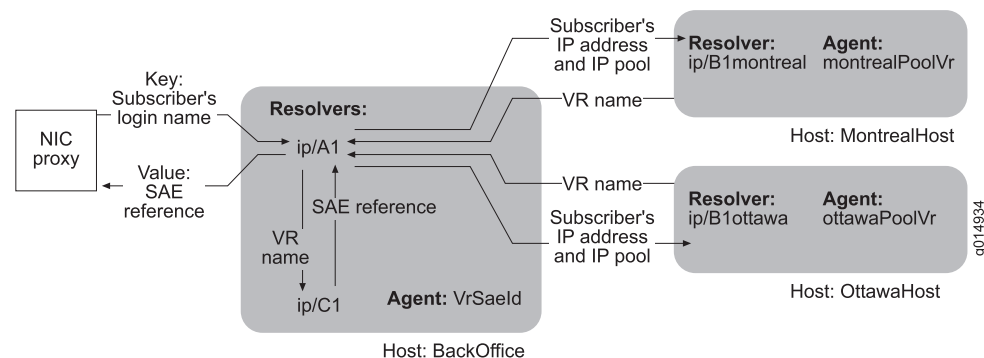
- Directory agents montrealPoolVr and ottawaPoolVr collect and publish information that maps IP address pools to VRs. Each agent publishes only the information that is relevant to its POP. You achieve selective publishing by relating an Ottawa scope to the VRs in the Ottawa POP and a Montreal scope to the VRs in the Montreal POP and defining a search filter for the agents to load only the VRs in its POP.
- Directory agent VrSaeld in the back office collects and publishes information that maps VRs to SAEs for both POPs.

When the NIC proxy sends a subscriber's IP address to host BackOffice, the following sequence of events occurs:

1. Host BackOffice passes the IP address to resolver ip/A1.
2. Resolver ip/A1 obtains an IP pool for the IP address.
3. Resolver ip/A1, based on the value of the IpPool, forwards the request to ip/B1montreal or ip/B1ottawa.
4. Resolver ip/B1montreal or resolver ip/B1ottawa obtains a VR name for this IP pool and returns the VR name to resolver ip/A1.
5. Resolver ip/A1 forwards the VR name to resolver ip/C1.
6. Resolver ip/C1 obtains the SAE identity for this VR and returns the value to resolver ip/A1.
7. Resolver ip/A1 returns the SAE reference to its host.
8. Host BackOffice returns the SAE reference to the NIC proxy.

[Figure 51 on page 242](#) illustrates the interactions of the NIC components for this realm.

Figure 51: iP Realm for MultiPop Configuration



Shared IP Realm

This realm accommodates the situation in which IP address pools are shared by VRs in the same POP. The realm takes a subscriber's IP address as the key and returns the

corresponding SAE as the value. To see the resolution graph for this realm, see [“OnePop Scenario” on page 210](#).

The following agents interact with resolvers in this realm:

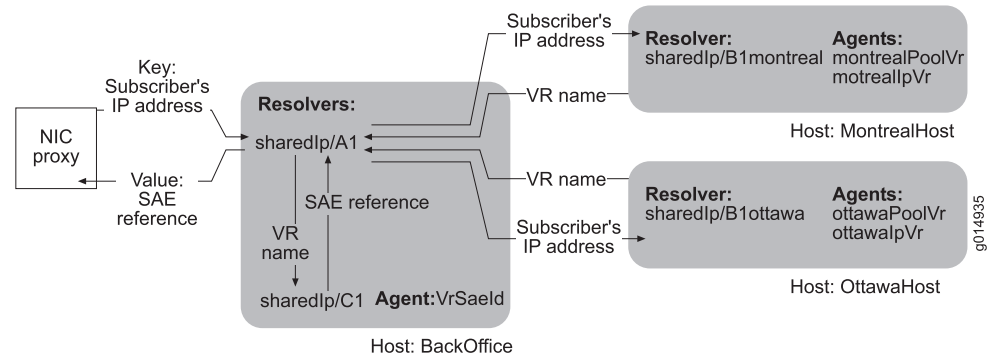
- Directory agents `montrealPoolVr` and `ottawaPoolVr` collect and publish information about the mappings of IP address pools to VRs. Each agent publishes only the information that is relevant to its POP.
- SAE plug-in agents `montrealIpVr` and `ottawaIpVr` collect and publish information about the mappings of subscriber IP addresses to VRs. Each agent publishes only the information that is relevant to its POP.
- Directory agent `VrSaeld` in the back office collects and publishes information about the mappings of VRs to SAEs for both POPs.

When the NIC proxy sends a subscriber's IP address to host BackOffice, the following sequence of events occurs:

1. Host BackOffice passes the IP address to resolver `sharedIp/A1`.
2. Resolver `sharedIp/A1` obtains an IP pool for the IP address.
3. Resolver `sharedIp/A1`, based on the value of the IP pool, forwards the request to `sharedIp/B1montreal` or `sharedIp/B1ottawa`.
4. Resolver `sharedIp/B1montreal` or resolver `sharedIp/B1ottawa` obtains a VR name for this IP address and returns the VR name to resolver `sharedIp/A1`.
5. Resolver `sharedIp/A1` forwards the VR name to resolver `sharedIp/C1`.
6. Resolver `sharedIp/C1` obtains the SAE identity for this VR and returns the value to resolver `sharedIp/A1`.
7. Resolver `sharedIp/A1` passes the SAE reference to its host.
8. Host BackOffice returns the SAE reference to the NIC proxy.

[Figure 52 on page 243](#) illustrates the interactions of the NIC components for this realm.

Figure 52: sharedIP Realm for MultiPop Configuration

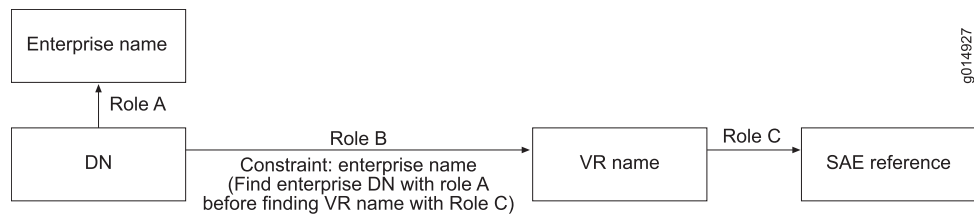


DN Realm

The DN realm takes the DN of an access subscriber (an access DN) as the key and returns the corresponding SAE as the value. [Figure 53 on page 244](#) shows the resolution process for this realm.

[Figure 53 on page 244](#) shows the resolution graph for this realm.

Figure 53: Resolution Graph for MultiPop dn Realm



The following agents interact with resolvers in this realm:

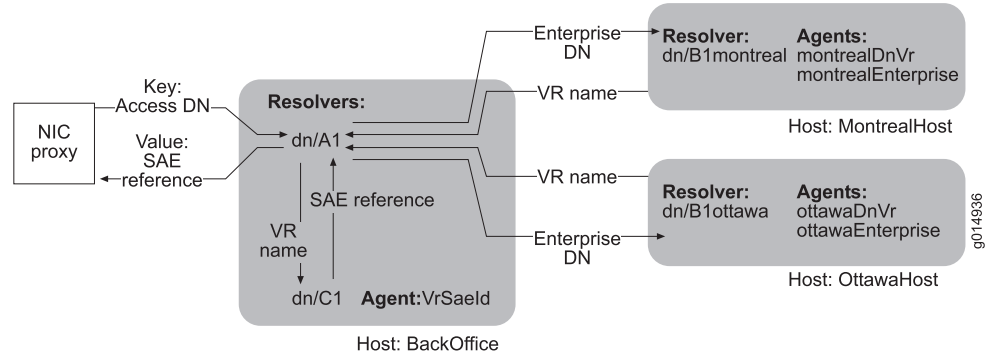
- Directory agents `ottawaEnterprise` and `montrealEnterprise` collect and publish information about the DNs of enterprise subscribers (enterprise DNs). Each agent publishes only the information that is relevant to its POP. You achieve selective publishing by relating an Ottawa service scope to the enterprises in the Ottawa POP and a Montreal service scope to the enterprises in the Montreal POP and defining a search filter for the agents to load only the enterprises in its POP.
- SAE plug-in agents `montrealDnVr` and `ottawaDnVr` collect and publish information about the mappings of access DNs to VRs. Each agent publishes only the information that is relevant to its POP.
- Directory agent `VrSaeld` collects and publishes information about the mappings of VRs to SAEs for both POPs.

When the NIC proxy sends an access DN to host BackOffice, the following sequence of events occurs:

1. Host BackOffice passes the access DN to resolver `dn/A1`.
2. Resolver `dn/A1` obtains an enterprise DN for the access DN.
3. Resolver `dn/A1`, based on the value of the enterprise DN, forwards the request to `dn/B1montreal` or `dn/B1ottawa`.
4. Resolver `dn/B1montreal` or resolver `dn/B1ottawa` obtains a VR name for this enterprise DN and returns the VR name to resolver `dn/A1`.
5. Resolver `dn/A1` forwards the VR name to resolver `dn/C1`.
6. Resolver `dn/C1` obtains the SAE reference for this VR and returns the value to resolver `dn/A1`.
7. Resolver `dn/A1` passes the SAE reference to its host.
8. Host BackOffice returns the SAE reference to the NIC proxy.

Figure 54 on page 245 illustrates the interactions of the NIC components for this realm.

Figure 54: dn Realm for MultiPop Configuration



- Related Documentation**
- [NIC Configuration Scenarios Overview on page 209](#)
 - [Configuring a NIC Scenario \(SRC CLI\) on page 156](#)

PART 5

Providing Admission Control with SRC ACP

- [Overview of Providing Admission Control with SRC ACP on page 249](#)
- [Configuring Admission Control \(SRC CLI\) on page 259](#)
- [Configuring Congestion Point Classification \(SRC CLI\) on page 297](#)
- [Managing SRC ACP \(SRC CLI\) on page 307](#)
- [Monitoring Admission Control \(SRC CLI\) on page 311](#)
- [Monitoring Admission Control \(C-Web Interface\) on page 329](#)

Overview of Providing Admission Control with SRC ACP

- [SRC ACP Overview on page 249](#)
- [Deriving Congestion Points Automatically on page 251](#)
- [Allocating Bandwidth to Applications Not Controlled by SRC ACP on page 254](#)
- [Use of Multiple SRC ACPs on page 254](#)
- [Interactions Between SRC ACP and Other Components on page 254](#)
- [Redundancy and State Synchronization on page 256](#)
- [Fault Recovery on page 257](#)
- [Creating an Application to Update Information for SRC ACP on page 258](#)

SRC ACP Overview

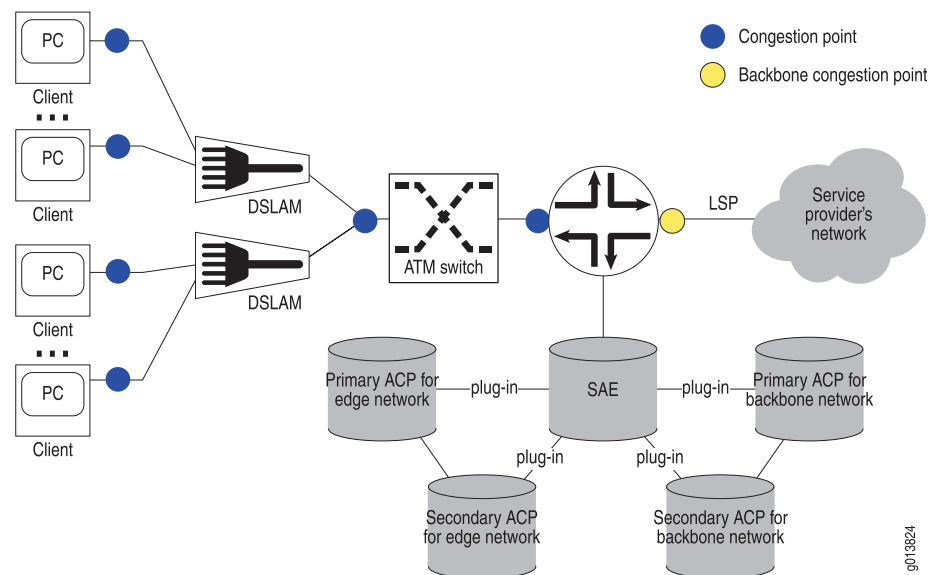
SRC ACP is an external plug-in for the SAE. SRC ACP authorizes and tracks subscribers' use of network resources associated with services that the SRC software manages. Service providers can implement SRC ACP configurations for both residential and enterprise subscribers. Consequently, both JunosE routers and devices running Junos OS are compatible with SRC ACP. References to virtual routers (VRs) in this documentation refer to an actual VR on a JunosE router or the single VR called default that the SRC software associates with each device running Junos OS.

SRC ACP operates in two separate regions of the SRC network: the *edge* network and the *backbone* network. The edge network is the layer 2 access network through which subscribers connect to the router. The backbone network is the region between the router and the service provider's network.

Congestion often occurs in the network at points where connections are aggregated. SRC ACP monitors congestion points at interfaces between devices in the edge network. In the backbone network, SRC ACP monitors one congestion point, a point-to-point label-switched path (LSP) between the router and the service provider's network.

[Figure 55 on page 250](#) shows a typical network topology.

Figure 55: Position of SRC ACP in Network



In the edge network, SRC ACP performs the following procedures to determine whether there are sufficient resources to activate a service:

- Tracks active services for each subscriber and the guaranteed traffic rate (bandwidth) at the congestion points associated with a subscriber.
- Tracks the rate of traffic between the subscriber and the network (upstream bandwidth) and the rate of traffic between the network and subscriber (downstream bandwidth).
- Monitors new requests for activation of services.
- Compares the resources required for the new services with the resources available for the subscriber and the congestion points.
- Activates the service if sufficient resources are available, and prevents activation of the service if sufficient resources are not available.

In the backbone network, SRC ACP performs the following procedures to determine whether there are sufficient resources to activate a service:

- Tracks the guaranteed traffic rate for a service at the congestion point.
- Tracks the actual traffic rate for the service at the congestion point.
- Monitors new requests for activation of services.
- Compares the resources required for the new services with the resources available at the congestion point.
- Activates the service if sufficient resources are available, and prevents activation of the service if sufficient resources are not available.

Typically, network administrators use their own network management applications and external applications to provide data for SRC ACP. SRC ACP first obtains updates from external applications through its remote CORBA interface, and then obtains updates

from the directory by means of LDAP. For information about developing external applications that send data to SRC ACP, see [“Creating an Application to Update Information for SRC ACP” on page 258](#). SRC ACP does not interact directly with the network to assess the capacity of a congestion point or actual use of network resources.

In the backbone network, SRC ACP can also execute applications defined in the action congestion point. Some applications require real-time congestion point status. If SRC ACP must provide real-time congestion point status to the application, state synchronization must be enabled to handle interface tracking events so that the congestion points are updated properly.



BEST PRACTICE: Be careful with service sessions that require asymmetrical bandwidth—for example, more downstream bandwidth than upstream bandwidth—especially when dealing with backbone congestion points. The upstream and downstream bandwidth required by a service session always consumes the available upstream and downstream bandwidth defined in the congestion point. However, what is upstream in one part of the network may be considered downstream in another part of the network.

When the SRC ACP's congestion point classification script maps from a session authorization start or stop event to a set of congestion points, it cannot indicate whether the upstream bandwidth required by the service session maps to the upstream or downstream available bandwidth defined in the congestion point. For asymmetrical service sessions, we recommend that you define action congestion points that contain the logic required to implement this mapping.

Related Documentation

- [Allocating Bandwidth to Applications Not Controlled by SRC ACP on page 254](#)
- [Use of Multiple SRC ACPs on page 254](#)
- [Interactions Between SRC ACP and Other Components on page 254](#)
- [Configuring SRC ACP \(SRC CLI\) on page 262](#)

Deriving Congestion Points Automatically

SRC ACP can derive some congestion points automatically. Depending on your network configuration and requirements, however, you may need to enter congestion points manually. This topic describes the conditions and requirements for SRC ACP to derive congestion points automatically.

Deriving Edge Congestion Points

For SRC ACP to derive edge congestion points, subscribers must always connect through the same interface on the router. In addition, SRC ACP requires one of the following conditions to derive edge congestion points if you are not using a congestion point profile:

- Access to subscriber profiles that define bandwidth values and a list of the distinguished names (DNs) of congestion points between the subscriber and the router.
- An ATM access network between the subscriber and the router for which all the traffic coming from one DSLAM travels on a single virtual path. In this case, SRC ACP automatically derives three congestion points through the network access server (NAS) port ID. [Table 15 on page 252](#) shows the edge congestion points and the corresponding locations in the directory.

For information about the NAS port ID, see *Using Flexible RADIUS Packet Definitions*.

SRC ACP does not use bandwidth statistics from subscriber profiles when it derives congestion points, because the congestion points already use that data.

Table 15: Congestion Points Derived Through NAS Port ID

Congestion Points	Location of Object in Directory
Physical interface on router	interfaceName=ATM<slot>/<port>, <i>orderedCimKeys</i> =<routerName>, <i>o</i> =AdmissionControl, <i>o</i> =umc <slot>—Number of port on router <port>—Number of port on router <routerName>—Hostname configured for router
ATM virtual path	interfaceName=ATM<slot>/<port>:<vpi> <i>orderedCimKeys</i> =<routerName>, <i>o</i> =AdmissionControl, <i>o</i> =umc <vpi>—Number of virtual path on router
ATM virtual connection	interfaceName=ATM<slot>/<port>:<vpi>.<vci> <i>orderedCimKeys</i> =<routerName>, <i>o</i> =AdmissionControl, <i>o</i> =umc <vci>—Number of virtual connection on router

Deriving Congestion Points from a Profile

If you configure a congestion point profile, SRC ACP can automatically derive congestion points for cases in which:

- There is no subscriber profile.
- The congestion points can be derived from information provided by the access interface on B-RAS. For example, in an ATM or VLAN connection, you can derive congestion points representing physical interfaces and intermediate switches based on the NAS port ID reported by B-RAS.

When SRC ACP receives notification to start subscriber tracking and to load congestion points for a subscriber, it runs a congestion point classification and accesses the configured congestion point profile. Congestion point classification uses the same classification engine as subscriber and interface classification in the SAE.

For this feature to operate correctly, you create a congestion point profile that automatically performs congestion point classification.

Deriving Backbone Congestion Points

SRC ACP can automatically derive backbone congestion points if you specify the setting `<-vrName->/<-serviceName->` for the congestion point associated with a service. When the SRC ACP starts operating, it will substitute the name of the VR and the service name from the activation request.

For example, you can specify the setting `<-vrName->/<-serviceName->` for the congestion point associated with a service called News. Then, when a subscriber who connects to the network through a VR called boston requests activation of this service, SRC ACP receives the request and proceeds as follows:

1. SRC ACP reads the congestion point specification, `<-vrName->/<-serviceName->`, from the congestion point defined for the service News.
2. SRC ACP substitutes the actual information, `boston/News`, in the variables.
3. SRC ACP uses this information to generate the DN `cn=News, cn=boston, o=CongestionPoints, o=umc`.
4. SRC ACP uses this DN to obtain from the directory the network interface, which defines the location of the congestion point, for this DN.

For this feature to operate correctly, you must configure the DN for each combination of VR and service to point to an actual network interface.

In cases where the combination of VR and service name do not uniquely identify the backbone congestion point, you can use backbone congestion point expressions and Python scripts to classify the backbone congestion point. Python scripts are executed when evaluating the congestion point expression. The format of the backbone congestion point expression is similar to the expression used in the congestion point profile. You can embed Python expressions, such as service plug-in attributes, in the congestion point expression. As a result, you can derive multiple backbone congestion points from a single service session.

For example, you can have a video-on-demand service that uses multiple video servers. One label-switched path (LSP) with the same parameters is created for each link between a video server and an access router. SRC ACP uses the network interface configuration information to generate the DN `interfaceName=<NetworkInterface>`, `orderedCimKeys=<NetworkDevice>`, `o=AdmissionControl`, `o=umc` as a template for the congestion point. When receiving a service request, the server activates the service for the subscriber on the appropriate congestion point. The backbone congestion point corresponds to the evaluation of the backbone congestion point expression.

Related Documentation

- [Defining a Congestion Point Profile \(SRC CLI\) on page 304](#)
- [Configuring SRC ACP \(SRC CLI\) on page 262](#)
- [Congestion Point Classification Overview on page 297](#)
- [Viewing Congestion Point Information by DN \(SRC CLI\) on page 326](#)

- [Viewing Congestion Point Information by Name \(SRC CLI\) on page 327](#)

[Allocating Bandwidth to Applications Not Controlled by SRC ACP](#)

If you control the bandwidth of some applications by means of SRC ACP, you can accommodate the applications that are not controlled by SRC ACP by assigning *background* bandwidths for the edge congestion points. The background bandwidth is the total bandwidth allocated to the applications for which bandwidth is not controlled by SRC ACP.

Because the total background bandwidth is unlikely to be used at a particular time, you can also specify a tuning factor that provides an estimation of the fraction of the background bandwidth that will be used. You can configure multiple values for the background bandwidth with corresponding tuning factors.

Related Documentation

- [Use of Multiple SRC ACPs on page 254](#)
- [Interactions Between SRC ACP and Other Components on page 254](#)
- [SRC ACP Overview on page 249](#)
- [Configuring SRC ACP to Manage the Edge Network \(SRC CLI\) on page 281](#)

[Use of Multiple SRC ACPs](#)

An SRC ACP can support one or more SAEs. Similarly, multiple SRC ACPs can support one SAE; for example, if an SAE is managing multiple VRs, you may have an SRC ACP for each VR. However, only one SRC ACP can manage a particular congestion point.

Related Documentation

- [SRC ACP Overview on page 249](#)
- [Allocating Bandwidth to Applications Not Controlled by SRC ACP on page 254](#)
- [Interactions Between SRC ACP and Other Components on page 254](#)
- [Configuring the SAE for SRC ACP \(SRC CLI\) on page 267](#)
- [Configuring SRC ACP \(SRC CLI\) on page 262](#)
- [Configuring SRC ACP Properties \(SRC CLI\) on page 269](#)

[Interactions Between SRC ACP and Other Components](#)

This topic describes how SRC ACP interacts with other components to track data.

1. (Edge and dual mode only) When a subscriber connects to the router, SRC ACP loads the subscriber profile from the directory. If the subscriber profile contains provisioned and actual traffic rates for the subscriber's interface and the set of congestion points between the subscriber and the router, SRC ACP caches the information while the subscriber is connected to the router. SRC ACP automatically updates the subscriber's actual upstream and downstream rates if the subscriber profile changes in the directory.
2. (Backbone mode only) When a subscriber activates a service, SRC ACP loads the network interfaces defined in the service and caches the information.
3. (Optional) SRC ACP obtains through its remote CORBA interface data from external applications about subscribers and congestion points. If a congestion point is unavailable, SRC ACP denies service activation requests on the associated network interface until the interface is available again.
4. If SRC ACP does not receive data from an external application, SRC ACP loads data about congestion points from the directory. For each congestion point the following data is retrieved:
 - Provisioned bandwidth
 - Background bandwidths (if used for edge congestion points)

SRC ACP caches this information and automatically updates the cache when the information changes in the directory.

5. (Edge and dual modes) If SRC ACP does not receive data from an external application, SRC ACP loads a subscriber's provisioned or actual bandwidth from the subscriber profile. If the actual bandwidth is available, SRC ACP ignores the provisioned bandwidth.

SRC ACP caches this information and automatically updates the cache when the information changes in the directory.

6. (Backbone and dual modes only) Using a hosted plug-in, the SAE monitors the states of router interfaces associated with backbone congestion points. The SAE sends relevant data to SRC ACP through the SRC ACP's remote interface.
7. When the subscriber requests activation of a service subscription (either through the SAE core API or automatically for activate-on-login services), the SAE notifies SRC ACP to authorize and track the service usage.
 - a. The SAE sends the requested bandwidth to SRC ACP.
 - b. SRC ACP authorizes or denies service activation.

If SRC ACP authorizes the service activation, the SAE activates the service and sends a tracking event to SRC ACP. SRC ACP updates the current bandwidth for all congestion points with the requested bandwidth.

If SRC ACP authorizes the service activation with state synchronization enabled, SRC ACP reserves the requested bandwidth on all congestion points until the reservation expires. You can specify the reservation timeout value when configuring SRC ACP operation.

- For each congestion point, SRC ACP verifies whether:

$$(\text{current bw} + \text{reserved bw} + \text{requested bw}) > [\text{provisioned bw} - (\text{background bw} \times \text{tuning factor})]$$

If the desired bandwidth exceeds the allocated bandwidth, SRC ACP denies service activation.

- When SRC ACP receives a service start tracking event, the requested bandwidth is committed. That is, for each congestion point, the requested bandwidth reservation is removed and the requested bandwidth is added to the current bandwidth.
- When the bandwidth reservation expires, the reserved bandwidth is released.

If SRC ACP does not authorize the service activation, the SAE delivers a message detailing the reason to the originator of the activation request.

SRC ACP distinguishes between bandwidth exceeded on the subscriber interface (first congestion point) and bandwidth exceeded on a network interface by sending two different messages back to the SAE. In the first case, the subscriber may resolve the bandwidth problem by deactivating another service.

8. When a service is deactivated (either through the SAE core API or because a session times out), SRC ACP updates the current bandwidth for all congestion points by removing the original requested bandwidth.
9. SRC ACP stores all information about subscribers, services, and congestion points in a set of files.

SRC ACP continually adds data to these files, but does not delete old data. Consequently, the sizes of the files continue to increase. SRC ACP does, however, reorganize the files when the sum of their sizes increments by a specified value. Reorganizing the files reduces their sizes. You can also reorganize the files by using the SRC CLI (see [“Reorganizing the File That Contains ACP Data” on page 308](#).)

Related Documentation

- [SRC ACP Overview on page 249](#)
- [Allocating Bandwidth to Applications Not Controlled by SRC ACP on page 254](#)
- [Use of Multiple SRC ACPs on page 254](#)
- [Configuration Statements for SRC ACP on page 259](#)
- [Configuring SRC ACP \(SRC CLI\) on page 262](#)
- [Configuring SRC ACP Properties \(SRC CLI\) on page 269](#)

Redundancy and State Synchronization

You can configure SRC ACP to synchronize states with the SAE.

State synchronization enables the current state to be transferred when SRC ACP has started up or lost its state. SRC ACP does not have to keep a local and persistent copy of the state. However, SRC ACP requires additional bandwidth to transfer state information that can affect performance.

You can configure SRC ACP redundancy for a region of the network by installing SRC ACP on two different hosts and connecting both SRC ACP hosts to the SAE (see [Figure 55 on page 250](#)). One SRC ACP acts as the primary application, and the other as the secondary application.



NOTE: Both SRC ACPs in a redundant pair must operate in the same mode. You cannot configure an SRC ACP in edge mode and an SRC ACP in backbone mode as a redundant pair.

To configure SRC ACP redundancy, enable redundancy. In this situation, the primary and secondary SRC ACPs are set up as a community and will communicate with each other to determine the primary SRC ACP. The primary SRC ACP registers its interoperable object reference (IOR) with the SAE so that the SAE will communicate only with the primary SRC ACP. When the primary SRC ACP becomes unavailable, the secondary SRC ACP assumes the role of the primary SRC ACP and performs state synchronization if necessary.

**Related
Documentation**

- [SRC ACP Overview on page 249](#)
- [Interactions Between SRC ACP and Other Components on page 254](#)
- [Configuration Statements for SRC ACP on page 259](#)
- [Configuring SRC ACP \(SRC CLI\) on page 262](#)
- [Configuring SRC ACP Properties \(SRC CLI\) on page 269](#)

Fault Recovery

If the SAE cannot reach SRC ACP, the SAE will deny all service activation requests. As soon as it reaches SRC ACP, the SAE again sends authorization requests to SRC ACP.

SRC ACP keeps the state of the congestion points in persistent storage, and if SRC ACP becomes unavailable, the service authorization can continue in the correct state. Because service activation requests are automatically denied when the SAE cannot reach SRC ACP, SRC ACP does not miss any active service sessions. The SAE will resend all service deactivation requests after SRC ACP is reachable again.

SRC ACP monitors SAE synchronization events for information about VR availability and SAE availability. If a VR reboots or an SAE becomes unavailable, SRC ACP updates the states of congestion points associated with those devices accordingly.

If the SAE becomes unavailable, the router will automatically reestablish connection to either the redundant SAE or, if a redundant SAE is not available, to the original SAE when it again becomes available. The new SAE notifies SRC ACP that the original SAE failed and specifies which subscriber and service sessions were logged during this time. SRC ACP uses this information to update its state.

**Related
Documentation**

- [SRC ACP Overview on page 249](#)
- [Interactions Between SRC ACP and Other Components on page 254](#)

- [Allocating Bandwidth to Applications Not Controlled by SRC ACP on page 254](#)
- [Use of Multiple SRC ACPs on page 254](#)

Creating an Application to Update Information for SRC ACP

You can develop your own application to update information about subscribers and congestion points for SRC ACP. The application can call one method to interact with SRC ACP. This method is called:

update (in RemoteUpdateType rut, in TagValueList attrs)

The method takes a property-value pair and passes the information to SRC ACP. For information about the properties and values you can pass to SRC ACP, see the file *acpPlugin.idl* in the folder *SDK/idl* in the **SDK+AppSupport+Demos+Samples.tar.gz** file on the Juniper Networks Web site at:

<https://www.juniper.net/support/products/src/index.html>.

To create an application that updates SRC ACP remotely:

1. Compile the IDL file, and generate the code in the language in which you want to write the application.
2. Write the application, and include the generated code for the IDL file.
3. Use the CORBA object reference defined in the property *ACP.syncRateAdaptor.ior* to send data from the application to SRC ACP.

For information about the interfaces, properties, and methods available in the CORBA remote API for ACP, see the documentation in the

SDK+AppSupport+Demos+Samples.tar.gz file on the Juniper Networks Web site at:

<https://www.juniper.net/support/products/src/index.html> . The files are in the *SDK/doc/idl/acp/html/index.html* directory.

Related Documentation

- [SRC ACP Overview on page 249](#)
- [Interactions Between SRC ACP and Other Components on page 254](#)
- [Allocating Bandwidth to Applications Not Controlled by SRC ACP on page 254](#)
- [Use of Multiple SRC ACPs on page 254](#)

Configuring Admission Control (SRC CLI)

- Configuration Statements for SRC ACP on page 259
- Configuring SRC ACP (SRC CLI) on page 262
- Creating Grouped Configurations for SRC ACP (SRC CLI) on page 262
- Configuring Local Properties for SRC ACP (SRC CLI) on page 263
- Configuring the SAE for SRC ACP (SRC CLI) on page 267
- Configuring SRC ACP Properties (SRC CLI) on page 269
- Configuring SRC ACP to Manage the Edge Network (SRC CLI) on page 281
- Configuring SRC ACP to Manage the Backbone Network (SRC CLI) on page 285
- Defining SRC ACP Congestion Point Usage Trap Thresholds (SRC CLI) on page 294

Configuration Statements for SRC ACP

Use the following configuration statements to configure SRC ACP at the **[edit]** hierarchy level:

```
shared acp configuration acp-options {  
    backup-directory backup-directory;  
    mode (edge | backbone | dual);  
    event-cache-size event-cache-size;  
    overload-method overload-method;  
    reservation-timeout reservation-timeout;  
    congestion-point-auto-completion;  
    tuning-factor tuning-factor;  
    subscriber-bandwidth-exceed-message subscriber-bandwidth-exceed-message;  
    network-bandwidth-exceed-message network-bandwidth-exceed-message;  
    backup-database-maximum-size backup-database-maximum-size;  
    remote-update-database-index-keys remote-update-database-index-keys;  
    interface-tracking-filter interface-tracking-filter;  
    state-sync-bulk-size state-sync-bulk-size;  
}  
shared acp configuration corba {  
    acp-ior acp-ior;  
    remote-update-ior remote-update-ior;  
}  
shared acp configuration ldap service-data {  
    edge-congestion-point-dn edge-congestion-point-dn;  
    backbone-congestion-point-dn backbone-congestion-point-dn;
```

```
    reload-congestion-points;
    congestion-points-eventing;
    server-address server-address;
    server-port server-port;
    dn dn;
    principal principal;
    password password;
    event-dn event-dn;
    directory-eventing;
    polling-interval polling-interval;
}
shared acp configuration ldap subscriber-data {
    congestion-points-eventing;
    server-address server-address;
    server-port server-port;
    dn dn;
    principal principal;
    password password;
    event-dn event-dn;
    directory-eventing;
    polling-interval polling-interval;
}
shared acp configuration logger name ...
shared acp configuration logger name file {
    filter filter;
    filename filename;
    rollover-filename rollover-filename;
    maximum-file-size maximum-file-size;
}
shared acp configuration logger name syslog {
    filter filter;
    host host;
    facility facility;
    format format;
}
shared acp configuration redundancy {
    enable-redundancy;
    local-ior local-ior;
    remote-ior remote-ior;
    ignore-user-tracking-out-of-sync;
    community-heartbeat community-heartbeat;
    community-acquire-timeout community-acquire-timeout;
    community-blackout-timeout community-blackout-timeout;
    redundant-naming-service redundant-naming-service;
}
shared acp configuration scripts-and-classification {
    script-factory-class script-factory-class;
    classification-factory-class classification-factory-class;
    classification-script classification-script;
    congestion-point-profile-script congestion-point-profile-script;
    extension-path extension-path;
}
shared acp configuration snmp congestion-point-usage-trap {
    selector [selector...];
    critical-threshold critical-threshold;
    major-threshold major-threshold;
```

```

    minor-threshold minor-threshold;
}
shared admission-control device name {
    description description;
}
shared admission-control device name interface name {
    description description;
    upstream-provisioned-rate upstream-provisioned-rate;
    downstream-provisioned-rate downstream-provisioned-rate;
    upstream-background-bandwidth upstream-background-bandwidth;
    downstream-background-bandwidth downstream-background-bandwidth;
    action-type (url | python | java-class | java-archive);
    action-class-name action-class-name;
    action-file-url action-file-url;
    action-parameters [action-parameters...];
    action-file-name action-file-name;
    detect-link-rate;
}
shared congestion-points profile name {
    interface [interface...];
}
slot number acp {
    java-runtime-environment java-runtime-environment;
    java-heap-size java-heap-size;
    java-garbage-collection-options java-garbage-collection-options;
    base-dn base-dn;
    snmp-agent;
    shared shared;
}
slot number acp initial {
    static-dn static-dn;
    dynamic-dn dynamic-dn;
}
slot number acp initial directory-connection {
    url url;
    backup-urls [backup-urls...];
    principal principal;
    credentials credentials;
    protocol (ldaps);
    timeout timeout;
    check-interval check-interval;
    blacklist;
    snmp-agent;
}
slot number acp initial directory-eventing {
    eventing;
    signature-dn signature-dn;
    polling-interval polling-interval;
    event-base-dn event-base-dn;
    dispatcher-pool-size dispatcher-pool-size;
}

```

Related Documentation

- For detailed information about each configuration statement, see the *SRC PE CLI Command Reference*.

- [Configuring SRC ACP \(SRC CLI\) on page 262](#)
- [Configuring the SAE for SRC ACP \(SRC CLI\) on page 267](#)
- [Configuring SRC ACP Properties \(SRC CLI\) on page 269](#)

Configuring SRC ACP (SRC CLI)

To use SRC ACP in an SRC network, perform the following configuration tasks:

1. (Optional) [“Creating Grouped Configurations for SRC ACP \(SRC CLI\)” on page 262](#)
2. [Configuring Local Properties for SRC ACP \(SRC CLI\) on page 263](#)
3. [Configuring the SAE for SRC ACP \(SRC CLI\) on page 267](#)
4. [Configuring SRC ACP Properties \(SRC CLI\) on page 269](#)
5. (Edge and dual mode only) [“Configuring SRC ACP to Manage the Edge Network \(SRC CLI\)” on page 281](#)
6. (Backbone and dual mode only) [“Configuring SRC ACP to Manage the Backbone Network \(SRC CLI\)” on page 285](#)
7. [Starting SRC ACP on page 307](#)

You can automate and scale the configuration of congestion points using congestion point classification. For more information, see [“Classifying Congestion Points \(SRC CLI\)” on page 298](#).

Related Documentation

- [Configuring SRC ACP \(C-Web Interface\)](#)
- [Viewing SNMP Information for SRC ACP \(SRC CLI\) on page 328](#)
- [Configuration Statements for SRC ACP on page 259](#)
- [SRC ACP Overview on page 249](#)

Creating Grouped Configurations for SRC ACP (SRC CLI)

We recommend that you configure SRC ACP within a group. When you create a configuration group, the software creates a configuration with default values filled in.

Configuration groups allow you to share the SRC ACP configuration with different SRC ACP instances in the SRC network. You can also set up different configurations for different instances.

You can then create a grouped SRC ACP configuration that is shared with some SRC ACP instances. For example, if you create two different SRC ACP groups called config1 and config2 within the shared SRC ACP configuration, you could select the SRC ACP configuration that should be associated with a particular SRC ACP instance.

Use the **shared** option of the **slot *number* acp** statement to select the group for an SRC ACP instance as part of the local configuration. Use the **shared acp group *name*** statements to configure the group.

To select and configure a group:

1. From configuration mode, select a group for an SRC ACP instance. For example, to select a group called `config1` in the path `/`:

```
[edit]
user@host# set slot 0 acp shared /config1
```

For more information, see “[Configuring Local Properties for SRC ACP \(SRC CLI\)](#)” on [page 263](#).

2. Commit the configuration.

```
[edit]
user@host# commit
commit complete.
```

3. From configuration mode, configure a group. For example, to configure a group called `config1`, specify the group as part of the SRC ACP configuration.

```
[edit]
user@host# edit shared acp group config1 ?
Possible completions:
  <[Enter]>          Execute this command
  > configuration
  > congestion-point-classifier
  > group             Group of ACP configuration properties
  |                  Pipe through a command
```

For more information, see “[Configuring SRC ACP \(SRC CLI\)](#)” on [page 262](#).

Related Documentation

- [Configuring the SAE for SRC ACP \(SRC CLI\) on page 267](#)
- [Configuring SRC ACP Properties \(SRC CLI\) on page 269](#)
- [Creating an Application to Update Information for SRC ACP on page 258](#)
- [Configuration Statements for SRC ACP on page 259](#)
- [Interactions Between SRC ACP and Other Components on page 254](#)

Configuring Local Properties for SRC ACP (SRC CLI)

Configure initial properties, including Java heap memory, including directory connection and directory eventing properties.

Tasks to configure the local properties for SRC ACP are:

- [Configuring Basic Local Properties for SRC ACP on page 264](#)
- [Configuring Initial Properties for SRC ACP on page 265](#)

- [Configuring Directory Connection Properties for SRC ACP on page 265](#)
- [Configuring Initial Directory Eventing Properties for SRC ACP on page 266](#)

Configuring Basic Local Properties for SRC ACP

Use the following configuration statements to configure basic local properties for SRC ACP:

```
slot number acp {  
  java-runtime-environment java-runtime-environment;  
  java-heap-size java-heap-size;  
  java-garbage-collection-options java-garbage-collection-options;  
  base-dn base-dn;  
  snmp-agent;  
  shared shared;  
}
```

To configure basic local properties:

1. From configuration mode, access the configuration statement that configures the local properties.

```
user@host# edit slot 0 acp
```

2. Specify the basic local properties for ACP.

```
[edit slot 0 acp]  
user@host# set ?
```

For more information about configuring local properties for the SRC components, see *Configuring Basic Local Properties*.

3. Configure the garbage collection functionality of the Java Virtual Machine.

```
[edit slot 0 acp]  
user@host# set java-garbage-collection-options java-garbage-collection-options
```

4. Select an SRC ACP group configuration.

```
[edit slot 0 acp]  
user@host# set shared shared
```

For more information, see [“Creating Grouped Configurations for SRC ACP \(SRC CLI\)” on page 262](#).

5. (Optional) Verify your configuration.

```
[edit slot 0 acp]  
user@host# show  
shared /config;  
initial {  
  directory-connection {  
    url ldap://127.0.0.1:389/;  
    principal cn=conf,o=Operators,<base>;  
    credentials *****;  
  }  
  directory-eventing {
```



```

        eventing;
        polling-interval 30;
    }
}

```

Configuring Initial Properties for SRC ACP

Use the following configuration statements to configure initial properties for SRC ACP:

```

slot number acp initial {
    static-dn static-dn;
    dynamic-dn dynamic-dn;
}

```

To configure initial local properties:

1. From configuration mode, access the configuration statement that configures the initial properties.

```
user@host# edit slot 0 acp initial
```

2. Specify the properties for SRC ACP.

```

[edit slot 0 acp initial]
user@host# set ?

```

For more information about configuring local properties for the SRC components, see *Configuring Basic Local Properties*.

3. (Optional) Verify your configuration.

```

[edit slot 0 acp initial]
user@host# show

```

Configuring Directory Connection Properties for SRC ACP

Use the following configuration statements to configure directory connection properties for SRC ACP:

```

slot number acp initial directory-connection {
    url url;
    backup-urls [backup-urls...];
    principal principal;
    credentials credentials;
    protocol (ldaps);
    timeout timeout;
    check-interval check-interval;
    blacklist;
    snmp-agent;
}

```

To configure directory connection properties:

1. From configuration mode, access the configuration statement that configures the directory connection properties.

```
user@host# edit slot 0 acp initial directory-connection
```

2. Specify the properties for ACP.

```
[edit slot 0 acp initial directory-connection]
user@host# set ?
```

For more information about configuring local properties for the SRC components, see *Configuring Basic Local Properties*.

3. (Optional) Verify your configuration.

```
[edit slot 0 acp initial directory-connection]
user@host# show
url ldap://127.0.0.1:389/;
principal cn=conf,o=Operators,<base>;
credentials *****;
```

Configuring Initial Directory Eventing Properties for SRC ACP

Use the following configuration statements to configure directory eventing properties for SRC ACP:

```
slot number acp initial directory-eventing {
  eventing;
  signature-dn signature-dn;
  polling-interval polling-interval;
  event-base-dn event-base-dn;
  dispatcher-pool-size dispatcher-pool-size;
}
```

To configure initial directory eventing properties:

1. From configuration mode, access the configuration statement that configures the local properties.

```
user@host# edit slot 0 acp initial eventing
```

2. Specify the initial directory eventing properties for SRC ACP.

```
[edit slot 0 acp initial directory-eventing]
user@host# set ?
```

For more information about configuring local properties for the SRC components, see *Configuring Basic Local Properties*.

3. (Optional) Verify your configuration.

```
[edit slot 0 acp initial directory-eventing]
user@host# show
eventing;
polling-interval 30;
```

Related Documentation

- [Configuring SRC ACP \(SRC CLI\) on page 262](#)
- [Creating Grouped Configurations for SRC ACP \(SRC CLI\) on page 262](#)
- [Configuring Local Properties for SRC ACP \(C-Web Interface\)](#)
- [Configuring SRC ACP Properties \(SRC CLI\) on page 269](#)

Configuring the SAE for SRC ACP (SRC CLI)

You must configure the SAE to recognize SRC ACP by adding information about SRC ACP to the SAE properties. The tasks for configuring the SAE for SRC ACP are:

- [Configuring SRC ACP as an External Plug-In on page 267](#)
- [Configuring Event Publishers on page 267](#)
- [Configuring the SAE to Monitor Interfaces for Congestion Points on page 267](#)

Configuring SRC ACP as an External Plug-In

To configure an external plug-in for the SAE:

1. From configuration mode, access the configuration statement that configures the external plug-ins.

```
user@host# edit shared sae configuration plug-ins name name external
```

2. Specify the plug-in attributes.

```
[edit shared sae configuration plug-ins name name external]
user@host# set attributes ?
```

For edge and dual modes—upstream-bandwidth, downstream-bandwidth, service-name, router-name, login-name, user-dn, port-id, session-id, user-ip-address, nas-ip, user-session-id, event-time

For backbone mode—upstream-bandwidth, downstream-bandwidth, service-name, router-name, session-id, nas-ip, event-time

For more information about configuring plug-in attributes, see *Configuring the SAE for External Plug-Ins (SRC CLI)*.

Configuring Event Publishers

You must configure the SAE to publish the following types of events to SRC ACP:

- (Edge and dual mode only) Global subscriber tracking
- Global service authorization
- Global service tracking

For information about configuring event publishers, see *Special Types of Event Publishers*. Identify the instance of SRC ACP by the name of the host on which you configured it.

Configuring the SAE to Monitor Interfaces for Congestion Points



NOTE: Configure this feature only if SRC ACP is in backbone or dual mode.

The SAE uses a hosted internal plug-in to monitor the state of interfaces on a VR for backbone congestion points. If a subscriber tries to activate a service on an interface that is unavailable, the SAE denies the request. The plug-in also monitors the directory for new backbone congestion points.

When this plug-in initializes, it reads all the backbone services from the directory and generates a list of the DNS (network interfaces) of the backbone congestion points. The SAE sends interface tracking events, which contain the names of the interfaces, VRs, and routers to this plug-in. For this feature to work correctly, the interface, VR, and router must be configured (see [“Configuring Network Interfaces in the Directory for the Backbone Network” on page 285](#)).

To configure the ACP interface listener as an internal plug-in for the SAE:

1. From configuration mode, access the configuration statement that configures the ACP interface listener.

```
user@host# edit shared sae configuration plug-ins name name acp-interface-listener
```

2. Specify the IP address or name of the host that supports the directory that contains backbone service definitions and network interfaces.

```
[edit shared sae configuration plug-ins name name acp-interface-listener]  
user@host# set ldap-server ldap-server
```

3. Specify the DN of the directory entry that defines the username with which the plug-in accesses the directory.

```
[edit shared sae configuration plug-ins name name acp-interface-listener]  
user@host# set bind-dn bind-dn
```

4. Specify the password with which the plug-in accesses the directory.

```
[edit shared sae configuration plug-ins name name acp-interface-listener]  
user@host# set bind-password bind-password
```

5. Specify whether the connection to the directory uses secure LDAP. If you do not configure a security protocol, plain socket is used.

```
[edit shared sae configuration plug-ins name name acp-interface-listener]  
user@host# set ldaps
```

6. Specify the DN at which SRC ACP stores backbone congestion points.

```
[edit shared sae configuration plug-ins name name acp-interface-listener]  
user@host# set congestion-points-base-dn congestion-points-base-dn
```

7. Specify the DN at which SRC ACP stores edge congestion points.

```
[edit shared sae configuration plug-ins name name acp-interface-listener]  
user@host# set admission-control-base-dn admission-control-base-dn
```

8. (Optional) Specify the maximum time that the plug-in waits for the router to respond.

```
[edit shared sae configuration plug-ins name name acp-interface-listener]
user@host# set timeout timeout
```

9. Specify the object reference for the ACP plug-in, as defined by the object reference for SRC ACP (see information about the **acp-ior** option in [“Configuring SRC ACP Properties \(SRC CLI\)” on page 269](#)).

```
[edit shared sae configuration plug-ins name name acp-interface-listener]
user@host# set acp-remote-corba-ior acp-remote-corba-ior
```

10. (Optional) Verify your configuration.

```
[edit shared sae configuration plug-ins name name acp-interface-listener]
user@host# show
```

Related Documentation

- [Configuring the SAE for SRC ACP \(C-Web Interface\)](#)
- [Configuring SRC ACP \(SRC CLI\) on page 262](#)
- [Configuring SRC ACP to Manage the Edge Network \(SRC CLI\) on page 281](#)
- [Configuring SRC ACP to Manage the Backbone Network \(SRC CLI\) on page 285](#)

Configuring SRC ACP Properties (SRC CLI)

To configure SRC ACP properties, perform these tasks:

1. [Configuring Logging Destinations for SRC ACP on page 269](#)
2. [Configuring SRC ACP Operation on page 271](#)
3. [Configuring CORBA Interfaces on page 274](#)
4. [Configuring SRC ACP Redundancy on page 275](#)
5. [Configuring Connections to the Subscribers' Directory on page 277](#)
6. [Configuring Connections to the Services' Directory on page 278](#)
7. [Configuring SRC ACP Scripts and Classification on page 280](#)

Configuring Logging Destinations for SRC ACP

You can store log messages in a file or in the system logging facility. To format log messages in an easy to understand audit pattern, define the logger name as “audit”. The events captured in an audit logger include:

- ACP's calculation result of edge congestion points for a user session.
- ACP's calculation result of backbone congestion points for a service authorization request.
- ACP's decision (grant or deny) on a service authorization event. If denied, the congestion point that is over the limit is logged.

- Bandwidth usage update to congestion points following a service start, interim (only when it is different from start event), or stop tracking event.
- Bandwidth usage update to congestion points following an interface tracking event.

Use the following configuration statements to configure logging destinations for SRC ACP:

```
shared acp configuration logger name ...
shared acp configuration logger name file {
    filter filter;
    filename filename;
    rollover-filename rollover-filename;
    maximum-file-size maximum-file-size;
}
shared acp configuration logger name syslog {
    filter filter;
    host host;
    facility facility;
    format format;
}
```

Configuring Logging Destinations to Store Messages in a File

To configure logging destinations to store log messages in a file:

1. From configuration mode, access the configuration statement that configures the name and type of logging destination. In this sample procedure, the logging destination called file-1 is configured in the config group.

```
user@host# edit shared acp group config configuration logger file-1 file
```

2. Specify the properties for the logging destination.

```
[edit shared acp group config configuration logger file-1 file]
user@host# set ?
```

For more information about configuring properties for the logging destination, see *Configuring an SRC Component to Store Log Messages in a File (SRC CLI)*.

3. (Optional) Verify your configuration.

```
[edit shared acp group config configuration logger file-1 file]
user@host# show
filename var/log/acp_debug.log;
rollover-filename var/log/acp_debug.alt;
```

Configuring Logging Destinations to Send Messages to System Logging Facility

To configure logging destinations to send log messages to the system logging facility:

1. From configuration mode, access the configuration statement that configures the name and type of logging destination. In this sample procedure, the logging destination called syslog-1 is configured in the config group.

```
user@host# edit shared acp group config configuration logger syslog-1 syslog
```

2. Specify the properties for the logging destination.

```
[edit shared acp group config configuration logger syslog-1 syslog]
```

```
user@host# set ?
```

For more information about configuring properties for the logging destination, see *Configuring System Logging (SRC CLI)*.

3. (Optional) Verify your configuration.

```
[edit shared acp group config configuration logger syslog-1 syslog]
user@host# show
filter /error-;
host loghost;
```

Configuring SRC ACP Operation

Use the following configuration statements to configure how SRC ACP operates:

```
shared acp configuration acp-options {
  backup-directory backup-directory;
  mode (edge | backbone | dual);
  event-cache-size event-cache-size;
  overload-method overload-method;
  reservation-timeout reservation-timeout;
  congestion-point-auto-completion;
  tuning-factor tuning-factor;
  subscriber-bandwidth-exceed-message subscriber-bandwidth-exceed-message;
  network-bandwidth-exceed-message network-bandwidth-exceed-message;
  backup-database-maximum-size backup-database-maximum-size;
  remote-update-database-index-keys remote-update-database-index-keys;
  interface-tracking-filter interface-tracking-filter;
  state-sync-bulk-size state-sync-bulk-size;
}
```

To configure SRC ACP operation:

1. From configuration mode, access the configuration statement that configures SRC ACP operation. In this sample procedure, the SRC ACP operating properties are configured in the config group.

```
user@host# edit shared acp group config configuration acp-options
```

2. Specify the folder that stores backup information about subscribers, services, and congestion points.

```
[edit shared acp group config configuration acp-options]
user@host# set backup-directory
```

3. Specify the regions of the network that SRC ACP manages.

```
[edit shared acp group config configuration acp-options]
user@host# set mode (edge | backbone | dual)
```

4. Specify the number of plug-in events from the SAE that SRC ACP can store in its cache.

```
[edit shared acp group config configuration acp-options]
user@host# set event-cache-size event-cache-size
```

5. Specify how SRC ACP deals with situations in which the components exceed the allocated bandwidth because the service was activated after the authorization was granted.

```
[edit shared acp group config configuration acp-options]  
user@host# set overload-method overload-method
```

If you specify -1, SRC ACP ignores overload. An integer greater than or equal to 0 specifies the bandwidth (in bits per second) by which the maximum may be exceeded.

6. Specify the time to wait before a bandwidth reservation expires. The reserved bandwidth is reclaimed by SRC ACP when the reservation expires.

```
[edit shared acp group config configuration acp-options]  
user@host# set reservation-timeout reservation-timeout
```

7. Specify whether SRC ACP uses the information acquired from the router to determine the congestion points.

```
[edit shared acp group config configuration acp-options]  
user@host# set congestion-point-auto-completion
```

8. Specify the factors that compensate for actual use of bandwidth, as opposed to allocated bandwidth.

```
[edit shared acp group config configuration acp-options]  
user@host# set tuning-factor tuning-factor
```

9. Specify the error message that SRC ACP sends when the subscriber exceeds the allocated bandwidth.

```
[edit shared acp group config configuration acp-options]  
user@host# set subscriber-bandwidth-exceed-message  
subscriber-bandwidth-exceed-message
```

10. Specify the error message that SRC ACP sends when traffic flow exceeds the allocated bandwidth on an interface between the subscriber and the router.

```
[edit shared acp group config configuration acp-options]  
user@host# set network-bandwidth-exceed-message  
network-bandwidth-exceed-message
```

11. Specify the value by which the sum of the sizes of the files that contain SRC ACP data can increment before SRC ACP reorganizes the files.

```
[edit shared acp group config configuration acp-options]  
user@host# set backup-database-maximum-size backup-database-maximum-size
```

Choose a value that is significantly lower than the capacity of the machine's hard disk.

12. Specify the values to look for in the configuration data. Specifying index keys can improve performance by filtering the data.

```
[edit shared acp group config configuration acp-options]
```



```
user@host# set remote-update-database-index-keys
remote-update-database-index-keys
```

The value is a list of attributes, separated by commas. An attribute is one of the following text strings:

- accountingId—Value of directory attribute accountingUserId.
- dhcpPacket—Content of the DHCP discover request.
- hostname— Name of the host on which the SAE is installed.
- ifIndex—SNMP index of the interface. This attribute is not supported on devices running Junos OS.
- ifRadiusClass—RADIUS class attribute on the JunosE interface. This attribute is not supported on devices running Junos OS.
- ifSessionId—Identifier for RADIUS accounting on the JunosE interface. This attribute is not supported on devices running Junos OS.
- interfaceAlias—Alias of the interface; that is, the IP description in the interface configuration.
- interfaceDescr—SNMP description of the interface.
- interfaceName—Name of the interface.
- loginName—Subscriber's login name.
- nasInetAddress—IP address of the router; using a byte array instead of an integer.
- nasPort—NAS port used by the router to identify the interface to RADIUS.
- portId—Identifier of VLAN or virtual circuit. For a virtual circuit, use the format <VPI>/<VCI>. This attribute is not supported on devices running Junos OS.
 - <VPI>—Virtual path identifier
 - <VCI>—Virtual connection identifier
- primaryUserName—PPP login name or the public DHCP username. This attribute is not supported on devices running Junos OS.
- routerName—Name of the virtual router in the format <virtualRouter>@<router>.
 - <virtualRouter>—Virtual router name
 - <router>—Router name
- routerType—Type of router driver.
- userInetAddress—IP address of the subscriber that uses a byte array instead of an integer.
- userMacAddress—MAC address of the DHCP subscriber. This attribute is not supported on devices running Junos OS.

- **userRadiusClass**—RADIUS class attribute of the subscriber session for a service. This attribute can occur multiple times and can be returned by an authorization plug-in.
- **userType**—Type of subscriber.

13. Specify the interface tracking event to be ignored by SRC ACP.

```
[edit shared acp group config configuration acp-options]  
user@host# set interface-tracking-filter interface-tracking-filter
```

The value is filter strings in the format of a list of <attribute>=<value> pairs. The filter strings can be contained within query operations.

- **<attribute>**—Name of an attribute for an interface tracking event. See value for the **remote-update-database-index-keys** option described *Configuring SRC ACP Properties (SRC CLI)*.
- **<value>**—Filtering string of the following types:
 - *****—Any value
 - **Explicit string**—Any value matching the specified string (not case-sensitive)
 - **String containing an asterisk**—Any value containing the specified string (not case-sensitive)
- To perform query operations on filter strings, you can use the following values in your filter strings:
 - **()**—Match no objects.
 - **(*)**—Match all objects.
 - **(&<filter><filter>...)**—Performs logical AND operation on filter strings; true if all filter strings match.
 - **(|<filter><filter>...)**—Performs logical OR operation on filter strings; true if at least one filter string matches.
 - **(!<filter>)**—Performs logical NOT operation on filter string; true if the filter string does not match.

14. (Optional) Specify the number of events the SAE sends to SRC ACP in a single method call during state synchronization.

```
[edit shared acp group config configuration acp-options]  
user@host# set state-sync-bulk-size state-sync-bulk-size
```

15. (Optional) Verify your configuration.

```
[edit shared acp group config configuration acp-options]  
user@host# show
```

Configuring CORBA Interfaces

Use the following configuration statements to configure CORBA interfaces for SRC ACP:

```
shared acp configuration corba {
  acp-ior acp-ior;
  remote-update-ior remote-update-ior;
}
```

To configure CORBA interfaces:

1. From configuration mode, access the configuration statement that configures CORBA interfaces for SRC ACP. In this sample procedure, the CORBA interfaces are configured in the config group.

```
user@host# edit shared acp group config configuration corba
```

2. Export the object reference for SRC ACP through either a local file or a Common Object Services (COS) naming service.

```
[edit shared acp group config configuration corba]
user@host# set acp-ior acp-ior
```

3. Specify the object reference for the ACP external interface.

```
[edit shared acp group config configuration corba]
user@host# set remote-update-ior remote-update-ior
```

4. (Optional) Verify your configuration.

```
[edit shared acp group config configuration corba]
user@host# show
  acp-ior file:///var/acp/acp.ior;
  remote-update-ior file:///var/acp/sra.ior;
```

Configuring SRC ACP Redundancy

Use the following configuration statements to configure SRC ACP redundancy and state synchronization with the SAE:

```
shared acp configuration redundancy {
  enable-redundancy;
  local-ior local-ior;
  remote-ior remote-ior;
  ignore-user-tracking-out-of-sync;
  community-heartbeat community-heartbeat;
  community-acquire-timeout community-acquire-timeout;
  community-blackout-timeout community-blackout-timeout;
  redundant-naming-service redundant-naming-service;
}
```

To configure SRC ACP redundancy and state synchronization with the SAE:

1. From configuration mode, access the configuration statement that configures SRC ACP redundancy. In this sample procedure, the properties are configured in the config group.

```
user@host# edit shared acp group config configuration redundancy
```

2. (Optional) Enable SRC ACP redundancy.

```
[edit shared acp group config configuration redundancy]
user@host# set enable-redundancy
```

3. Export the object reference for this SRC ACP (local interface) through a Common Object Services (COS) naming service in a redundant SRC ACP configuration.

```
[edit shared acp group config configuration redundancy]
user@host# set local-ior local-ior
```

4. Resolves the object reference for the other SRC ACP (remote interface) through a Common Object Services (COS) naming service in a redundant SRC ACP configuration. For redundancy, the remote IOR value of one SRC ACP must match the local IOR value of the other SRC ACP.

```
[edit shared acp group config configuration redundancy]
user@host# set remote-ior remote-ior
```

5. (Optional) Specify whether user-tracking events should be ignored when they raise an OutOfSync exception to the SAE when state synchronization is enabled. SRC ACP raises an OutOfSync exception when SRC ACP handles service tracking or authentication events without receiving a user start event first.

```
[edit shared acp group config configuration redundancy]
user@host# set ignore-user-tracking-out-of-sync
```

6. (Optional) Specify the time interval for community members to check each other's availability when both redundancy and state synchronization are enabled.

```
[edit shared acp group config configuration redundancy]
user@host# set community-heartbeat community-heartbeat
```

7. (Optional) Specify the time to wait before trying to reacquire the distributed lock when both redundancy and state synchronization are enabled.

```
[edit shared acp group config configuration redundancy]
user@host# set community-acquire-timeout community-acquire-timeout
```

8. (Optional) Specify the time to wait before regaining control when both redundancy and state synchronization are enabled.

```
[edit shared acp group config configuration redundancy]
user@host# set community-blackout-timeout community-blackout-timeout
```

9. Export the object reference for the backup naming service through a local file or COS naming service in a redundant SRC ACP configuration. The primary SRC ACP registers the IOR and redundancy IOR to both naming services, while the secondary SRC ACP registers the redundancy IOR to both naming services.

```
[edit shared acp group config configuration redundancy]
user@host# set redundant-naming-service redundant-naming-service
```

10. (Optional) Verify your configuration.

```
[edit shared acp group config configuration redundancy]
user@host# show
```

Configuring Connections to the Subscribers' Directory

Use the following configuration statements to configure how SRC ACP connects to the directory that contains subscriber information:

```
shared acp configuration ldap subscriber-data {
  congestion-points-eventing;
  server-address server-address;
  server-port server-port;
  dn dn;
  principal principal;
  password password;
  event-dn event-dn;
  directory-eventing;
  polling-interval polling-interval;
}
```

To configure connections to the directory that stores subscriber information:

1. From configuration mode, access the configuration statement that configures SRC ACP connections to the subscribers' directory. In this sample procedure, the connections are configured in the config group.

```
user@host# edit shared acp group config configuration ldap subscriber-data
```

2. (Optional) Enable directory eventing for congestion points.

```
[edit shared acp group config configuration ldap subscriber-data]
user@host# set congestion-points-eventing
```

3. Specify the list of primary and redundant servers that manage data for subscribers.

```
[edit shared acp group config configuration ldap subscriber-data]
user@host# set server-address server-address
```

4. Specify the TCP port for the directory.

```
[edit shared acp group config configuration ldap subscriber-data]
user@host# set server-port server-port
```

5. Specify the DN of the root of the directory.

```
[edit shared acp group config configuration ldap subscriber-data]
user@host# set dn dn
```

6. Specify the DN used to authorize connections to the directory.

```
[edit shared acp group config configuration ldap subscriber-data]
user@host# set principal principal
```

7. Specify the password used to authorize connections to the directory.

```
[edit shared acp group config configuration ldap subscriber-data]
```

```
user@host# set password password
```

8. Specify the DN of the directory that contains event information.

```
[edit shared acp group config configuration ldap subscriber-data]  
user@host# set event-dn event-dn
```

9. (Optional) Enable directory eventing.

```
[edit shared acp group config configuration ldap subscriber-data]  
user@host# set directory-eventing
```

10. Specify the time interval at which the SRC component polls the directory.

```
[edit shared acp group config configuration ldap subscriber-data]  
user@host# set polling-interval polling-interval
```

11. (Optional) Verify your configuration.

```
[edit shared acp group config configuration ldap subscriber-data]  
user@host# show
```

Configuring Connections to the Services' Directory

Use the following configuration statements to configure how SRC ACP connects to the directory that contains information about services:

```
shared acp configuration ldap service-data {  
  edge-congestion-point-dn edge-congestion-point-dn;  
  backbone-congestion-point-dn backbone-congestion-point-dn;  
  reload-congestion-points;  
  congestion-points-eventing;  
  server-address server-address;  
  server-port server-port;  
  dn dn;  
  principal principal;  
  password password;  
  event-dn event-dn;  
  directory-eventing;  
  polling-interval polling-interval;  
}
```

To configure connections to the directory that stores service information:

1. From configuration mode, access the configuration statement that configures SRC ACP connections to the services' directory. In this sample procedure, the connections are configured in the config group.

```
user@host# edit shared acp group config configuration ldap service-data
```

2. Specify the DN of the directory that contains information about network interfaces for edge congestion points.

```
[edit shared acp group config configuration ldap service-data]  
user@host# set edge-congestion-point-dn edge-congestion-point-dn
```

3. Specify the DN of the directory that contains information about network interfaces for backbone congestion point objects.

```
[edit shared acp group config configuration ldap service-data]
user@host# set backbone-congestion-point-dn backbone-congestion-point-dn
```

4. (Optional) Specify whether SRC ACP detects changes in the backbone congestion point for a service while SRC ACP is operative.

```
[edit shared acp group config configuration ldap service-data]
user@host# set reload-congestion-points
```

Set this value only when you want to modify a congestion point.

5. (Optional) Enable directory eventing for congestion points.

```
[edit shared acp group config configuration ldap service-data]
user@host# set congestion-points-eventing
```

6. Specify the list of primary and redundant servers that manage data for subscribers.

```
[edit shared acp group config configuration ldap service-data]
user@host# set server-address server-address
```

7. Specify the TCP port for the directory.

```
[edit shared acp group config configuration ldap service-data]
user@host# set server-port server-port
```

8. Specify the DN of the root of the directory.

```
[edit shared acp group config configuration ldap service-data]
user@host# set dn dn
```

9. Specify the DN used to authorize connections to the directory.

```
[edit shared acp group config configuration ldap service-data]
user@host# set principal principal
```

10. Specify the password used to authorize connections to the directory.

```
[edit shared acp group config configuration ldap service-data]
user@host# set password password
```

11. Specify the DN of the directory that contains event information.

```
[edit shared acp group config configuration ldap service-data]
user@host# set event-dn event-dn
```

12. (Optional) Enable directory eventing.

```
[edit shared acp group config configuration ldap service-data]
user@host# set directory-eventing
```

13. Specify the time interval at which the SRC component polls the directory.

```
[edit shared acp group config configuration ldap service-data]
user@host# set polling-interval polling-interval
```

14. (Optional) Verify your configuration.

```
[edit shared acp group config configuration ldap service-data]
user@host# show
```

Configuring SRC ACP Scripts and Classification

Use the following configuration statements to configure SRC ACP scripts and classification:

```
shared acp configuration scripts-and-classification {
  script-factory-class script-factory-class;
  classification-factory-class classification-factory-class;
  classification-script classification-script;
  congestion-point-profile-script congestion-point-profile-script;
  extension-path extension-path;
}
```

To configure scripts and classification:

1. From configuration mode, access the configuration statement that configures SRC ACP scripts and classification. In this sample procedure, the properties are configured in the config group.

```
user@host# edit shared acp group config configuration scripts-and-classification
```

2. Specify the script factory class name.

```
[edit shared acp group config configuration scripts-and-classification]
user@host# set script-factory-class script-factory-class
```

3. Specify the congestion point classifier factory class name.

```
[edit shared acp group config configuration scripts-and-classification]
user@host# set classification-factory-class classification-factory-class
```

4. Specify the class name for congestion point classification.

```
[edit shared acp group config configuration scripts-and-classification]
user@host# set classification-script classification-script
```

5. Specify the class name for generating the congestion point DN by using the congestion point profile.

```
[edit shared acp group config configuration scripts-and-classification]
user@host# set congestion-point-profile-script congestion-point-profile-script
```

6. Specify the extension class path for classes not located in the */opt/UMC/acp/lib* directory.

```
[edit shared acp group config configuration scripts-and-classification]
user@host# set extension-path extension-path
```


7. (Optional) Verify your configuration.

```
[edit shared acp group config configuration scripts-and-classification]
user@host# show
```

Related Documentation

- [Configuring SRC ACP Properties \(C-Web Interface\)](#)
- [Configuring Local Properties for SRC ACP \(SRC CLI\) on page 263](#)
- [Configuring SRC ACP \(SRC CLI\) on page 262](#)
- [Configuring ACP to Store Log Messages in a File \(C-Web Interface\)](#)

Configuring SRC ACP to Manage the Edge Network (SRC CLI)

The tasks to configure SRC ACP to manage the edge network are:

- [Configuring Network Interfaces in the Directory for the Edge Network on page 281](#)
- [Configuring Bandwidths for Subscribers on page 282](#)
- [Assigning Network Interfaces to Subscribers on page 283](#)
- [Configuring Bandwidths for Services in the Edge Network on page 284](#)

Configuring Network Interfaces in the Directory for the Edge Network

You must add network interfaces to the directory. For the edge network, you do so by specifying the network interfaces of the routers and the switches in the access network between subscribers and the SRC network.

Use the following configuration statements to configure a network interface:

```
shared admission-control device name {
  description description;
}
shared admission-control device name interface name {
  description description;
  upstream-provisioned-rate upstream-provisioned-rate;
  downstream-provisioned-rate downstream-provisioned-rate;
  upstream-background-bandwidth upstream-background-bandwidth;
  downstream-background-bandwidth downstream-background-bandwidth;
  detect-link-rate;
}
```

To configure the network interfaces of the routers and the switches in the access network:

1. From configuration mode, access the configuration statement that configures network interfaces.

```
user@host# edit shared admission-control device name
```

Enter the name of the network device.

2. (Optional) Specify a description for the network device.

```
[edit shared admission-control device name]
user@host# set description description
```

3. Specify the network interface.

```
user@host# edit shared admission-control device name interface name
```

Enter the name of the virtual router.

4. (Optional) Specify the provisioned bandwidth for the network interface.

```
[edit shared admission-control device name interface name]  
user@host# set upstream-provisioned-rate upstream-provisioned-rate  
user@host# set downstream-provisioned-rate downstream-provisioned-rate
```

5. (Optional) Specify the background bandwidth for the network interface.

```
[edit shared admission-control device name interface name]  
user@host# set upstream-background-bandwidth upstream-background-bandwidth  
user@host# set downstream-background-bandwidth  
          downstream-background-bandwidth
```

For information about background bandwidths, see [“Allocating Bandwidth to Applications Not Controlled by SRC ACP” on page 254](#).

6. (Optional) Specify whether SRC ACP detects the link rate for the network interface.

```
[edit shared admission-control device name interface name]  
user@host# set detect-link-rate
```

If you set this option, specify `portId` as an index key when configuring SRC ACP operations so that updated sync rates are provided from interface tracking events. If the sync rate is not available, then the provisioned bandwidth configured in the subscriber profile is used.

7. (Optional) Verify your configuration.

```
[edit shared admission-control device name interface name]  
user@host# show
```

Configuring Bandwidths for Subscribers

You must configure bandwidths for subscribers that SRC ACP manages in the edge region of the network.

If the access network between the subscriber and the router uses ATM, and all the traffic coming from one DSLAM travels on a single virtual path, you do not need to provision bandwidths for each subscriber. In this case, SRC ACP can derive the congestion points from the router (see [“Deriving Congestion Points Automatically” on page 251](#)).

However, if the access network uses a protocol other than ATM, you must provide the following information for each subscriber.

- Provisioned downstream bandwidth
- Provisioned upstream bandwidth
- Actual downstream bandwidth for the current subscriber session

- Actual upstream bandwidth for the current subscriber session
- List of DNS of interfaces associated with congestion points

To configure bandwidths for subscribers:

1. From configuration mode, access the configuration statement that configures residential subscribers.

```
user@host# edit subscribers retailer name subscriber-folder folder-name subscriber
name admission-control
```

For more information about configuring residential subscribers, see *Adding Residential Subscribers (SRC CLI)*.

2. (Optional) Specify the provisioned downstream bandwidth. This rate is used if the subscriber bandwidth settings are not provided by remote update (through the API for ACP) or by the **downstream-sync-rate** value.

```
[edit subscribers retailer name subscriber-folder folder-name subscriber name
admission-control]
user@host# set downstream-provisioned-rate downstream-provisioned-rate
```

3. (Optional) Specify the provisioned upstream bandwidth. This rate is used if the subscriber bandwidth settings are not provided by remote update (through the API for ACP) or by the **upstream-sync-rate** value.

```
[edit subscribers retailer name subscriber-folder folder-name subscriber name
admission-control]
user@host# set upstream-provisioned-rate upstream-provisioned-rate
```

4. (Optional) Specify the actual downstream bandwidth for the current subscriber session. If you do not set this value and it is not provided by remote update (through the API for ACP), then the **downstream-provisioned-rate** value is used.

```
[edit subscribers retailer name subscriber-folder folder-name subscriber name
admission-control]
user@host# set downstream-sync-rate downstream-sync-rate
```

5. (Optional) Specify the actual upstream bandwidth for the current subscriber session. If you do not set this value and it is not provided by remote update (through the API for ACP), then the **upstream-provisioned-rate** value is used.

```
[edit subscribers retailer name subscriber-folder folder-name subscriber name
admission-control]
user@host# set upstream-sync-rate upstream-sync-rate
```

Assigning Network Interfaces to Subscribers

You must assign to the subscriber object interfaces (including the router interfaces) for all congestion points between the subscriber and the router.



NOTE: You must define the interface in the directory before you can assign it to a residential subscriber (see [“Configuring Network Interfaces in the Directory for the Edge Network”](#) on page 281).

To assign an interface:

1. From configuration mode, access the configuration statement that configures residential subscribers.

```
user@host# edit subscribers retailer name subscriber-folder folder-name subscriber
name admission-control
```

For more information about configuring residential subscribers, see *Adding Residential Subscribers (SRC CLI)*.

2. (Optional) Specify the DNs of interfaces associated with congestion points for this subscriber.

```
[edit subscribers retailer name subscriber-folder folder-name subscriber name
admission-control]
user@host# set congestion-points [congestion-points...]
```

Configuring Bandwidths for Services in the Edge Network

Upstream and downstream bandwidths must be specified for services that SRC ACP manages. You can obtain bandwidths for services in two ways:

- Provide static values through the directory.
- Allow the values to be provided through the SAE core API.

For example, a business partner may need to specify the required values for a particular piece of content through the SAE core API.

To configure values for services:

1. From configuration mode, access the configuration statement that configures services.

```
user@host# edit services global service name admission-control
```

For more information about configuring services, see *Services for the SRC Software Overview*.

2. (Optional) Specify the required downstream and upstream bandwidths.

```
[edit services global service name admission-control]
user@host# set required-downstream-bandwidth required-downstream-bandwidth
user@host# set required-upstream-bandwidth required-upstream-bandwidth
```

Related Documentation

- [Configuring SRC ACP to Manage the Edge Network \(C-Web Interface\)](#)
- [Configuring SRC ACP to Manage the Backbone Network \(SRC CLI\) on page 285](#)

- [Viewing Information About Subscriber Sessions in the Edge Network \(SRC CLI\) on page 311](#)
- [SRC ACP Overview on page 249](#)

Configuring SRC ACP to Manage the Backbone Network (SRC CLI)

The tasks to configure SRC ACP to manage the backbone network are:

- [Configuring Network Interfaces in the Directory for the Backbone Network on page 285](#)
- [Extending SRC ACP Congestion Points for the Backbone Network on page 285](#)
- [Configuring Action Congestion Points on page 286](#)
- [Configuring Bandwidths for Services in the Backbone Network on page 287](#)
- [Configuring Congestion Points for Services in the Backbone Network on page 287](#)
- [Using Functions for Backbone Congestion Point Classification Scripts on page 292](#)
- [Configuring Congestion Point Profiles in the Directory on page 293](#)
- [Assigning Interfaces to Congestion Point Profiles on page 293](#)

Configuring Network Interfaces in the Directory for the Backbone Network

You configure network interfaces in the directory in the same way for edge and backbone congestion points.

- For backbone congestion points, add only VRs and their interfaces. For information about this procedure, see “[Configuring Network Interfaces in the Directory for the Edge Network](#)” on page 281.

Extending SRC ACP Congestion Points for the Backbone Network

You can extend SRC ACP congestion points to initialize and execute applications defined in a backbone congestion point.

SRC ACP provides a service provider interface (SPI) to:

- Create custom congestion point applications that authorize service activation and track service start and stop events.
- Obtain congestion point information from remote update.
- Retrieve congestion point status.
- Track congestion point state.

The SPI for ACP provides a Java interface that a congestion point application implements. For information about the SPI for ACP, see the SDK documentation in the **SDK+AppSupport+Demos+Samples.tar.gz** file on the Juniper Networks Web site at: <https://www.juniper.net/support/products/src/index.html> You can locate the files in the **SDK/doc/acp** directory.

The implementation of the SPI for ACP can be a customized application that performs certain tasks, such as creating or removing congestion points on the router. SRC ACP acts as an interface tracking plug-in, and interface tracking events are treated as remote updates for congestion points when they are created, modified, or removed.

SRC ACP supports applications written in Java or Jython. For scripts written in Java, you must compile and package the implemented SPI for ACP to make it available for use by SRC ACP. A Java implementation can include more than one Java archive (JAR) file.

To use congestion point applications with SRC ACP, configure an action congestion point that references the script.

Configuring Action Congestion Points

You can define an application in a backbone congestion point so that SRC ACP can execute it in a predefined manner. Backbone congestion points that are configured to run an application are called action congestion points. If you want to use an action congestion point to execute an application that requires real-time congestion point status, you must enable SRC ACP state synchronization with the SAE).

Before you configure an action congestion point, make sure that you know the location of the application file.

Use the following configuration statements to configure action congestion points:

```
shared admission-control device name interface name {  
    action-type (url | python | java-class | java-archive);  
    action-class-name action-class-name;  
    action-file-url action-file-url;  
    action-parameters [action-parameters...];  
    action-file-name action-file-name;  
}
```

To configure an action congestion point:

1. From configuration mode, access the configuration statement that configures network interfaces.

```
user@host# edit shared admission-control device name interface name
```

Enter the name of the network device and the name of the virtual router.

2. (Optional) Specify the file type of the application.

```
[edit shared admission-control device name interface name]  
user@host# set action-type (url | python | java-class | java-archive);
```

3. (Optional) Specify the name of the class implementing the SPI.

```
[edit shared admission-control device name interface name]  
user@host# set action-class-name action-class-name
```

4. (Optional) Specify the URL or the content of the file. For action congestion point implementations written in Java of the url action type, configure the URL that specifies the location of the Java archives (.jar files) containing the action congestion point

implementation. For other action types, you must load the action congestion point implementation with the action file name option.

```
[edit shared admission-control device name interface name]
user@host# set action-file-url action-file-url
```

5. (Optional) Specify the parameter as an attribute=value pair.

```
[edit shared admission-control device name interface name]
user@host# set action-parameters [action-parameters...]
```

6. (Optional) Load the local file that contains the action congestion point implementation. This file is the uncompiled Python source code or the compiled result of the Java file (binary *.class* or *.jar* file).

```
[edit shared admission-control device name interface name]
user@host# set action-file-name action-file-name
```

7. (Optional) Verify your configuration.

```
[edit shared admission-control device name interface name]
user@host# show
```

Configuring Bandwidths for Services in the Backbone Network

To configure bandwidths for services in the same way for edge and backbone congestion points:

- See “Configuring SRC ACP to Manage the Edge Network (SRC CLI)” on page 281.

Configuring Congestion Points for Services in the Backbone Network

You must assign a congestion point to each service that SRC ACP manages. When SRC ACP receives a service authorization event, congestion points for a service session can be determined by:

- Congestion point classification
- Congestion point profiles

To configure congestion points with congestion point classification:

1. From configuration mode, access the configuration statement that configures services.

```
user@host# edit services global service name admission-control
congestion-point-classification
```

For more information about services, see *Services for the SRC Software Overview*.

2. Specify the backbone congestion point expression.

```
[edit services global service name admission-control congestion-point-classification]
user@host# set expression [expression...]
```

The syntax for a backbone congestion point expression is defined in the format `<NetworkDevice>/<NetworkInterface>/<InstanceID>` which maps to a congestion point.

- `<NetworkDevice>`—Network device listed in the directory.
- `<NetworkInterface>`—Network interface listed in the directory.
- `<InstanceID>`—Name of an instance of a congestion point that is automatically created.

For information about congestion point expressions, see [“Congestion Point Expressions” on page 304](#). For information about the attributes that can be embedded in the expression, see [“Plug-In Attributes for Use with Backbone Congestion Point Expressions” on page 289](#).

3. (Optional) Specify the backbone congestion point script.

```
[edit services global service name admission-control congestion-point-classification]
user@host# set script script
```

For information about congestion point functions, see [“Using Functions for Backbone Congestion Point Classification Scripts” on page 292](#).

To configure congestion points with congestion point profiles:

1. From configuration mode, access the configuration statement that configures services.

```
user@host# edit services global service name admission-control
```

For more information about services, see *Services for the SRC Software Overview*.

2. (Optional) Specify the backbone congestion points. This value is ignored if you configure congestion points with congestion point classification.

```
[edit services global service name admission-control]
user@host# set congestion-points [congestion-points...]
```

The backbone congestion point is defined in the format `<-vrName->/<-serviceName->`, which locates a congestion point profile that contains a list of congestion points.

- To allow the software to automatically define the congestion point, use the entry `<-vrName->/<-serviceName->`. When SRC ACP starts operating, it will substitute the VR name and the service name from the request for service activation.
- To restrict the congestion point to a specific VR or service, enter the actual VR name or service name.

Plug-In Attributes for Use with Backbone Congestion Point Expressions

These plug-in attributes must be available for service authorization and service tracking events.

accountingId

- Value of accountingUserId attribute.

ifRadiusClass

- RADIUS class attribute on the JunosE interface.
- Value—String array
- Example—ifRadiusClass="acpe"

ifSessionId

- Identifier for RADIUS accounting on the JunosE interface.

interfaceAlias

- Description of the interface.
- Value—Interface description that is configured on the JunosE router with the **interface ip description** command
- Example—interfaceAlias="dhcp-subscriber12"

interfaceDescr

- Alternate name for the interface that is used by SNMP. This name is a system-generated name.
- Value
 - On a JunosE router, the format of the description is
ip<slot>/<port>.<subinterface>
 - On the device running Junos OS, interfaceDescr is the same as interfaceName.
- Example—interfaceDescr="IP3/1"

interfaceName

- Name of the interface.
- Value
 - Name of the interface in your router CLI syntax
 - FORWARDING_INTERFACE for routing instance (used by traffic mirroring)
- Example—For JunosE routers: interfaceName="fastEthernet6/0"
For devices running Junos OS: interfaceName="fe-0/1/0.0"
For forwarding interface: interfaceName="FORWARDING_INTERFACE"

localQosProfiles.<layer name>

- Local QoS profile in the specified layer. Local QoS profiles refer to profiles that are attached using the JunosE router CLI or the Service Manager and not through a SAE.
- Value—String
 - The <layer name> is one of the following values: ip, ipv6, lac, svlan, vlan, ethernet, atmVp, atmVc, atm, bridge, frVc, ipTunnel, l2tpTunnel.
- Example—Specifying "localQosProfiles.vlan" returns the name of the QoS profile in the VLAN layer.

loginName

- Subscriber's login name.
- Value—Login name
- Guidelines—The format of the login name varies. A loginName can be of form subscriber, domain\subscriber, subscriber@domain, or as otherwise defined by the login setup of the manager.
- Example—idp@idp

nasIp

- IP address of the router.
- Value—String

nasPort

- Numeric identifier that the router uses to identify the interface to RADIUS.
- Value—Integer

portId

- Port identifier of an interface.
- Value—Includes interface name and additional layer 2 information
- Example—portId=" fastEthernet 3/1" (There is a space between fastEthernet and slot number 3/1 in the nasPort field.)

primaryUserName

- PPP login name or the public DHCP username.
- Value—Subscriber name
- Example—primaryUserName=" peter"

radiusClass

- RADIUS class attribute of the service definition.
- Value—String
- Example—radiusClass=" Premium"

serviceName

- Identifier of the service.

serviceScope

- Identifier of the service scope.

serviceSessionName

- Identifier of the service session.

serviceSessionTag

- Tag for the service session.

sspHost

- Name of host on which the SAE is installed.

substitutions.<substitution name>

- Substitution with the specified name passed in at service activation.

userIp

- IP address of the subscriber.
- Value—String

userMacAddress

- Media access control (MAC) address of the DHCP subscriber.
- Value—Valid MAC address
- Example—userMacAddress=" 00:11:22:33:44:55"

userType

- Type of subscriber.

vrName

- Name of virtual router.
- Value—Virtual router name in the format <virtualRouter>@<router>
- Example—vrName="default@e_series5"

Using Functions for Backbone Congestion Point Classification Scripts

SRC ACP provides the following functions to use in backbone congestion point classification scripts:

- **getNicProxy(name)**—Get the NIC proxy defined under the current SRC ACP configuration group.
 - name—The name of the NIC proxy as defined under the SRC ACP configuration group.
- **nicLookupSingle(name, nicKey, constraints)**—Perform a NIC lookup using the specified NIC key and constraints with the NIC proxy defined under the current SRC ACP shared configuration group. The NIC key must uniquely identify a NIC value. If more than one result matches the same key, this function will raise the `AmbiguousKeyException` exception.
 - name—Name of the NIC proxy.
 - nicKey—String used as key for NIC lookup.
 - constraints (optional)—Map of NIC constraint information associated with the NIC key.

This function returns the lookup result as (nicValue, intermediateValues), where intermediateValues is a map of the intermediate name and value pair.

- **nicLookup(name, nicKey, constraints)**—Perform a NIC lookup using the specified NIC key and constraints for the NIC proxy defined under the current SRC ACP shared configuration group.
 - name—Name of the NIC proxy.
 - nicKey—String used as key for NIC lookup.
 - constraints (optional)—Map of NIC constraint information associated with the NIC key.

This function returns the lookup result as an array of (nicValue, intermediateValues), where intermediateValues is a map of the intermediate name and value pair.

- `nicInvalidateLookup(name, nicKey, nicValue, constraints)`--Used to signal to a NIC proxy that a key/value pair (returned from one of the lookup methods) resulted in a failure when the value was used. If the NIC proxy has this result cached, it will be removed from the cache.
 - `name`—Name of the NIC proxy.
 - `nicKey`—A string used as NIC key that was passed to the previous lookup operation.
 - `nicValue`—The NIC value returned from the previous lookup operation.
 - `constraints(optional)`—Map of NIC constraint information associated with the NIC key.
- `slot(nasPortId)`—Collects the slot number from the `nasPortId` or `interfaceName`.
- `port(nasPortId)`—Collects the port number from the `nasPortId` or `interfaceName`.
- `l2id(nasPortId)`—Collects the layer 2 ID from the `nasPortId` (VLAN id or ATM vpi.vci).
- `escape(string)`—Replaces any slash with the escape sequence `\`.

Configuring Congestion Point Profiles in the Directory

If you are using congestion point classification, you do not need to configure congestion point profiles.

To configure individual backbone congestion point profiles:

1. From configuration mode, access the configuration statement that configures congestion point profiles.

```
user@host# edit shared congestion-points profile name
```

Enter the name of the virtual router that supports the congestion point.

2. (Optional) Verify your configuration.

```
[edit shared congestion-points profile name]
user@host# show
```

Assigning Interfaces to Congestion Point Profiles

If you are using congestion point classification, you do not need to assign interfaces to congestion point profiles.

You must assign interfaces either to VRs or to individual services under the VRs. Services inherit interface assignments from the associated VR unless you assign an interface to the individual service. This network interface lists the DNs of interfaces associated with backbone congestion point profiles.

Use the following configuration statements to configure interface assignments:

```
shared congestion-points profile name {
```

```
interface [interface...];  
}
```

To assign interfaces to congestion point profiles:

1. From configuration mode, access the configuration statement that configures congestion point profiles.

```
user@host# edit shared congestion-points profile name
```

Enter the name of the network device to which you want to assign the congestion point profile.

2. (Optional) Specify the interfaces associated with a congestion point profile for this subscriber.

```
[edit shared congestion-points profile name]  
user@host# set interface interface
```

3. (Optional) Verify your configuration.

```
[edit shared congestion-points profile name]  
user@host# show
```

Related Documentation

- [Configuring SRC ACP to Manage the Backbone Network \(C-Web Interface\)](#)
- [Configuring SRC ACP to Manage the Edge Network \(SRC CLI\) on page 281](#)
- [Viewing Information About Services in the Backbone Network \(SRC CLI\) on page 313](#)
- [SRC ACP Overview on page 249](#)

Defining SRC ACP Congestion Point Usage Trap Thresholds (SRC CLI)

Four alarms are used for the ACP congestion point usage trap (acpCPUsage): critical, major, minor, and clear trap. Critical, major, and minor traps are sent when the corresponding thresholds are exceeded. The clear trap is sent when the previous event is cleared. When the sampled value falls below a higher threshold but is still above a lower threshold, a clear trap is sent followed by a trap corresponding to the lower threshold.

Objects sent with the traps include the congestion point ID (congestion point DN plus instance ID, which is optional), the CP's upstream bandwidth, downstream bandwidth, upstream bandwidth in use, and downstream bandwidth in use.

To avoid configuring thresholds for every congestion point, the threshold is defined as a percentage of in-use bandwidth, which is calculated with the following expression:

$$\text{used bandwidth} / (\text{total bandwidth} - \text{background bandwidth})$$

Where, total bandwidth is usually the provisioned bandwidth of the congestion point but can be overridden if the ACP receives an update through L2C or RemoteUpdateInterface.

A congestion point's upstream bandwidth and downstream bandwidth are evaluated with this expression and compared to the configured threshold separately. If either the upstream or downstream bandwidth of the congestion point exceeds a threshold, a trap corresponding to the threshold is generated. A clear trap is sent only when the usage of both the upstream and downstream bandwidth fall below the threshold. The same trap is not sent multiple times consecutively.

Use the following configuration statements to configure the acpCPUUsage trap:

```
shared acp configuration snmp {
  selector [selector...];
  critical-threshold critical-threshold;
  major-threshold major-threshold;
  minor-threshold minor-threshold;
}
```

To configure the acpCPUUsage trap:

1. From configuration mode, access the configuration statement that configures the acpCPUUsage trap.

```
user@host# edit shared acp configuration snmp
```

2. (Optional) Configure Java regular expressions to match against congestion point DN. A congestion point is chosen for notification if its DN matches any one of the expressions. If not specified, all congestion points are selected.

```
[edit shared acp configuration snmp]
user@host# set selector selector;
```

3. Configure the critical threshold for the congestion point usage trap. The threshold is a percentage of used bandwidth out of the total accessible bandwidth, which is the current bandwidth minus the background bandwidth of a congestion point.

```
[edit shared acp configuration snmp]
user@host# set critical-threshold critical-threshold ;
```

4. Configure the major threshold for the congestion point usage trap. The threshold is a percentage of used bandwidth out of the total accessible bandwidth, which is the current bandwidth minus the background bandwidth of a congestion point.

```
[edit shared acp configuration snmp]
user@host# set major-threshold major-threshold ;
```

5. Configure the minor threshold for the congestion point usage trap. The threshold is a percentage of used bandwidth out of the total accessible bandwidth, which is the current bandwidth minus the background bandwidth of a congestion point.

```
[edit shared acp configuration snmp]
user@host# set minor-threshold minor-threshold ;
```

Related Documentation

- [SNMP Traps Overview](#)

- *Event Traps*
- *Decoding Trap Numbers for Raised Trap Actions*
- *Decoding Trap Numbers for Clear Trap Actions*

CHAPTER 20

Configuring Congestion Point Classification (SRC CLI)

- [Congestion Point Classification Overview on page 297](#)
- [Configuration Statements for Congestion Point Classification on page 298](#)
- [Classifying Congestion Points \(SRC CLI\) on page 298](#)
- [Defining a Congestion Point Profile \(SRC CLI\) on page 304](#)
- [Congestion Point Expressions on page 304](#)

Congestion Point Classification Overview

Congestion point classification allows you to automate and scale the configuration of congestion points. SRC ACP uses classification scripts to determine which congestion point to load for a subscriber. SRC ACP can select the congestion point from congestion point profiles or subscriber profiles.

Congestion Point Classification Scripts

The congestion point classification scripts consist of targets and criteria.

- A target is the result of the classification script. The result of congestion point classification scripts is an LDAP search string that is used to find a unique congestion point in the directory. If no classification scripts are configured, the result of congestion point classification scripts is an LDAP search string for the subscriber profile of the particular subscriber.
- Criteria are match criteria. The script attempts to match criteria in the script to information sent from the router. Match criteria for a congestion point classification script might be a subscriber distinguished name (DN) or an interface name.

Each script can have multiple targets, and each target can have multiple criteria. When an object needs classification, the script processes the targets in turn. Within each target, the script processes criteria sequentially. When it finds that the classification criteria for a target match, it returns the target to SRC ACP.

Because classification scripts examine criteria sequentially as the criteria appear in the script, you should put more specific criteria at the beginning of the script and less specific criteria at the end of the script.

Congestion Point Profiles

Congestion point profiles are used to share congestion points that are generated based on dynamic configuration information. SRC ACP uses congestion point profiles to determine the set of congestion points based on the classification script results.

Changes that you make to classification scripts do not affect subscriber sessions that are already established.

Related Documentation

- [SRC ACP Overview on page 249](#)
- [Classifying Congestion Points \(SRC CLI\) on page 298](#)
- [Congestion Point Classification Criteria on page 300](#)
- [Configuration Statements for Congestion Point Classification on page 298](#)

Configuration Statements for Congestion Point Classification

Use the following configuration statements to configure congestion point classification at the **[edit]** hierarchy level.

```
shared acp congestion-point-classifier rule name {  
    target target;  
    script script;  
}  
shared acp congestion-point-classifier rule name condition name ...  
shared congestion-points congestion-point-profile name {  
    expression [expression...];  
}
```

For detailed information about each configuration statement, see the *SRC PE CLI Command Reference*.

Related Documentation

- [Classifying Congestion Points \(SRC CLI\) on page 298](#)
- [Defining a Congestion Point Profile \(SRC CLI\) on page 304](#)
- [Congestion Point Expressions on page 304](#)
- [Congestion Point Classification Overview on page 297](#)

Classifying Congestion Points (SRC CLI)

The tasks to classify congestion points are:

1. [Configuring Targets and Criteria for Classification Scripts on page 298](#)
2. [Configuring Classification Scripts Contents for Classification Scripts on page 299](#)
3. [Configuring Congestion Point Classification Targets on page 299](#)

Configuring Targets and Criteria for Classification Scripts

To define a target and criteria for the congestion point classification script:

1. From configuration mode, access the configuration statement that configures congestion point scripts. In this sample procedure, the scripts are configured in the config group.

```
user@host# edit shared acp group config congestion-point-classifier rule name
```

Enter a name for the congestion point classification script.

2. Specify the target for the classification script.

```
[edit shared acp group config congestion-point-classifier rule name]
user@host# set target target
```

3. Specify the classification criteria for the target.

```
[edit shared acp group config congestion-point-classifier rule name]
user@host# set condition condition
```

For information about classification criteria, see [“Congestion Point Classification Criteria” on page 300](#).

Configuring Classification Scripts Contents for Classification Scripts

To use the contents of a classification script to another object for the congestion point classification script:

1. From configuration mode, access the configuration statement that configures congestion point scripts. In this sample procedure, the scripts are configured in the config group.

```
user@host# edit shared acp group config congestion-point-classifier rule name
```

Enter a name for the congestion point classification script.

2. Specify the classification script that you want to use.

```
[edit shared acp group config congestion-point-classifier rule name]
user@host# set script script
```

Configuring Congestion Point Classification Targets

The target of the congestion point classification script is an LDAP search string. The search string uses a syntax similar to an LDAP URL (see RFC 2255—The LDAP URL Format (December 1997)). The syntax is:

```
baseDN [ ? [ attributes ] [ ? [ scope ] [ ? [ filter ] ] ] ]
```

- baseDN—Distinguished name (DN) of the object where the LDAP search starts.
- attributes—Is ignored.
- scope—Scope of search in the directory:
 - base—Default; searches the base DN only.

- one—Searches the direct children of the base DN.
- sub—Searches the complete subtree below the base DN.
- filter—An RFC 2254–style LDAP search filter expression; for example, (uniqueId=<-userName->). See RFC 2254—The String Representation of LDAP Search Filters (December 1997).

With the exception of baseDN all the fields are optional.

The result of the LDAP search must be exactly one directory object. If no object or more than one object is found, congestion points for the subscriber are not loaded and all service activations for the subscriber are denied.

Congestion Point Classification Criteria

Congestion point classification criteria define match criteria that are used to find the congestion point profile. Use the fields in this topic to define classification criteria.

accountingId

- Value of directory attribute accountingUserId.

authUserId

- Identifier that a subscriber uses for authentication.
- Value—Username

dhcpPacket

- Content of the DHCP discover request.
- Value—Byte array
 - First 4 octets—Gateway IP address (giaddr field)
 - Remaining octets—DHCP options

For more information, see RFC 2131—Dynamic Host Configuration Protocol (March 1997) and RFC 2132—DHCP Options and BOOTP Vendor Extensions (March 1997).

domain

- Name of the domain used for secondary authentication.
- Value—Valid domain name
- Example—domain=" isp99.com"

ifRadiusClass

- RADIUS class attribute on the JunosE interface.
- Value—RADIUS class name
- Example—ifRadiusClass=" acpe"

ifSessionId

- Identifier for RADIUS accounting on the JunosE interface.

interfaceAlias

- Description of the interface.
- Value—Interface description that is configured on the JunosE router with the **interface ip description** command
- Example—interfaceAlias=" dhcp-subscriber12"

interfaceDescr

- Alternate name for the interface that is used by SNMP. This name is a system-generated name.
- Value
 - On a JunosE router, the format of the description is
ip<slot>/<port>.<subinterface>
 - On the device running Junos OS, interfaceDescr is the same as interfaceName.
- Example—interfaceDescr=" IP3/1"

interfaceName

- Name of the interface.
- Value
 - Name of the interface in your router CLI syntax
 - FORWARDING_INTERFACE for routing instance (used by traffic mirroring)
- Example—For JunosE routers: interfaceName=" fastEthernet6/0"
For devices running Junos OS: interfaceName="fe-0/1/0.0"
For forwarding interface: interfaceName="FORWARDING_INTERFACE"

localQosProfiles.<layer name>

- Local QoS profile in the specified layer. Local QoS profiles refer to profiles that are attached using the JunosE router CLI or the Service Manager and not through a SAE.
- Value—String
 - The <layer name> is one of the following values: ip, ipv6, lac, svlan, vlan, ethernet, atmVp, atmVc, atm, bridge, frVc, ipTunnel, l2tpTunnel.
- Example—Specifying "localQosProfiles.vlan" returns the name of the QoS profile in the VLAN layer.

loginName

- Subscriber's login name.
- Value—Login name
- Guidelines—The format of the login name varies. A loginName can be of form subscriber, domain\subscriber, subscriber@domain, or as otherwise defined by the login setup of the manager.
- Example—idp@idp

nasIp

- IP address of the router.
- Value—Byte array
 - For IPv4 address—4 octets in network byte order
 - For IPv6 address—16 octets in network byte order

nasPort

- Numeric identifier that the router uses to identify the interface to RADIUS.
- Value—Integer

portId

- Port identifier of an interface.
- Value—Includes interface name and additional layer 2 information
- Example—portId=" fastEthernet 3/1" (There is a space between fastEthernet and slot number 3/1 in the nasPort field.)

primaryUserName

- PPP login name or the public DHCP username.
- Value—Subscriber name
- Example—primaryUserName=" peter"

radiusClass

- RADIUS class attribute of the service definition.
- Value—RADIUS class name
- Example—radiusClass=" Premium"

routerName

- Name of virtual router.
- Value—Virtual router name in the format <virtualRouter>@<router>
- Example—routerName=" default@e_series5"

sessionId

- Identifier of RADIUS session for the subscriber session.

serviceBundle

- Content of the RADIUS vendor-specific attribute for the service bundle.
- Value—Name of a service bundle
- Example—serviceBundle=" goldSubscriber"

sppHost

- Name of host on which the SAE is installed.

userDn

- DN of a subscriber in the directory.
- Value—DN of a subscriber profile

userIp

- IP address of the subscriber.
- Value—Byte array
 - For IPv4 address—4 octets in network byte order
 - For IPv6 address—16 octets in network byte order

userMacAddress

- Media access control (MAC) address of the DHCP subscriber.
- Value—Valid MAC address
- Example—userMacAddress=" 00:11:22:33:44:55"

userType

- Type of subscriber.

Related Documentation

- [Configuring Congestion Point Classification \(C-Web Interface\)](#)
- [Configuration Statements for Congestion Point Classification on page 298](#)
- [Viewing Congestion Point Information by DN \(SRC CLI\) on page 326](#)
- [Congestion Point Expressions on page 304](#)

- [Congestion Point Classification Overview on page 297](#)

Defining a Congestion Point Profile (SRC CLI)

You can create a congestion point profile that automatically performs congestion point classification. This profile supports only access network mode for SRC ACP.

Use the following configuration statements to configure congestion point profiles:

```
shared congestion-points congestion-point-profile name {  
    expression [expression...];  
}
```

To define a congestion point profile:

1. From configuration mode, access the configuration statement that configures congestion point profiles.

```
user@host# edit shared congestion-points congestion-point-profile name
```

Enter a name for the profile.

2. Specify congestion point expressions.

```
[edit shared congestion-points congestion-point-profile name]  
user@host# set expression [expression...]
```

For information about congestion point expressions, see [“Congestion Point Expressions” on page 304](#).

Related Documentation

- [Defining a Congestion Point Profile \(C-Web Interface\)](#)
- [Classifying Congestion Points \(SRC CLI\) on page 298](#)
- [Configuration Statements for Congestion Point Classification on page 298](#)
- [Congestion Point Classification Overview on page 297](#)

Congestion Point Expressions

You can enter a congestion point expression by using the syntax listed in this topic. You can also embed Python scripting expressions within the congestion point expression.

If you embed Python expressions within a congestion point expression, use the escape sequence `<- then ->` to enclose the Python expression. See [“Methods for Use with Scripting Expressions” on page 305](#) and [“Match Criteria for Congestion Point Classification” on page 305](#).

The syntax for a congestion point expression is:

```
<NetworkDevice>/<NetworkInterface>[/<CongestionPoint>]
```

- `<NetworkDevice>`—Network device listed in the directory.

- `<NetworkInterface>`—Network interface listed in the directory.

For information about interfaces, see *Classification Scripts Overview*.

- `<CongestionPoint>`—(Optional) Name of an instance of a congestion point that is automatically created.

If one of the elements with the path contains a slash (/), use a backslash (\) as an escape character for the slash. For example, \.

Expressions in Templates for Congestion Point Profiles

You can create a congestion point profile to be used as a template for other profiles. Templates simplify management of congestion points. Rather than configuring each congestion point individually, you can create templates to define common parameters for a class of individual congestion points.

For example, in an environment in which VLAN interfaces GigabitEthernet1/0.1 through GigabitEthernet1/0.1000 have the same available bandwidth, you can specify the characteristics of the VLAN interface once and have SRC ACP create the congestion points based on the template configuration.

When a congestion point expression has the third element (`<CongestionPoint>`), SRC ACP uses the `<NetworkDevice>/<NetworkInterface>` part of the expression to load the congestion point from the directory, and uses it as a template to create a congestion point in memory for subscriber. The `<CongestionPoint>` part of the expression distinguishes each congestion point (available bandwidth) created from this template.

Methods for Use with Scripting Expressions

SRC ACP provides the following methods to use in scripting expressions:

- `slot(nasPortId)`—Collects the slot number from the `nasPortId` or `interfaceName`
Example—`slot(" atm 4/5:0.32") == " 4"`
- `port(nasPortId)`—Collects the port number from the `nasPortId` or `interfaceName`
Example—`port(" atm 4/5:0.32") == " 5"`
- `l2id(nasPortId)`—Collects the layer 2 ID from the `nasPortId` (VLAN id or ATM vpi.vci)
Example—`l2id(" atm 4/5:0.32") == " 0.32"`
- `escape(string)`—Replaces any slash with the escape sequence \.
Example—`escape("atm 4/5") == "atm 4\5"`

Match Criteria for Congestion Point Classification

You can use the match criteria in Python scripting expressions for a congestion point expression. For more information about the match criteria, see [“Congestion Point Classification Criteria” on page 300](#).

Related Documentation

- [Congestion Point Classification Overview on page 297](#)
- [Classifying Congestion Points \(SRC CLI\) on page 298](#)

- [Defining a Congestion Point Profile \(SRC CLI\) on page 304](#)
- [Configuration Statements for Congestion Point Classification on page 298](#)

CHAPTER 21

Managing SRC ACP (SRC CLI)

- [Starting SRC ACP on page 307](#)
- [Stopping SRC ACP on page 307](#)
- [Reorganizing the File That Contains ACP Data on page 308](#)
- [Modifying Congestion Points on page 308](#)

Starting SRC ACP

To start SRC ACP:

```
user@host> enable component acp
```

Related Documentation

- [Stopping SRC ACP on page 307](#)
- [Configuring SRC ACP \(SRC CLI\) on page 262](#)
- [Reorganizing the File That Contains ACP Data on page 308](#)
- [Viewing General Statistics for SRC ACP \(C-Web Interface\) on page 342](#)
- [SRC ACP Overview on page 249](#)

Stopping SRC ACP

To stop SRC ACP:

```
user@host> disable component acp
```

Related Documentation

- [Starting SRC ACP on page 307](#)
- [Reorganizing the File That Contains ACP Data on page 308](#)
- [Viewing General Statistics for SRC ACP \(C-Web Interface\) on page 342](#)
- [SRC ACP Overview on page 249](#)

Reorganizing the File That Contains ACP Data

Periodically, you should reorganize the files that contain ACP data about subscribers, services, and congestion points. This action reduces the sizes of these files. To do so:

```
user@host> request acp reorganize-backup-database
```

Related Documentation

- [Stopping SRC ACP on page 307](#)
- [Starting SRC ACP on page 307](#)
- [Viewing General Statistics for SRC ACP \(C-Web Interface\) on page 342](#)
- [SRC ACP Overview on page 249](#)

Modifying Congestion Points

By default, SRC ACP does not register changes in congestion points until you stop and restart SRC ACP. To modify the congestion point associated with a service without stopping and starting SRC ACP:

1. Make sure that no subscribers have subscriptions to services that use the congestion point you want to modify.
2. From configuration mode, access the configuration statement that configures SRC ACP connections to the services' directory.

```
user@host# edit shared acp configuration ldap service-data
```

3. Specify whether SRC ACP detects changes in the backbone congestion point for a service while SRC ACP is operative.

```
[edit shared acp configuration ldap service-data]  
user@host# set reload-congestion-points
```

4. Wait for 30 seconds before you proceed to the next step.

Depending on the value of the polling interval for directory eventing, SRC ACP may take up to 30 seconds to register the change to the **reload-congestion-points** option. If you modify the congestion point before SRC ACP registers the new setting for the **reload-congestion-points** option, SRC ACP will not register the change for the congestion point.

5. Modify the congestion point in the service definition.

SRC ACP immediately registers the change.

6. From configuration mode, access the configuration statement that configures SRC ACP connections to the services' directory.

```
user@host# edit shared acp configuration ldap service-data
```

7. Specify whether SRC ACP detects changes in the backbone congestion point for a service while SRC ACP is operative.

```
[edit shared acp configuration ldap service-data]  
user@host# set reload-congestion-points
```


CHAPTER 22

Monitoring Admission Control (SRC CLI)

- Viewing Information About Subscriber Sessions in the Edge Network (SRC CLI) on page 311
- Viewing Edge Congestion Point Information by DN (SRC CLI) on page 312
- Viewing Edge Congestion Point Information by Subscriber Session (SRC CLI) on page 313
- Viewing Information About Services in the Backbone Network (SRC CLI) on page 313
- Viewing Backbone Congestion Point Information by DN (SRC CLI) on page 314
- Viewing Backbone Congestion Point Information by Service (SRC CLI) on page 315
- Viewing Congestion Point Information by Subscriber IP Address and Associated Service Sessions (SRC CLI) on page 315
- Viewing Congestion Point Information by Session ID and Associated Service Sessions (SRC CLI) on page 318
- Viewing Congestion Point Information by Login Name and Associated Service Sessions (SRC CLI) on page 321
- Viewing Action Congestion Point Information by Service (SRC CLI) on page 324
- Viewing Action Congestion Point Information by Congestion Point (SRC CLI) on page 324
- Viewing Information About Subscribers Obtained from External Applications (SRC CLI) on page 325
- Viewing Congestion Point Information by DN (SRC CLI) on page 326
- Viewing Congestion Point Information by Name (SRC CLI) on page 327
- Viewing SNMP Information for Devices (SRC CLI) on page 327
- Viewing SNMP Information for the Directory (SRC CLI) on page 328
- Viewing SNMP Information for SRC ACP (SRC CLI) on page 328

Viewing Information About Subscriber Sessions in the Edge Network (SRC CLI)

Purpose Display information about the subscriber session.

Action To display information about the current subscriber sessions in memory:

```
user@host> show acp edge subscriber
```

To display information about specific subscriber sessions:

```
user@host> show acp edge subscriber session-id session-id
```

Enter all or part of the subscriber session ID to list all matching subscriber sessions.

To display information about the subscriber sessions from a specific virtual router:

```
user@host> show acp edge subscriber virtual-router-name virtual-router-name
```

Enter a virtual router name to list subscriber sessions from a particular virtual router.

To display subscriber session attributes for the current subscriber sessions:

```
user@host> show acp edge subscriber brief
```

By default, information about the subscriber session attributes, service sessions, and associated congestion points is displayed.

**Related
Documentation**

- [Configuring SRC ACP to Manage the Edge Network \(SRC CLI\) on page 281](#)
- [Viewing Information About Subscriber Sessions in the Edge Network \(C-Web Interface\) on page 329](#)
- [Viewing Information About Subscribers Obtained from External Applications \(SRC CLI\) on page 325](#)

Viewing Edge Congestion Point Information by DN (SRC CLI)

Purpose View edge congestion point information by DN.

Action To display information about edge congestion points by DN:

```
user@host> show acp edge congestion-point dn
```

To display information about specific congestion points by DN:

```
user@host> show acp edge congestion-point dn congestion-point-dn congestion-point-dn
```

Enter a partial congestion point DN to list all matching congestion points.

To display information about specific congestion points that were generated dynamically by instance ID:

```
user@host> show acp edge congestion-point dn instance-id instance-id
```

```
user@host> show acp edge congestion-point dn congestion-point-dn congestion-point-dn  
instance-id instance-id
```

When a congestion point is dynamically generated with a congestion point profile, the generated instance ID is appended to the congestion point DN. Enter a partial instance ID to list all matching congestion points.

To display information about the congestion points from a specific virtual router:

```
user@host> show acp edge congestion-point dn virtual-router-name virtual-router-name
```

Enter a virtual router name to list congestion points from a particular virtual router.

To display congestion point DNs:

```
user@host> show acp edge congestion-point dn brief
```


By default, information about the congestion point attributes and congestion point bandwidth usage is displayed.

To restrict the number of displayed results:

```
user@host> show acp edge congestion-point dn maximum-results maximum-results
```

Related Documentation

- [Configuring SRC ACP to Manage the Edge Network \(SRC CLI\) on page 281](#)
- [Viewing Information About Edge Congestion Points by DN \(C-Web Interface\) on page 330](#)
- [Viewing Edge Congestion Point Information by Subscriber Session \(SRC CLI\) on page 313](#)

Viewing Edge Congestion Point Information by Subscriber Session (SRC CLI)

Purpose View edge congestion point information by subscriber session.

Action To display information about edge congestion points by subscriber session:

```
user@host> show acp edge congestion-point subscriber-session-id
```

To display information about specific congestion points by subscriber session:

```
user@host> show acp edge congestion-point subscriber-session-id session-id session-id
```

Enter a partial subscriber session ID to list all matching congestion points.

To display information about the congestion points from a specific virtual router:

```
user@host> show acp edge congestion-point subscriber-session-id virtual-router-name virtual-router-name
```

Enter a virtual router name to list congestion points from a particular virtual router.

To display congestion point DNs:

```
user@host> show acp edge congestion-point subscriber-session-id brief
```

By default, information about the congestion point attributes and congestion point bandwidth is displayed.

To restrict the number of displayed results:

```
user@host> show acp edge congestion-point subscriber-session-id maximum-results maximum-results
```

Related Documentation

- [Configuring SRC ACP to Manage the Edge Network \(SRC CLI\) on page 281](#)
- [Viewing Information About Edge Congestion Points by Subscriber Session \(C-Web Interface\) on page 331](#)
- [Viewing Edge Congestion Point Information by DN \(SRC CLI\) on page 312](#)

Viewing Information About Services in the Backbone Network (SRC CLI)

Purpose View information about services in the backbone network.

Action To display information about services that SRC ACP manages in the backbone network:

```
user@host> show acp backbone service
```

To display information about specific backbone service used to generate congestion points:

```
user@host> show acp backbone service service-name service-name
```

Enter a partial service name to list all matching backbone services.

To display information about the backbone services from a specific virtual router:

```
user@host> show acp backbone service virtual-router-name virtual-router-name
```

Enter a virtual router name to list backbone services from a particular virtual router.

To display backbone service attributes:

```
user@host> show acp backbone service brief
```

By default, information about the backbone service attributes, service sessions, and associated congestion points is displayed.

**Related
Documentation**

- [Configuring SRC ACP to Manage the Backbone Network \(SRC CLI\) on page 285](#)
- [Viewing Information About Services in a Backbone Network \(C-Web Interface\) on page 332](#)
- [Viewing Backbone Congestion Point Information by Service \(SRC CLI\) on page 315](#)

Viewing Backbone Congestion Point Information by DN (SRC CLI)

Purpose View backbone congestion point information by DN.

Action To display information about backbone congestion points by DN:

```
user@host> show acp backbone congestion-point dn
```

To display information about specific congestion points by DN:

```
user@host> show acp backbone congestion-point dn congestion-point-dn congestion-point-dn
```

Enter a partial congestion point DN to list all matching congestion points.

To display information about the congestion points from a specific virtual router:

```
user@host> show acp backbone congestion-point dn virtual-router-name virtual-router-name
```

Enter a virtual router name to list congestion points from a particular virtual router.

To display congestion point DNs:

```
user@host> show acp backbone congestion-point dn brief
```

By default, information about the congestion point attributes and congestion point bandwidth usage is displayed.

- Related Documentation**
- [Configuring SRC ACP to Manage the Backbone Network \(SRC CLI\) on page 285](#)
 - [Viewing Information About Congestion Points in a Backbone Network by DN \(C-Web Interface\) on page 335](#)
 - [Viewing Backbone Congestion Point Information by Service \(SRC CLI\) on page 315](#)

Viewing Backbone Congestion Point Information by Service (SRC CLI)

Purpose View backbone congestion point information by service.

Action To display information about backbone congestion points by service:

```
user@host> show acp backbone congestion-point congestion-point-expression
```

To display information about specific backbone services used to generate congestion points:

```
user@host> show acp backbone congestion-point congestion-point-expression service-name
service-name
```

Enter a partial service name to list all matching backbone services.

To display information about the backbone services from a specific virtual router:

```
user@host> show acp backbone congestion-point congestion-point-expression
virtual-router-name virtual-router-name
```

Enter a virtual router name to list backbone services from a particular virtual router.

To display congestion point DNs:

```
user@host> show acp backbone congestion-point congestion-point-expression brief
```

By default, information about the congestion point attributes and congestion point bandwidth is displayed.

- Related Documentation**
- [Configuring SRC ACP to Manage the Backbone Network \(SRC CLI\) on page 285](#)
 - [Viewing Information About Congestion Points in a Backbone Network by Expression \(C-Web Interface\) on page 334](#)
 - [Viewing Backbone Congestion Point Information by DN \(SRC CLI\) on page 314](#)

Viewing Congestion Point Information by Subscriber IP Address and Associated Service Sessions (SRC CLI)

Purpose View edge and backbone congestion point information by subscriber IP address.

The command supports looking up congestion points affecting a subscriber, including congestion points affecting the subscriber's use of a specific service.

Action To display information about congestion points affecting a subscriber by subscriber IP address:

```
user@host> show acp congestion-point by-subscriber ip ip
```

Enter the subscriber's IP address.

To also display information about the congestion points for a specific service session associated with the subscriber:

```
user@host> show acp congestion-point by-subscriber ip ip service-name service-name
```

If the **service-name** is not provided, the command displays only edge congestion points accessed by the subscriber. If **service-name** is provided and any corresponding service sessions are active, the command also displays backbone congestion points accessed by those service sessions.

If the IP address matches multiple subscriber sessions, all matched subscriber sessions are displayed. If a service name matches multiple service sessions in a subscriber session, all the service sessions are displayed. Following is an example of the structure of the output:

User Session and Edge Congestion Points

```
USER SESSION
  User name                fred@default
  User DN
uniqueid=fred,ou=local,retailername=default,o=users,o=umc
  Current upstream bandwidth [kpbs]  0
  Current downstream bandwidth [kpbs] 0
  Upstream bandwidth usage [kpbs]    0
  Downstream bandwidth usage [kpbs]   0

  User information from tracking event
  NAS port ID                ip10.227.1.97
  NAS IP address             /0.0.0.0
  Virtual router name        acptest
  User IP address            N/A
  User session ID            HJo3uRLpxeNMkAAH
```

CONGESTION POINT

DN

```
interfacename=interface1,orderedcimkeys=device1,o=admissioncontrol,o=umc
```

```
Instance ID                instance1
Upstream current bandwidth  1000
Upstream bandwidth in use   500
Upstream provisioned bandwidth 1000
Upstream background bandwidth 250kbps
Downstream current bandwidth 1000
Downstream bandwidth in use  500
Downstream provisioned bandwidth 1000
Downstream background bandwidth 250kbps
```

CONGESTION POINT

DN

```
interfacename=interface1,orderedcimkeys=device1,o=admissioncontrol,o=umc
```

```
Instance ID                instance2
Upstream current bandwidth  1000
Upstream bandwidth in use   500
Upstream provisioned bandwidth 1000
Upstream background bandwidth 250kbps
Downstream current bandwidth 1000
Downstream bandwidth in use  500
Downstream provisioned bandwidth 1000
Downstream background bandwidth 250kbps
```

User Session and Edge Congestion Points

```

USER SESSION
  User name                fred@default
  User DN
uniqueid=fred,ou=local,retailername=default,o=users,o=umc
  Current upstream bandwidth [kpbs]  0
  Current downstream bandwidth [kpbs] 0
  Upstream bandwidth usage [kpbs]    500
  Downstream bandwidth usage [kpbs]   500

  User information from tracking event
  NAS port ID              ip10.227.1.96
  NAS IP address           /0.0.0.0
  Virtual router name      acptest
  User IP address          N/A
  User session ID          HJo3uRLpxeNMkAAG

SERVICE SESSION
  Accounting session ID      Video-Gold:fred:1300199748595:6
  Required upstream bandwidth [kbps] 500
  Required downstream bandwidth [kbps] 500
  State for redundancy       Started

```

CONGESTION POINT

DN

```

interfacename=interface1,orderedcimkeys=device1,o=admissioncontrol,o=umc
  Instance ID              instance1
  Upstream current bandwidth 1000
  Upstream bandwidth in use 500
  Upstream provisioned bandwidth 1000
  Upstream background bandwidth 250kbps
  Downstream current bandwidth 1000
  Downstream bandwidth in use 500
  Downstream provisioned bandwidth 1000
  Downstream background bandwidth 250kbps

```

CONGESTION POINT

DN

```

interfacename=interface1,orderedcimkeys=device1,o=admissioncontrol,o=umc
  Instance ID              instance2
  Upstream current bandwidth 1000
  Upstream bandwidth in use 500
  Upstream provisioned bandwidth 1000
  Upstream background bandwidth 250kbps
  Downstream current bandwidth 1000
  Downstream bandwidth in use 500
  Downstream provisioned bandwidth 1000
  Downstream background bandwidth 250kbps

```

Service Session and Backbone Congestion Points

```

SERVICE SESSION
  Accounting session ID      Video-Gold:fred:1300199748595:6
  Required upstream bandwidth [kbps] 500
  Required downstream bandwidth [kbps] 500
  State for redundancy       Started

CONGESTION POINT
DN
interfacename=interface2,orderedcimkeys=device1,o=admissioncontrol,o=umc
  Instance ID              instance1

```

```
Upstream current bandwidth      2000
Upstream bandwidth in use       500
Upstream provisioned bandwidth  2000
Upstream background bandwidth   500kbps
Downstream current bandwidth    2000
Downstream bandwidth in use     500
Downstream provisioned bandwidth 2000
Downstream background bandwidth 500kbps
```

CONGESTION POINT

DN

```
interfacename=interface2,orderedcimkeys=device1,o=admissioncontrol,o=umc
```

```
Instance ID      instance2
Upstream current bandwidth      2000
Upstream bandwidth in use       500
Upstream provisioned bandwidth  2000
Upstream background bandwidth   500kbps
Downstream current bandwidth    2000
Downstream bandwidth in use     500
Downstream provisioned bandwidth 2000
Downstream background bandwidth 500kbps
```

Related Documentation

- [Configuring SRC ACP to Manage the Backbone Network \(SRC CLI\) on page 285](#)
- [Configuring SRC ACP to Manage the Edge Network \(SRC CLI\) on page 281](#)
- [Viewing Congestion Point Information by Session ID and Associated Service Sessions \(SRC CLI\) on page 318](#)
- [Viewing Congestion Point Information by Login Name and Associated Service Sessions \(SRC CLI\) on page 321](#)

Viewing Congestion Point Information by Session ID and Associated Service Sessions (SRC CLI)

Purpose View edge and backbone congestion point information by subscriber session ID.

The command supports looking up congestion points affecting a specific subscriber, including congestion points affecting the subscriber's use of a specific service.

Action To display information about congestion points affecting a subscriber by subscriber session ID:

```
user@host> show acp congestion-point by-subscriber session-id session-id
```

Enter the subscriber session ID.

To also display information about the congestion points for a specific service session associated with the session ID:

```
user@host> show acp congestion-point by-subscriber session-id session-id service-name
service-name
```

If the **service-name** is not provided, the command displays only edge congestion points accessed by the subscriber session. If the **service-name** is provided and any corresponding service sessions are active, the command also displays backbone congestion points accessed by those service sessions.

If a service name matches multiple service sessions in a subscriber session, all the service sessions are displayed. Following is an example of the structure of the output:

User Session and Edge Congestion Points

```

USER SESSION
User name                fred@default
User DN
uniqueid=fred,ou=local,retailername=default,o=users,o=umc
Current upstream bandwidth [kpbs]  0
Current downstream bandwidth [kpbs] 0
Upstream bandwidth usage [kpbs]    0
Downstream bandwidth usage [kpbs]  0

```

User information from tracking event

```

NAS port ID      ip10.227.1.97
NAS IP address   /0.0.0.0
Virtual router name acptest
User IP address  N/A
User session ID  HJo3uRLpxeNMkAAH

```

CONGESTION POINT

DN

```

interfacename=interface1,orderedcimkeys=device1,o=admissioncontrol,o=umc
Instance ID          instance1
Upstream current bandwidth      1000
Upstream bandwidth in use      500
Upstream provisioned bandwidth 1000
Upstream background bandwidth  250kbps
Downstream current bandwidth    1000
Downstream bandwidth in use     500
Downstream provisioned bandwidth 1000
Downstream background bandwidth 250kbps

```

CONGESTION POINT

DN

```

interfacename=interface1,orderedcimkeys=device1,o=admissioncontrol,o=umc
Instance ID          instance2
Upstream current bandwidth      1000
Upstream bandwidth in use      500
Upstream provisioned bandwidth 1000
Upstream background bandwidth  250kbps
Downstream current bandwidth    1000
Downstream bandwidth in use     500
Downstream provisioned bandwidth 1000
Downstream background bandwidth 250kbps

```

User Session and Edge Congestion Points

```

USER SESSION
User name                fred@default
User DN
uniqueid=fred,ou=local,retailername=default,o=users,o=umc
Current upstream bandwidth [kpbs]  0
Current downstream bandwidth [kpbs] 0
Upstream bandwidth usage [kpbs]    500
Downstream bandwidth usage [kpbs]  500

```

User information from tracking event

```

NAS port ID      ip10.227.1.96
NAS IP address   /0.0.0.0
Virtual router name acptest
User IP address  N/A

```

User session ID HJo3uRLpxeNMkAAG

SERVICE SESSION

Accounting session ID Video-Gold:fred:1300199748595:6
Required upstream bandwidth [kbps] 500
Required downstream bandwidth [kbps] 500
State for redundancy Started

CONGESTION POINT

DN

interfacename=interface1,orderedcimkeys=device1,o=admissioncontrol,o=umc

Instance ID instance1
Upstream current bandwidth 1000
Upstream bandwidth in use 500
Upstream provisioned bandwidth 1000
Upstream background bandwidth 250kbps
Downstream current bandwidth 1000
Downstream bandwidth in use 500
Downstream provisioned bandwidth 1000
Downstream background bandwidth 250kbps

CONGESTION POINT

DN

interfacename=interface1,orderedcimkeys=device1,o=admissioncontrol,o=umc

Instance ID instance2
Upstream current bandwidth 1000
Upstream bandwidth in use 500
Upstream provisioned bandwidth 1000
Upstream background bandwidth 250kbps
Downstream current bandwidth 1000
Downstream bandwidth in use 500
Downstream provisioned bandwidth 1000
Downstream background bandwidth 250kbps

Service Session and Backbone Congestion Points

SERVICE SESSION

Accounting session ID Video-Gold:fred:1300199748595:6
Required upstream bandwidth [kbps] 500
Required downstream bandwidth [kbps] 500
State for redundancy Started

CONGESTION POINT

DN

interfacename=interface2,orderedcimkeys=device1,o=admissioncontrol,o=umc

Instance ID instance1
Upstream current bandwidth 2000
Upstream bandwidth in use 500
Upstream provisioned bandwidth 2000
Upstream background bandwidth 500kbps
Downstream current bandwidth 2000
Downstream bandwidth in use 500
Downstream provisioned bandwidth 2000
Downstream background bandwidth 500kbps

CONGESTION POINT

DN

interfacename=interface2,orderedcimkeys=device1,o=admissioncontrol,o=umc

Instance ID instance2
Upstream current bandwidth 2000
Upstream bandwidth in use 500
Upstream provisioned bandwidth 2000


```

Upstream background bandwidth    500kbps
Downstream current bandwidth     2000
Downstream bandwidth in use      500
Downstream provisioned bandwidth 2000
Downstream background bandwidth  500kbps

```

- Related Documentation**
- [Configuring SRC ACP to Manage the Backbone Network \(SRC CLI\) on page 285](#)
 - [Configuring SRC ACP to Manage the Edge Network \(SRC CLI\) on page 281](#)
 - [Viewing Congestion Point Information by Subscriber IP Address and Associated Service Sessions \(SRC CLI\) on page 315](#)
 - [Viewing Congestion Point Information by Login Name and Associated Service Sessions \(SRC CLI\) on page 321](#)

Viewing Congestion Point Information by Login Name and Associated Service Sessions (SRC CLI)

Purpose View edge and backbone congestion point information by subscriber login name.

The command supports looking up congestion points affecting a subscriber, including congestion points affecting the subscriber's use of a specific service.

Action To display information about congestion points affecting a subscriber by subscriber login name:

```
user@host> show acp congestion-point by-subscriber login login
```

To also display information about the congestion points for a specific service session associated with the subscriber login:

```
user@host> show acp congestion-point by-subscriber login login service-name service-name
```

If the **service-name** is not provided, the command displays only edge congestion points accessed by the subscriber. If the **service-name** is provided and any corresponding service sessions are active, the command also displays backbone congestion points accessed by those service sessions.

If the login name matches multiple subscriber sessions, all matched subscriber sessions are displayed. If a service name matches multiple service sessions in a subscriber session, all the service sessions are displayed. Following is an example of the structure of the output:

User Session and Edge Congestion Points

```

USER SESSION
User name                fred@default
User DN
uniqueid=fred,ou=local,retailername=default,o=users,o=umc
Current upstream bandwidth [kpbs] 0
Current downstream bandwidth [kpbs] 0
Upstream bandwidth usage [kpbs] 0
Downstream bandwidth usage [kpbs] 0

```

```

User information from tracking event
NAS port ID      ip10.227.1.97
NAS IP address   /0.0.0.0
Virtual router name acptest

```

```
User IP address      N/A
User session ID     HJo3uRLpxeNMkAAH
```

CONGESTION POINT

DN

```
interfacename=interface1,orderedcimkeys=device1,o=admissioncontrol,o=umc
```

```
Instance ID          instance1
Upstream current bandwidth 1000
Upstream bandwidth in use 500
Upstream provisioned bandwidth 1000
Upstream background bandwidth 250kbps
Downstream current bandwidth 1000
Downstream bandwidth in use 500
Downstream provisioned bandwidth 1000
Downstream background bandwidth 250kbps
```

CONGESTION POINT

DN

```
interfacename=interface1,orderedcimkeys=device1,o=admissioncontrol,o=umc
```

```
Instance ID          instance2
Upstream current bandwidth 1000
Upstream bandwidth in use 500
Upstream provisioned bandwidth 1000
Upstream background bandwidth 250kbps
Downstream current bandwidth 1000
Downstream bandwidth in use 500
Downstream provisioned bandwidth 1000
Downstream background bandwidth 250kbps
```

User Session and Edge Congestion Points

USER SESSION

```
User name              fred@default
User DN
```

```
uniqueid=fred,ou=local,retailername=default,o=users,o=umc
```

```
Current upstream bandwidth [kpbs] 0
Current downstream bandwidth [kpbs] 0
Upstream bandwidth usage [kpbs] 500
Downstream bandwidth usage [kpbs] 500
```

User information from tracking event

```
NAS port ID           ip10.227.1.96
NAS IP address         /0.0.0.0
Virtual router name    acptest
User IP address        N/A
User session ID        HJo3uRLpxeNMkAAG
```

SERVICE SESSION

```
Accounting session ID      Video-Gold:fred:1300199748595:6
Required upstream bandwidth [kpbs] 500
Required downstream bandwidth [kpbs] 500
State for redundancy       Started
```

CONGESTION POINT

DN

```
interfacename=interface1,orderedcimkeys=device1,o=admissioncontrol,o=umc
```

```
Instance ID          instance1
Upstream current bandwidth 1000
Upstream bandwidth in use 500
Upstream provisioned bandwidth 1000
Upstream background bandwidth 250kbps
Downstream current bandwidth 1000
```

```

Downstream bandwidth in use      500
Downstream provisioned bandwidth 1000
Downstream background bandwidth 250kbps

```

CONGESTION POINT

DN

```
interfacename=interface1,orderedcimkeys=device1,o=admissioncontrol,o=umc
```

```

Instance ID           instance2
Upstream current bandwidth      1000
Upstream bandwidth in use      500
Upstream provisioned bandwidth 1000
Upstream background bandwidth 250kbps
Downstream current bandwidth    1000
Downstream bandwidth in use     500
Downstream provisioned bandwidth 1000
Downstream background bandwidth 250kbps

```

Service Session and Backbone Congestion Points

SERVICE SESSION

```

Accounting session ID           Video-Gold:fred:1300199748595:6
Required upstream bandwidth [kbps] 500
Required downstream bandwidth [kbps] 500
State for redundancy            Started

```

CONGESTION POINT

DN

```
interfacename=interface2,orderedcimkeys=device1,o=admissioncontrol,o=umc
```

```

Instance ID           instance1
Upstream current bandwidth      2000
Upstream bandwidth in use      500
Upstream provisioned bandwidth 2000
Upstream background bandwidth 500kbps
Downstream current bandwidth    2000
Downstream bandwidth in use     500
Downstream provisioned bandwidth 2000
Downstream background bandwidth 500kbps

```

CONGESTION POINT

DN

```
interfacename=interface2,orderedcimkeys=device1,o=admissioncontrol,o=umc
```

```

Instance ID           instance2
Upstream current bandwidth      2000
Upstream bandwidth in use      500
Upstream provisioned bandwidth 2000
Upstream background bandwidth 500kbps
Downstream current bandwidth    2000
Downstream bandwidth in use     500
Downstream provisioned bandwidth 2000
Downstream background bandwidth 500kbps

```

Related Documentation

- [Configuring SRC ACP to Manage the Backbone Network \(SRC CLI\) on page 285](#)
- [Configuring SRC ACP to Manage the Edge Network \(SRC CLI\) on page 281](#)
- [Viewing Congestion Point Information by Subscriber IP Address and Associated Service Sessions \(SRC CLI\) on page 315](#)
- [Viewing Congestion Point Information by Session ID and Associated Service Sessions \(SRC CLI\) on page 318](#)

Viewing Action Congestion Point Information by Service (SRC CLI)

Purpose	View action congestion point information by service.
Action	<p>To display information about services that SRC ACP manages in the backbone network:</p> <pre>user@host> show acp backbone service</pre> <p>To display information about specific backbone services used to generate congestion points:</p> <pre>user@host> show acp backbone service service-name service-name</pre> <p>Enter a partial service name to list all matching backbone services.</p> <p>To display information about the backbone services from a specific virtual router:</p> <pre>user@host> show acp backbone service virtual-router-name virtual-router-name</pre> <p>To display backbone service attributes:</p> <pre>user@host> show acp backbone service brief</pre> <p>By default, information about the backbone service attributes, service sessions, and associated congestion points is displayed.</p>
Related Documentation	<ul style="list-style-type: none">• Configuring SRC ACP to Manage the Backbone Network (SRC CLI) on page 285• Viewing Information about Action Congestion Points in a Backbone Network by Service (C-Web Interface) on page 336• Viewing Action Congestion Point Information by Congestion Point (SRC CLI) on page 324

Viewing Action Congestion Point Information by Congestion Point (SRC CLI)

Purpose	View action congestion point information by congestion point.
Action	<p>To display information about backbone congestion points by service:</p> <pre>user@host> show acp backbone congestion-point congestion-point-expression</pre> <p>To display information about specific backbone services used to generate congestion points:</p> <pre>user@host> show acp backbone congestion-point congestion-point-expression service-name service-name</pre> <p>Enter a partial service name to list all matching backbone services.</p> <p>To display information about the backbone services from a specific virtual router:</p> <pre>user@host> show acp backbone congestion-point congestion-point-expression virtual-router-name virtual-router-name</pre> <p>Enter a virtual router name to list backbone services from a particular virtual router.</p>

To display information about the backbone services from a specific interface:

```
user@host> show acp backbone congestion-point congestion-point-expression interface-name
interface-name
```

Enter an interface name to list backbone services from a particular interface.

To display information about the backbone services for a specific interface description:

```
user@host> show acp backbone congestion-point congestion-point-expression
interface-description interface-description
```

Enter an interface description to list backbone services for a particular description.

To display information about the backbone services from a specific interface alias:

```
user@host> show acp backbone congestion-point congestion-point-expression interface-alias
interface-alias
```

Enter an interface alias to list backbone services from a particular alias.

To display information about the backbone services for a specific NAS port ID:

```
user@host> show acp backbone congestion-point congestion-point-expression nasPort-id
nasPort-id
```

Enter a NAS port ID to list backbone services from a particular ID.

To display congestion point DNs:

```
user@host> show acp backbone congestion-point congestion-point-expression brief
```

By default, information about the congestion point attributes and congestion point bandwidth is displayed.

Related Documentation

- [Configuring SRC ACP to Manage the Backbone Network \(SRC CLI\) on page 285](#)
- [Viewing Backbone Congestion Point Information by DN \(SRC CLI\) on page 314](#)
- [Viewing Backbone Congestion Point Information by Service \(SRC CLI\) on page 315](#)
- [Viewing Action Congestion Point Information by Service \(SRC CLI\) on page 324](#)

Viewing Information About Subscribers Obtained from External Applications (SRC CLI)

Purpose View information about subscribers obtained from external applications.

Action To display information about subscribers added through an external application:

```
user@host> show acp remote-update subscriber
```

To display information about subscribers connected from a specific device:

```
user@host> show acp remote-update subscriber device-name device-name
```

Enter a device name to list subscribers connected from a particular device.

To display information about specific subscribers connected from a specific interface:

```
user@host> show acp remote-update subscriber nas-port-id nas-port-id
```

Enter the NAS port ID of interface to list all matching subscribers connected from a particular interface.

To display information about specific subscribers connected from a specific NAS IP address:

```
user@host> show acp remote-update subscriber nas-ip nas-ip
```

Enter the NAS IP address of the device to list all matching subscribers connected from a particular device.

To display information about specific subscribers connected from a specific subscriber IP address:

```
user@host> show acp remote-update subscriber subscriber-ip subscriber-ip
```

Enter the subscriber IP address to list all matching subscribers connected from a particular address.

To display information about the subscribers from a specific phone number:

```
user@host> show acp remote-update subscriber phone phone
```

Enter a phone number to list subscribers from a particular phone number.

To display subscriber attributes:

```
user@host> show acp remote-update subscriber brief
```

By default, information about the subscriber attributes, service sessions, and associated congestion points is displayed.

- Related Documentation**
- [Viewing Information About Subscribers Obtained from External Applications \(C-Web Interface\) on page 339](#)
 - [Viewing Congestion Point Information by DN \(SRC CLI\) on page 326](#)
 - [Viewing Congestion Point Information by Name \(SRC CLI\) on page 327](#)

Viewing Congestion Point Information by DN (SRC CLI)

Purpose View congestion point information by DN.

Action To display information about congestion points added through an external application by DN:

```
user@host> show acp remote-update congestion-point dn
```

To display information about specific congestion points by DN:

```
user@host> show acp remote-update congestion-point dn congestion-point-dn  
congestion-point-dn
```

Enter a partial congestion point DN to list all matching congestion points.

To display congestion point DNs:

```
user@host> show acp remote-update congestion-point dn brief
```

By default, information about the congestion point attributes and congestion point bandwidth usage is displayed.

- Related Documentation**
- [Viewing Information About Congestion Points from an External Application by DN \(C-Web Interface\) on page 341](#)
 - [Viewing Information About Subscribers Obtained from External Applications \(SRC CLI\) on page 325](#)
 - [Viewing Congestion Point Information by Name \(SRC CLI\) on page 327](#)

Viewing Congestion Point Information by Name (SRC CLI)

Purpose View congestion point information by name.

Action To display information about congestion points added through an external application by interface name:

```
user@host> show acp remote-update congestion-point name
```

To display information about congestion points connected from a specific device:

```
user@host> show acp remote-update congestion-point name device-name device-name
```

Enter a device name to list congestion points connected from a particular device.

To display information about specific subscribers connected from a specific interface:

```
user@host> show acp remote-update congestion-point name interface-name interface-name
```

Enter the interface name to list all matching congestion points connected from a particular interface.

To display congestion point DN:

```
user@host> show acp remote-update congestion-point name brief
```

By default, information about the congestion point attributes and congestion point bandwidth usage is displayed.

- Related Documentation**
- [Viewing Information About Congestion Points from an External Application by Interface Name \(C-Web Interface\) on page 341](#)
 - [Viewing Information About Subscribers Obtained from External Applications \(SRC CLI\) on page 325](#)
 - [Viewing Congestion Point Information by DN \(SRC CLI\) on page 326](#)

Viewing SNMP Information for Devices (SRC CLI)

Purpose View SNMP information for devices.

Action To display statistics for SNMP information about each device:

```
user@host> show acp statistics device
```

To display statistics for SNMP information about specific devices:

```
user@host> show acp statistics device filter filter
```

Enter a partial device name to list information for all matching devices.

- Related Documentation**
- [Viewing SNMP Information for the Directory \(SRC CLI\) on page 328](#)
 - [Viewing SNMP Information for SRC ACP \(SRC CLI\) on page 328](#)

Viewing SNMP Information for the Directory (SRC CLI)

Purpose View SNMP information for the directory.

Action To display statistics for directory SNMP information:

```
user@host> show acp statistics directory
```

- Related Documentation**
- [Viewing SNMP Information for Devices \(SRC CLI\) on page 327](#)
 - [Viewing SNMP Information for SRC ACP \(SRC CLI\) on page 328](#)

Viewing SNMP Information for SRC ACP (SRC CLI)

Purpose View SNMP information for SRC ACP.

Action To display statistics for SRC ACP SNMP information:

```
user@host> show acp statistics general
```

- Related Documentation**
- [Viewing SNMP Information for Devices \(SRC CLI\) on page 327](#)
 - [Viewing SNMP Information for the Directory \(SRC CLI\) on page 328](#)

CHAPTER 23

Monitoring Admission Control (C-Web Interface)

- Viewing Information About Subscriber Sessions in the Edge Network (C-Web Interface) on page 329
- Viewing Information About Edge Congestion Points by DN (C-Web Interface) on page 330
- Viewing Information About Edge Congestion Points by Subscriber Session (C-Web Interface) on page 331
- Viewing Information About Services in a Backbone Network (C-Web Interface) on page 332
- Viewing Information About Congestion Points in a Backbone Network by Expression (C-Web Interface) on page 334
- Viewing Information About Congestion Points in a Backbone Network by DN (C-Web Interface) on page 335
- Viewing Information about Action Congestion Points in a Backbone Network by Service (C-Web Interface) on page 336
- Viewing Information about Action Congestion Points in a Backbone Network by Expression (C-Web Interface) on page 338
- Viewing Information About Subscribers Obtained from External Applications (C-Web Interface) on page 339
- Viewing Information About Congestion Points from an External Application by DN (C-Web Interface) on page 341
- Viewing Information About Congestion Points from an External Application by Interface Name (C-Web Interface) on page 341
- Viewing Statistics for the SRC ACP Configuration (C-Web Interface) on page 342

Viewing Information About Subscriber Sessions in the Edge Network (C-Web Interface)

Purpose View information about subscriber sessions in the edge network with the C-Web interface.

Action To view information about subscriber sessions:

1. Click **ACP>Edge>Subscriber**.

The Edge/Subscriber pane appears.

Monitor	Configure	Diagnose	Manage
ACP	ACP		
CLI	Edge / Subscriber		
Component			
Date			
Disk			
Interfaces...			
Iptables...			
JPS			
NIC			
NTP			
Redirect Server			
Route...			
SAE			
Security			
System			

Session Id

Slot

Style

Virtual Router Name

OK Reset

Copyright © 2007, Juniper Networks, Inc. All Rights Reserved. Trademark Notice. Privacy. Juniper Your Net.

2. In the Session ID box, enter a full or partial session ID name to display information about one or more specific sessions, or leave this field empty to display information about all sessions.
3. In the Slot box, enter the number of the slot for which you want to display subscriber session information.
4. Select an output style from the Style list.
5. In the Virtual Router Name box, enter a virtual router name to display information about a specific virtual router, or leave the box empty to display information about all virtual routers.
6. Click **OK**.

The Edge/Subscriber pane displays a list of current subscriber sessions.

Related Documentation

- [Configuring SRC ACP to Manage the Edge Network \(C-Web Interface\)](#)
- [Viewing Information About Subscriber Sessions in the Edge Network \(SRC CLI\) on page 311](#)
- [Viewing Information About Edge Congestion Points by Subscriber Session \(C-Web Interface\) on page 331](#)
- [Viewing Information About Edge Congestion Points by DN \(C-Web Interface\) on page 330](#)

Viewing Information About Edge Congestion Points by DN (C-Web Interface)

Purpose View information about edge congestion points by DN.

Action To view information about edge congestion points:

1. Click **ACP>Edge>Congestion Point>DN**.

The Edge/Congestion Point/DN pane appears.

The screenshot shows the Juniper C-Web Interface with the 'ACP' menu expanded and 'Edge / Congestion Point / DN' selected. The interface includes a top navigation bar with 'Monitor', 'Configure', 'Diagnose', and 'Manage' tabs. A sidebar on the left lists various configuration categories. The main content area displays the 'Edge / Congestion Point / DN' configuration pane. This pane contains four input fields: 'Congestion Point Dn' (with a description and default value of 'No value'), 'Slot' (with a description and default value of '0'), 'Style' (a dropdown menu with choices 'brief' and 'detail'), and 'Virtual Router Name' (with a description and default value of 'No value'). Below these fields are 'OK' and 'Reset' buttons. The footer of the interface shows the copyright notice for Juniper Networks, Inc. and the 'Juniper Your Net.' logo.

2. In the Congestion Point DN box, enter a congestion point DN, or leave the box blank to view information for all DNs.
3. In the Slot box, enter the number of the slot for which you want to display congestion point information.
4. Select an output style from the Style list.
5. In the Virtual Router Name box, enter a virtual router name to display information about a specific virtual router, or leave the box empty to display information about all virtual routers.
6. Click **OK**.

The Edge/Congestion Point/DN pane displays a list of congestion points.

Related Documentation

- [Configuring SRC ACP to Manage the Edge Network \(C-Web Interface\)](#)
- [Viewing Information About Edge Congestion Points by Subscriber Session \(C-Web Interface\) on page 331](#)
- [Viewing Information About Subscriber Sessions in the Edge Network \(C-Web Interface\) on page 329](#)

Viewing Information About Edge Congestion Points by Subscriber Session (C-Web Interface)

Purpose View information about edge congestion points by subscriber session.

Action To view information about edge congestion points:

1. Click **ACP>Edge>Congestion Point>Subscriber Session ID**.

The Edge/Congestion Point/Subscriber Session ID pane appears.

Monitor	Configure	Diagnose	Manage
ACP	Edge / Congestion Point / Subscriber Session ID		
CLI			
Component			
Date			
Disk			
Interfaces...			
Iptables...			
JPS			
NIC			
NTP			
Redirect Server			
Route...			
SAE			
Security			
System			

Session Id:

Slot:

Style:

Virtual Router Name:

OK Reset

Copyright © 2007, Juniper Networks, Inc. All Rights Reserved. Trademark Notice. Privacy. Juniper Your Net.

2. In the Session ID box, enter a full or partial session ID name to display information about one or more specific sessions, or leave the box empty to display information about all sessions.
3. In the Slot box, enter the number of the slot for which you want to display congestion point information.
4. Select an output style from the Style list.
5. In the Virtual Router Name box, enter a virtual router name to display information about a specific virtual router, or leave the box empty to display information about all virtual routers.
6. Click **OK**.

The Edge/Congestion Point/Subscriber Session ID pane displays a list of congestion points.

Related Documentation

- [Configuring SRC ACP to Manage the Edge Network \(C-Web Interface\)](#)
- [Viewing Information About Edge Congestion Points by DN \(C-Web Interface\) on page 330](#)
- [Viewing Information About Subscriber Sessions in the Edge Network \(C-Web Interface\) on page 329](#)

Viewing Information About Services in a Backbone Network (C-Web Interface)

Purpose View information about services in a backbone network with the C-Web interface.

Action To view information about services in a backbone network:

1. Click **ACP>Backbone>Service**.

The Backbone/Service pane appears.

Field	Description	Value	Default
Interface Alias	Interface alias used by backbone service to generate congestion points.	Interface alias	No value
Interface Description	Description of interface used by backbone service to generate congestion points.	Interface description	No value
Interface Name	Name of interface related to congestion points.	Interface name	No value
Nas Port Id	Interface NAS port ID used by backbone service to generate congestion points.	NAS port ID	No value
Service Name	Name of service used by backbone service to generate congestion points.	Service name	No value
Slot	Number of the slot for which you want to configure values. Currently, the chassis has only one slot. The valid value is 0.		0
Style	Output style. Choices: brief: Display backbone service attributes		detail
Virtual Router Name	Name of virtual router from which to list backbone services.	Virtual router name	No value

2. In the Interface Alias box, enter the interface alias used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.
3. In the Interface Description box, enter the interface description used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.
4. In the Interface Name box, enter the name of an interface to display information about one interface, or leave the box empty to display information about all interfaces.
5. In the NAS Port ID box, enter the NAS port ID used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.
6. In the Service Name box, enter the name of a service to display information about one service, or leave the box empty to display information about all services.
7. In the Slot box, enter the number of the slot for which you want to display congestion point information.
8. Select an output style from the Style list.
9. In the Virtual Router Name box, enter a virtual router name to display information about a specific virtual router, or leave the box empty to display information about all virtual routers.
10. Click **OK**.

The Backbone/Service pane displays a list of services.

For more information about viewing service information for action congestion points, see [“Viewing Information about Action Congestion Points in a Backbone Network by Service \(C-Web Interface\)”](#) on page 336.

- Related Documentation**
- [Configuring SRC ACP to Manage the Backbone Network \(SRC CLI\) on page 285](#)
 - [Viewing Information About Services in the Backbone Network \(SRC CLI\) on page 313](#)
 - [Viewing Information About Congestion Points in a Backbone Network by Expression \(C-Web Interface\) on page 334](#)
 - [Viewing Information About Congestion Points in a Backbone Network by DN \(C-Web Interface\) on page 335](#)

Viewing Information About Congestion Points in a Backbone Network by Expression (C-Web Interface)

Purpose View information about congestion points in a backbone network by expression.

Action To view information about congestion points by expression:

1. Click **ACP>Backbone>Congestion Point>Congestion Point Expression**.

The Backbone/Congestion Point/Congestion Point Expression pane appears.

Field	Description	Value	Default
Interface Alias	Interface alias used by backbone service to generate congestion points.	Interface alias	No value
Interface Description	Description of interface used by backbone service to generate congestion points.	Interface description	No value
Interface Name	Name of interface related to congestion points.	Interface name	No value
Nas Port Id	Interface NAS port ID used by backbone service to generate congestion points.	NAS port ID	No value
Service Name	Name of service used by backbone service to generate congestion points.	Service name	No value
Slot	Number of the slot for which you want to configure values.	Currently, the chassis has only one slot. The valid value is 0.	0
Style	Output style.	Choices: brief, detail	detail
Virtual Router Name	Name of virtual router from which to list congestion points.	Virtual router name	No value

2. In the Interface Alias box, enter the interface alias used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.
3. In the Interface Description box, enter the interface description used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.
4. In the Interface Name box, enter the name of an interface to display information about one interface, or leave the box empty to display information about all interfaces.
5. In the NAS Port ID box, enter the NAS port ID used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.
6. In the Service Name box, enter the name of a service to display information about one service, or leave the box empty to display information about all services.

7. In the Slot box, enter the number of the slot for which you want to display congestion point information.
8. Select an output style from the Style list.
9. In the Virtual Router Name box, enter a virtual router name to display information about a specific virtual router, or leave the box empty to display information about all virtual routers.
10. Click **OK**.

The Backbone/Congestion Point/Congestion Point Expression pane displays a list of congestion points.

For more information about viewing information for action congestion points by expression, see [“Viewing Information about Action Congestion Points in a Backbone Network by Expression \(C-Web Interface\)”](#) on page 338.

Related Documentation

- [Configuring Congestion Points in the Directory](#)
- [Viewing Information About Services in a Backbone Network \(C-Web Interface\)](#) on page 332
- [Viewing Information About Congestion Points in a Backbone Network by DN \(C-Web Interface\)](#) on page 335
- [Viewing Information about Action Congestion Points in a Backbone Network by Service \(C-Web Interface\)](#) on page 336

Viewing Information About Congestion Points in a Backbone Network by DN (C-Web Interface)

Purpose View information about congestion points in a backbone network by DN.

Action To view information about congestion points by DN:

1. Click **ACP>Backbone>Congestion Point>DN**.

The Backbone/Congestion Point/DN pane appears.

Field	Description	Default Value
Congestion Point Dn	DN of congestion point for which you want to list all matching congestion points.	All or part of the congestion point DN.
Slot	Number of the slot for which you want to configure values.	Currently, the chassis has only one slot. The valid value is 0.
Style	Output style.	brief: Display congestion point DN detail: Display congestion point detail
Virtual Router Name	Name of virtual router from which to list congestion points.	Virtual router name

2. In the Congestion Point DN box, enter a full or partial congestion point name to display information about one or more specific congestion points, or leave the box empty to display information about all congestion points.
3. In the Slot box, enter the number of the slot for which you want to display congestion point information.
4. Select an output style from the Style list.
5. In the Virtual Router Name box, enter a virtual router name to display information about a specific virtual router, or leave the box empty to display information about all virtual routers.
6. Click **OK**.

The Backbone/Congestion Point/DN pane displays a list of congestion points.

Related Documentation

- [Configuring Congestion Points in the Directory](#)
- [Viewing Information About Services in the Backbone Network \(SRC CLI\) on page 313](#)
- [Viewing Information About Congestion Points in a Backbone Network by Expression \(C-Web Interface\) on page 334](#)
- [Viewing Information about Action Congestion Points in a Backbone Network by Service \(C-Web Interface\) on page 336](#)

Viewing Information about Action Congestion Points in a Backbone Network by Service (C-Web Interface)

Purpose View information about action congestion points in a backbone network by service.

Action To view information about action congestion points in a backbone network by service:

1. Click **ACP>Backbone>Service**.

The Backbone/Service pane appears.

ACP		
Backbone / Service		
Interface Alias	<input type="text"/>	Interface alias used by backbone service to generate congestion points. <i>Value:</i> Interface alias <i>Default:</i> No value
Interface Description	<input type="text"/>	Description of interface used by backbone service to generate congestion points. <i>Value:</i> Interface description <i>Default:</i> No value
Interface Name	<input type="text"/>	Name of interface related to congestion points. <i>Value:</i> Interface name <i>Default:</i> No value
Nas Port Id	<input type="text"/>	Interface NAS port ID used by backbone service to generate congestion points. <i>Value:</i> NAS port ID <i>Default:</i> No value
Service Name	<input type="text"/>	Name of service used by backbone service to generate congestion points. <i>Value:</i> Service name <i>Default:</i> No value
Slot	<input type="text"/>	Number of the slot for which you want to configure values. <i>Value:</i> Currently, the chassis has only one slot. The valid value is 0. <i>Default value:</i> 0
Style	<input type="text" value="detail"/>	Output style. <i>Choices:</i> brief: Display backbone service attributes <i>Default value:</i> detail
Virtual Router Name	<input type="text"/>	Name of virtual router from which to list backbone services. <i>Value:</i> Virtual router name <i>Default:</i> No value

OK Reset

2. In the Interface Alias box, enter the interface alias used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.
3. In the Interface Description box, enter the interface description used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.
4. In the Interface Name box, enter the name of an interface to display information about one interface related to congestion points, or leave the box empty to display information about all interfaces.
5. In the NAS Port ID box, enter the NAS port ID used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.
6. In the Service Name box, enter the name of a service to display information about one service, or leave the box empty to display information about all services.
7. In the Slot box, enter the number of the slot for which you want to display congestion point information.
8. Select an output style from the Style list.
9. In the Virtual Router Name box, enter a virtual router name to display information about a specific virtual router, or leave the box empty to display information about all virtual routers.
10. Click **OK**.

The Backbone/Service pane displays a list of congestion points.

Related Documentation

- [Configuring Action Congestion Points](#)
- [Viewing Information About Services in the Backbone Network \(SRC CLI\) on page 313](#)

- Viewing Information About Congestion Points in a Backbone Network by Expression (C-Web Interface) on page 334
- Viewing Information About Congestion Points in a Backbone Network by DN (C-Web Interface) on page 335

Viewing Information about Action Congestion Points in a Backbone Network by Expression (C-Web Interface)

Purpose View information about action congestion points in a backbone network by expression.

Action To view information about action congestion points in a backbone network by expression:

1. Click **ACP>Backbone>Congestion Point>Congestion Point Expression**.

The Backbone/Congestion Point/Congestion Point Expression pane appears.

Field	Description	Value	Default
Interface Alias	Interface alias used by backbone service to generate congestion points.	Interface alias	No value
Interface Description	Description of interface used by backbone service to generate congestion points.	Interface description	No value
Interface Name	Name of interface related to congestion points.	Interface name	No value
Nas Port Id	Interface NAS port ID used by backbone service to generate congestion points.	NAS port ID	No value
Service Name	Name of service used by backbone service to generate congestion points.	Service name	No value
Slot	Number of the slot for which you want to configure values.	Currently, the chassis has only one slot. The valid value is 0.	0
Style	Output style.	Choices: brief; Display congestion point attributes	detail
Virtual Router Name	Name of virtual router from which to list congestion points.	Virtual router name	No value

2. In the Interface Alias box, enter the interface alias used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.
3. In the Interface Description box, enter the interface description used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.
4. In the Interface Name box, enter the name of an interface to display information about one interface related to congestion points, or leave the box empty to display information about all interfaces.
5. In the NAS Port ID box, enter the NAS port ID used by the backbone service to generate congestion points, or leave the box empty to display information about all interfaces.
6. In the Service Name box, enter the name of a service to display information about one service, or leave the box empty to display information about all services.

7. In the Slot box, enter the number of the slot for which you want to display congestion point information.
8. Select an output style from the Style list.
9. In the Virtual Router Name box, enter a virtual router name to display information about a specific virtual router, or leave the box empty to display information about all virtual routers.
10. Click **OK**.

The Backbone/Congestion Point/Congestion Point Expression pane displays a list of congestion points.

**Related
Documentation**

- *Configuring Action Congestion Points*
- [Viewing Information about Action Congestion Points in a Backbone Network by Service \(C-Web Interface\) on page 336](#)
- [Viewing Information About Congestion Points in a Backbone Network by Expression \(C-Web Interface\) on page 334](#)
- [Viewing Information About Congestion Points in a Backbone Network by DN \(C-Web Interface\) on page 335](#)

Viewing Information About Subscribers Obtained from External Applications (C-Web Interface)

Purpose View information about subscribers obtained from external applications with the C-Web interface.

Action To view information about subscribers obtained from external applications:

1. Click **ACP>Remote Update>Subscriber**.

The Remote Update/Subscriber pane appears.

2. In the Device Name box, enter the device name of the congestion point, or leave the box blank to display information about all devices.
3. In the NAS IP box, enter the NAS IP address of the device connected to the subscriber, or leave the box empty to display information about all subscribers.
4. In the NAS Port ID box, enter the NAS port ID connected to the subscriber, or leave the box empty to display information about all subscribers.
5. In the Phone box, enter the phone number of the subscriber, or leave the box blank to display information about all subscribers.
6. In the Slot box, enter the number of the slot for which you want to display external subscriber information.
7. Select an output style from the Style list.
8. In the Subscriber IP box, enter the subscriber IP address, or leave the box empty to display information about all subscribers.
9. Click **OK**.

The Remote Update/Subscriber pane displays the congestion points.

Related Documentation

- [Viewing Information About Subscribers Obtained from External Applications \(SRC CLI\) on page 325](#)
- [Viewing Information About Congestion Points from an External Application by DN \(C-Web Interface\) on page 341](#)
- [Viewing Information About Congestion Points from an External Application by Interface Name \(C-Web Interface\) on page 341](#)

Viewing Information About Congestion Points from an External Application by DN (C-Web Interface)

Purpose View information about congestion points from an external application by DN.

Action To view information about congestion points added through an external application by DN:

1. Click **ACP>Remote Update>Congestion Point>DN**.

The Remote Update/Congestion Point/DN pane appears.

The screenshot shows the Juniper C-Web Interface with the 'Monitor' tab selected. The left sidebar contains a tree view with 'ACP' expanded. The main content area is titled 'Remote Update / Congestion Point / DN'. It contains three input fields: 'Congestion Point Dn', 'Slot', and 'Style'. Each field has a help text box to its right. The 'Congestion Point Dn' field has a help text: 'DN of congestion point for which you want to list all matching congestion points. Value: All or part of the congestion point DN. Default: No value'. The 'Slot' field has a help text: 'Number of the slot for which you want to configure values. Value: Currently, the chassis has only one slot. The valid value is 0. Default value: 0'. The 'Style' field has a help text: 'Output style. Choices: brief; Display congestion point DN. Default value: detail'. At the bottom of the pane are 'OK' and 'Reset' buttons.

2. In the Congestion Point DN box, enter the DN of the congestion point, or leave the box blank to display information about all devices.
3. In the Slot box, enter the number of the slot for which you want to display congestion point information.
4. Select an output style from the Style list.
5. Click **OK**.

The Remote Update/Congestion Point/DN pane displays the congestion points.

Related Documentation

- [Viewing Congestion Point Information by DN \(SRC CLI\) on page 326](#)
- [Viewing Information About Subscribers Obtained from External Applications \(C-Web Interface\) on page 339](#)
- [Viewing Information About Congestion Points from an External Application by Interface Name \(C-Web Interface\) on page 341](#)

Viewing Information About Congestion Points from an External Application by Interface Name (C-Web Interface)

Purpose View information about congestion points from an external application by interface name.

Action 1. Click **ACP>Remote Update>Congestion Point>Name**.

The Remote Update/Congestion Point/Name pane appears.

2. In the Device Name box, enter the device name of the congestion point, or leave the box blank to display information about all devices.
3. In the Interface Name box, enter the interface name of the congestion point, or leave the box blank to display information about all interfaces.
4. In the Slot box, enter the number of the slot for which you want to display congestion point information.
5. Select an output style from the Style list.
6. Click **OK**.

The Remote Update/Congestion Point/Name pane displays the congestion points.

- Related Documentation**
- [Viewing Information About Subscribers Obtained from External Applications \(C-Web Interface\)](#) on page 339
 - [Viewing Information About Congestion Points from an External Application by DN \(C-Web Interface\)](#) on page 341
 - [Viewing Congestion Point Information by Name \(SRC CLI\)](#) on page 327

Viewing Statistics for the SRC ACP Configuration (C-Web Interface)

- [Viewing General Statistics for SRC ACP \(C-Web Interface\)](#) on page 342
- [Viewing Statistics for the SRC ACP Directory \(C-Web Interface\)](#) on page 343
- [Viewing Device Statistics for SRC ACP \(C-Web Interface\)](#) on page 344

Viewing General Statistics for SRC ACP (C-Web Interface)

Purpose View general statistics for SRC ACP.

Action To view general statistics for SRC ACP:

1. Click **ACP>Statistics>General**.

The Statistics/General pane appears.

Monitor Configure Diagnose Manage Logged in as: admin Refresh Preferences About Logout

ACP

CLI

Component

Date

Disk

Interfaces...

Iptables...

JPS

NIC

NTP

Redirect Server

Route...

SAE

Security

System

ACP

Statistics / General

Slot

Number of the slot for which you want to configure values.
Value: Currently, the chassis has only one slot. The valid value is 0.
Default value: 0

OK Reset

Copyright © 2007, Juniper Networks, Inc. All Rights Reserved. Trademark Notice. Privacy. Juniper Your Net.

2. In the Slot box, enter the number of the slot for which you want to display general statistics.
3. Click **OK**.

The Statistics/General pane displays general SRC ACP statistics.

Viewing Statistics for the SRC ACP Directory (C-Web Interface)

Purpose View statistics for the SRC ACP directory.

Action To view statistics about the SRC ACP directory:

1. Click **ACP>Statistics>Directory**.

The Statistics/Directory pane appears.

Monitor Configure Diagnose Manage Logged in as: admin Refresh Preferences About Logout

ACP

CLI

Component

Date

Disk

Interfaces...

Iptables...

JPS

NIC

NTP

Redirect Server

Route...

SAE

Security

System

ACP

Statistics / Directory

Slot

Number of the slot for which you want to configure values.
Value: Currently, the chassis has only one slot. The valid value is 0.
Default value: 0

OK Reset

Copyright © 2007, Juniper Networks, Inc. All Rights Reserved. Trademark Notice. Privacy. Juniper Your Net.

2. In the Slot box, enter the number of the slot for which you want to display directory statistics.
3. Click **OK**.

The Statistics/Directory pane displays statistics for the SRC ACP directory.

Viewing Device Statistics for SRC ACP (C-Web Interface)

Purpose View device statistics for SRC ACP.

Action To view device statistics for SRC ACP:

1. Click **ACP>Statistics>Device**.

The Statistics/Device pane appears.

Monitor Configure Diagnose Manage Logged in as: admin Refresh Preferences About Logout

ACP
Statistics / Device

Filter

Slot

Style

OK Reset

Name of the device.
Value: All or part of the device name.
Default: No value

Number of the slot for which you want to configure values.
Value: Currently, the chassis has only one slot. The valid value is 0.
Default value: 0

Output style.
Choices:
brief: Display only device names
Default value: detail

Copyright © 2007, Juniper Networks, Inc. All Rights Reserved. Trademark Notice. Privacy. Juniper Your Net.

2. In the Filter box, enter a substring of the virtual router name, or leave the box blank to display information for all virtual routers.
3. In the Slot box, enter the number of the slot for which you want to display device statistics.
4. Select an output style from the Style list.
5. Click **OK**.

The Statistics/Device pane displays router statistics for SRC ACP.

PART 6

Using External Subscriber Monitor

- [Configuring External Subscriber Monitor with the SRC CLI on page 347](#)
- [Monitoring External Subscriber Events with the SRC CLI on page 359](#)
- [Monitoring External Subscriber Events with the C-Web Interface on page 365](#)

CHAPTER 24

Configuring External Subscriber Monitor with the SRC CLI

This chapter describes how you can integrate IP address managers into an SRC-managed network so that the SAE is notified about subscriber events. Topics include:

- [External Subscriber Monitor Overview on page 347](#)
- [Configuring External Subscriber Monitor \(SRC CLI\) on page 348](#)
- [Configuring the NIC Proxy for the Pseudo-RADIUS Server \(SRC CLI\) on page 352](#)
- [Configuring the Pseudo-RADIUS Server for External Subscriber Monitor \(SRC CLI\) on page 354](#)
- [Configuring the Client Secret for External Subscriber Monitor \(SRC CLI\) on page 355](#)
- [Configuring Event Notification for External Subscriber Monitor \(SRC CLI\) on page 356](#)
- [Starting External Subscriber Monitor \(SRC CLI\) on page 357](#)
- [Stopping External Subscriber Monitor \(SRC CLI\) on page 357](#)

External Subscriber Monitor Overview

You use the External Subscriber Monitor application with the event notification method of logging in subscribers and creating subscriber sessions. You can use event notification when you integrate devices into the SRC network that do not notify the SAE about subscriber events, such as when a subscriber logs in or when the address assignment is terminated.

External Subscriber Monitor must view all RADIUS accounting messages associated with subscriber sessions. External Subscriber Monitor is stateless and cannot synchronize the current set of subscribers when there is a failure. If events are missed because of a software or network failure, the overall state recovers when RADIUS interim updates are sent. For example, missed ipUp events become effective when the next interim update is sent, and missed ipDown events time out after the configured RADIUS time to live.

External Subscriber Monitor is configured as a pseudo-RADIUS server and acts as a RADIUS accounting server. Configure the router or RADIUS server to duplicate accounting packets to External Subscriber Monitor. When External Subscriber Monitor is the pseudo-RADIUS server, it handles software failures more robustly. The pseudo-RADIUS

server does not acknowledge failed accounting requests and gives the RADIUS client the option to retransmit the accounting packet to a backup External Subscriber Monitor.

Related Documentation

- For information about event notification with other third-party network devices, see [Logging In Subscribers and Creating Sessions on page 115](#)
- [Configuring External Subscriber Monitor \(SRC CLI\) on page 348](#)
- [Starting External Subscriber Monitor \(SRC CLI\) on page 357](#)
- [Configuring the Pseudo-RADIUS Server for External Subscriber Monitor \(SRC CLI\) on page 354](#)
- [Configuring the Client Secret for External Subscriber Monitor \(SRC CLI\) on page 355](#)
- [Configuring Event Notification for External Subscriber Monitor \(SRC CLI\) on page 356](#)

Configuring External Subscriber Monitor (SRC CLI)

Configure initial properties, including directory connection and directory eventing properties.

Tasks to configure External Subscriber Monitor are:

1. [Configuring Basic Local Properties for External Subscriber Monitor on page 348](#)
2. [Configuring Initial Properties for External Subscriber Monitor on page 349](#)
3. [Configuring Directory Connection Properties for External Subscriber Monitor on page 349](#)
4. [Configuring Eventing Properties for External Subscriber Monitor on page 350](#)
5. [Configuring Logging Destinations for External Subscriber Monitor on page 350](#)

Configuring Basic Local Properties for External Subscriber Monitor

After you complete the configuration changes, restart External Subscriber Monitor for the configuration changes to take effect. Use the following configuration statements to configure basic local properties:

```
slot number external-subscriber-monitor {  
    java-garbage-collection-options java-garbage-collection-option;  
    java-heap-size java-heap-size;  
}
```

To configure basic local properties:

1. From configuration mode, access the configuration statement that configures the local properties.

 user@host# edit slot 0 external-subscriber-monitor
2. Configure the garbage collection functionality of the Java Virtual Machine.

 [edit slot 0 external-subscriber-monitor]
 user@host# set java-garbage-collection-options *java-garbage-collection-options*
3. (Optional) If you encounter problems caused by lack of memory, change the maximum memory size available to the JRE.

```
[edit slot 0 external-subscriber-monitor]
user@host# set java-heap-size java-heap-size
```

4. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor]
user@host# show
```

Configuring Initial Properties for External Subscriber Monitor

Use the following configuration statements to configure initial properties for External Subscriber Monitor:

```
slot number external-subscriber-monitor initial {
  dynamic-dn dynamic-dn;
}
```

To configure initial local properties:

1. From configuration mode, access the configuration statement that configures the initial properties.

```
user@host# edit slot 0 external-subscriber-monitor initial
```

2. Specify the properties for External Subscriber Monitor.

```
[edit slot 0 external-subscriber-monitor initial]
user@host# set ?
```

For more information about configuring local properties for the SRC components, see *Changing the Location of Data in the Directory*.

Configuring Directory Connection Properties for External Subscriber Monitor

Use the following configuration statements to configure directory connection properties for External Subscriber Monitor:

```
slot number external-subscriber-monitor initial directory-connection {
  url url;
  backup-urls backup-urls...;
  principal principal;
  credentials credentials;
  timeout timeout;
  check-interval check-interval;
  blacklist;
  protocol (ldaps);
  snmp-agent;
}
```

To configure directory connection properties:

1. From configuration mode, access the configuration statement that configures the directory connection properties.

```
user@host# edit slot 0 external-subscriber-monitor initial directory-connection
```

2. Specify the properties for External Subscriber Monitor.

```
[edit slot 0 external-subscriber-monitor initial directory-connection]
```

```
user@host# set ?
```

3. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor initial directory-connection]
```

```
user@host# show
```

Configuring Eventing Properties for External Subscriber Monitor

Use the following configuration statements to configure directory eventing properties for External Subscriber Monitor:

```
slot number external-subscriber-monitor initial directory-eventing {  
    eventing;  
    signature-dn signature-dn;  
    polling-intervall polling-interval;  
    event-base-dn event-base-dn;  
    dispatcher-pool-size dispatcherr-pool-size;  
}
```

To configure directory eventing properties:

1. From configuration mode, access the configuration statement that configures the directory eventing properties.

```
user@host# edit slot 0 external-subscriber-monitor initial directory-eventing
```

2. Specify the initial directory eventing properties for External Subscriber Monitor.

```
[edit slot 0 external-subscriber-monitor initial directory-eventing]
```

```
user@host# set ?
```

For more information about configuring local properties for the SRC components, see *Configuring Initial Directory Eventing Properties for SRC Components*.

3. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor initial directory-connection]
```

```
user@host# show
```

Configuring Logging Destinations for External Subscriber Monitor

Use the following configuration statements to configure directory logging destinations for External Subscriber Monitor:

```
slot number external-subscriber-monitor logger logger-name...  
slot number external-subscriber-monitor logger logger-name file {  
    filter filter;  
    filename filename;  
    rollover-filename rollover-filename;  
    maximum-file-size maximum-file-size;  
}  
slot number external-subscriber-monitor logger logger-name syslog {  
    filter filter;  
    host host;  
    facility facility;  
    format format;  
}
```

Configuring Logging Destinations to Store Messages in a File

To configure logging destinations to store log messages in a file:

1. From configuration mode, access the configuration statement that configures the name and type of logging properties. In this sample procedure, the logging destination called file-1 is configured.

```
user@host# edit slot 0 external-subscriber-monitor logger file-1 file
```

2. Specify the properties for the logging destination.

```
[edit slot 0 external-subscriber-monitor logger file-1 file]
user@host# set ?
```

For more information about configuring properties for the logging destination, see *Configuring an SRC Component to Store Log Messages in a File (SRC CLI)*.

3. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor logger file-1 file]
user@host# show
```

Configuring Logging Destinations to Send Messages to System Logging Facility

To configure logging destinations to send log messages to the system logging facility:

1. From configuration mode, access the configuration statement that configures the name and type of logging properties. In this sample procedure, the logging destination is called syslog-1.

```
user@host# edit slot 0 external-subscriber-monitor logger syslog-1 syslog
```

2. Specify the properties for the logging destination.

```
[edit slot 0 external-subscriber-monitor logger syslog-1 syslog]
user@host# set ?
```

For more information about configuring properties for the logging destination, see *Configuring System Logging (SRC CLI)*.

3. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor logger file-1 file]
user@host# show
```

Related Documentation

- [Configuring External Subscriber Monitor \(C-Web Interface\)](#)
- [Starting External Subscriber Monitor \(SRC CLI\) on page 357](#)
- [Viewing Statistics for External Subscriber Monitor \(C-Web Interface\) on page 365](#)
- [External Subscriber Monitor Overview on page 347](#)

Configuring the NIC Proxy for the Pseudo-RADIUS Server (SRC CLI)

Tasks to configure the NIC proxy are:

1. [Configuring Resolution Information for a NIC Proxy on page 352](#)
2. [Changing the Configuration for the NIC Proxy Cache on page 352](#)
3. [Configuring a NIC Proxy for NIC Replication on page 353](#)

Configuring Resolution Information for a NIC Proxy

Use the following configuration statements to configure the NIC proxy:

```
slot number external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
  resolution {
    resolver-name resolver-name;
    constraints constraints;
  }
```

To configure resolution information for a NIC proxy:

1. From configuration mode, access the configuration statement that configures the NIC proxy configuration. In this sample procedure, the NIC proxy called radius-accounting-nic is configured.

```
user@host# edit slot 0 external-subscriber-monitor nic-proxy-configuration
radius-accounting-nic resolution
```

2. Specify the resolution information for this NIC proxy.

```
[edit slot 0 external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
resolution]
user@host# set ?
```

For more information about configuring resolution information for a NIC proxy, see [“Configuring Resolution Information for a NIC Proxy \(SRC CLI\)” on page 188](#).

3. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
resolution]
user@host# show
```

Changing the Configuration for the NIC Proxy Cache

You can modify cache properties for the NIC proxy to optimize the resolution performance for your network configuration and system resources. Typically, you can use the default settings for the cache properties. The configuration statements are available at the Advanced editing level.

Use the following configuration statements to configure the NIC proxy cache:

```
slot number external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
  cache {
    cache-size cache-size;
    cache-cleanup-interval cache-cleanup-interval;
```



```

    cache-entry-age cache-entry-age;
}

```

To configure the cache for a NIC proxy:

1. From configuration mode, access the configuration statement that configures the NIC proxy configuration. In this sample procedure, the NIC proxy called `radius-accounting-nic` is configured.

```

user@host# edit slot 0 external-subscriber-monitor nic-proxy-configuration
radius-accounting-nic cache

```

2. Specify the cache properties for the NIC proxy.

```

[edit slot 0 external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
cache]
user@host# set ?

```

3. (Optional) Verify your configuration.

```

[edit slot 0 external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
cache]
user@host# show

```

Configuring a NIC Proxy for NIC Replication

Typically, you configure NIC replication to keep the NIC highly available. You configure NIC host selection to specify the groups of NIC hosts to be contacted to resolve a request, and to define how the NIC proxy handles NIC hosts that the proxy is unable to contact. The configuration statements are available at the Normal editing level.

Use the following configuration statements to configure NIC host selection for a NIC proxy:

```

slot number external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
  nic-host-selection {
    groups groups;
    selection-criteria (roundRobin | randomPick | priorityList);
  }
slot number external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
  nic-host-selection blacklisting {
    try-next-system-on-error;
    number-of-retries-before-blacklisting number-of-retries-before-blacklisting;
    blacklist-retry-interval blacklist-retry-interval;
  }

```

To configure a NIC proxy to use NIC replication:

1. From configuration mode, access the configuration statement that specifies the NIC proxy configuration. In this sample procedure, the NIC proxy called `radius-accounting-nic` is configured.

```

user@host# edit slot 0 external-subscriber-monitor nic-proxy-configuration
radius-accounting-nic nic-host-selection

```

2. (Optional) Configure NIC host selection for a NIC proxy.

```
[edit slot 0 external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
nic-host-selection]
user@host# set ?
```

For more information about configuring NIC host selection for a NIC proxy, see [“Configuring a NIC Proxy for NIC Replication \(SRC CLI\)” on page 191](#).

3. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
nic-host-selection]
user@host# show
```

4. Access the configuration statement that specifies the NIC proxy configuration for blacklisting—the process of handling nonresponsive NIC hosts.

```
[edit slot 0 external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
nic-host-selection]
user@host# edit blacklisting
[edit slot 0 external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
nic-host-selection blacklisting]
```

5. (Optional) Configure blacklisting for a NIC proxy.

```
[edit slot 0 external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
nic-host-selection blacklisting]
user@host# set ?
```

For more information about configuring NIC host selection for a NIC proxy, see [“Configuring a NIC Proxy for NIC Replication \(SRC CLI\)” on page 191](#).

6. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor nic-proxy-configuration radius-accounting-nic
nic-host-selection blacklisting]
user@host# show
```

Related Documentation

- [Configuring the NIC Proxy for the Pseudo-RADIUS Server \(C-Web Interface\)](#)
- [Configuring the Pseudo-RADIUS Server for External Subscriber Monitor \(SRC CLI\) on page 354](#)
- [External Subscriber Monitor Overview on page 347](#)

Configuring the Pseudo-RADIUS Server for External Subscriber Monitor (SRC CLI)

Use the following configuration statements to configure External Subscriber Monitor as a RADIUS accounting server:

```
slot number external-subscriber-monitor radius-accounting {
  port port;
  service-type (all | login | framed | callback-login | callback-framed | outbound |
    administrative | nas-prompt | authenticate-only | callback-nas-prompt | callback-check
    | callback-administrative);
  allow [allow...];
  deny [deny...];
  maximum-queue-length maximum-queue-length;
```

```
}
```

To configure the RADIUS accounting server:

1. From configuration mode, access the configuration statement that configures the RADIUS server.

```
user@host# edit slot 0 external-subscriber-monitor radius-accounting
```

2. (Optional) Specify the listening port for RADIUS requests.

```
[edit slot 0 external-subscriber-monitor radius-accounting]
user@host# set port port
```

3. (Optional) Specify the service type of the RADIUS packets that will be forwarded.

```
[edit slot 0 external-subscriber-monitor radius-accounting]
user@host# set service-type service-type
```

4. (Optional) Specify a list that filters which packets are forwarded to the SAE based on NAS ID or NAS IP.

```
[edit slot 0 external-subscriber-monitor radius-accounting]
user@host# set allow [allow...]
```

5. (Optional) Specify a list that filters which packets are forwarded to the SAE based on NAS ID or NAS IP.

```
[edit slot 0 external-subscriber-monitor radius-accounting]
user@host# set deny [deny...]
```

6. Specify the maximum number of unacknowledged RADIUS messages to be received from the RADIUS server before it discards new messages.

```
[edit slot 0 external-subscriber-monitor radius-accounting]
user@host# set maximum-queue-length set maximum-queue-length
```

7. (Optional) Verify your configuration.

```
[edit slot 0 external-subscriber-monitor radius-accounting]
user@host# show
```

Related Documentation

- [Configuring the Pseudo-RADIUS Server for External Subscriber Monitor \(C-Web Interface\)](#)
- [Configuring External Subscriber Monitor \(SRC CLI\) on page 348](#)
- [Configuring Event Notification for External Subscriber Monitor \(SRC CLI\) on page 356](#)
- [Configuring the NIC Proxy for the Pseudo-RADIUS Server \(SRC CLI\) on page 352](#)
- [Viewing Statistics for External Subscriber Monitor \(SRC CLI\) on page 359](#)

Configuring the Client Secret for External Subscriber Monitor (SRC CLI)

Use the following configuration statements to configure trusted clients for External Subscriber Monitor. If no clients are configured, all RADIUS accounting packets are discarded.

```
slot number external-subscriber-monitor radius-accounting client client-address {
  secrets secret;
```

```
}
```

To configure trusted clients for External Subscriber Monitor:

1. From configuration mode, access the configuration statement that configures the RADIUS server, and specify the client address.

```
user@host# edit slot 0 external-subscriber-monitor radius-accounting client  
client-address
```

2. Specify the shared secret of the RADIUS client.

```
[edit slot 0 external-subscriber-monitor radius-accounting]  
user@host# set secret secret
```

**Related
Documentation**

- [Configuring the Client Secret for External Subscriber Monitor \(C-Web Interface\)](#)
- [Configuring External Subscriber Monitor \(SRC CLI\) on page 348](#)
- [Configuring the Pseudo-RADIUS Server for External Subscriber Monitor \(SRC CLI\) on page 354](#)
- [Configuring Event Notification for External Subscriber Monitor \(SRC CLI\) on page 356](#)
- [External Subscriber Monitor Overview on page 347](#)

Configuring Event Notification for External Subscriber Monitor (SRC CLI)

Use the following configuration statements to configure External Subscriber Monitor as a RADIUS accounting server:

```
slot number external-subscriber-monitor event-notification {  
  event-threads event-threads;  
  event-thread-idle-timeout event-thread-idle-timeout;  
  event-retry-timeout event-retry-timeout;  
  event-retry-interval event-retry-interval;  
  session-timeout session-timeout;  
}
```

To configure event notification:

1. From configuration mode, access the configuration statement that configures the event notification.

```
user@host# edit slot 0 external-subscriber-monitor event-notification
```

2. (Optional) Specify the maximum number of concurrent threads in a pool for event handlers.

```
[edit slot 0 external-subscriber-monitor event-notification]  
user@host# set event-threads event-threads
```

3. (Optional) Specify the time to keep an event handler alive for reuse.

```
[edit slot 0 external-subscriber-monitor event-notification]  
user@host# set event-thread-idle timeout event-thread-idle-timeout
```

4. (Optional) Specify the maximum retry time before an event is discarded.

```
[edit slot 0 external-subscriber-monitor event-notification]
user@host# set event-retry-timeout event-retry-timeout.
```

5. (Optional) Specify the time to wait before the server retries failed events.

```
[edit slot 0 external-subscriber-monitor event-notification]
user@host# set event-retry-interval event-retry-interval
```

6. Specify the keepalive time for a RADIUS subscriber or service.

```
[edit slot 0 external-subscriber-monitor event-notification]
user@host# set session-timeout session-timeout
```

**Related
Documentation**

- [Configuring Event Notification for External Subscriber Monitor \(C-Web Interface\)](#)
- [Configuring External Subscriber Monitor \(SRC CLI\) on page 348](#)
- [Configuring the Client Secret for External Subscriber Monitor \(SRC CLI\) on page 355](#)
- [External Subscriber Monitor Overview on page 347](#)

Starting External Subscriber Monitor (SRC CLI)

To start External Subscriber Monitor:

- Start External Subscriber Monitor from its installation directory.

```
user@host# enable component extsubmon
```

**Related
Documentation**

- [Starting External Subscriber Monitor \(C-Web Interface\)](#)
- [Stopping External Subscriber Monitor \(SRC CLI\) on page 357](#)
- [Configuring External Subscriber Monitor \(SRC CLI\) on page 348](#)
- [Viewing Statistics for External Subscriber Monitor \(SRC CLI\) on page 359](#)
- [External Subscriber Monitor Overview on page 347](#)

Stopping External Subscriber Monitor (SRC CLI)

To stop External Subscriber Monitor:

- Stop External Subscriber Monitor from its installation directory.

```
user@host# disable component extsubmon
```

**Related
Documentation**

- [Stopping External Subscriber Monitor \(C-Web Interface\)](#)
- [Starting External Subscriber Monitor \(SRC CLI\) on page 357](#)
- [Viewing Statistics for External Subscriber Monitor \(SRC CLI\) on page 359](#)
- [External Subscriber Monitor Overview on page 347](#)

CHAPTER 25

Monitoring External Subscriber Events with the SRC CLI

- Viewing Statistics for External Subscriber Monitor (SRC CLI) on page 359
- Monitoring Statistics for External Subscriber Monitor (SRC CLI) on page 360
- Viewing Statistics for External Subscriber Monitor Event Notifications (SRC CLI) on page 360
- Monitoring Statistics for External Subscriber Monitor Event Notifications (SRC CLI) on page 362
- Viewing Statistics for the Agent Process (SRC CLI) on page 362

Viewing Statistics for External Subscriber Monitor (SRC CLI)

Purpose View RADIUS accounting statistics for External Subscriber Monitor.

Action user@host> **show external-subscriber-monitor statistics radius-accounting**

Client Statistics

Client Address	10.227.7.45
Number of accounting start received	4
Number of accounting stop received	0
Number of accounting interim received	0
Number of discarded accounting requests	0

Meaning [Table 16 on page 359](#) describes the output fields for the **show external-subscriber-monitor statistics radius-accounting** command. Output fields are listed in the order in which they appear.

Table 16: Output Fields for show external-subscriber-monitor statistics radius-accounting

Field Name	Field Description
Client Address	IP address of a RADIUS client. If not specified, displays statistics for all clients.
Number of accounting start received	Number of RADIUS start packets received.
Number of accounting stop received	Number of RADIUS stop packets received.

Table 16: Output Fields for show external-subscriber-monitor statistics radius-accounting (*continued*)

Field Name	Field Description
Number of accounting interim received	Number of RADIUS interim packets received.
Number of discarded accounting requests	Number of RADIUS packets discarded.

- Related Documentation**
- [Configuring External Subscriber Monitor \(SRC CLI\) on page 348](#)
 - [Viewing Statistics for External Subscriber Monitor \(C-Web Interface\) on page 365](#)
 - [Monitoring Statistics for External Subscriber Monitor \(SRC CLI\) on page 360](#)
 - [Viewing Statistics for External Subscriber Monitor Event Notifications \(SRC CLI\) on page 360](#)
 - [Viewing Statistics for the Agent Process \(SRC CLI\) on page 362](#)

Monitoring Statistics for External Subscriber Monitor (SRC CLI)

Purpose Display real-time statistics for External Subscriber Monitor.

Action To display real-time statistics about RADIUS accounting for External Subscriber Monitor:

```
user@host> monitor external-subscriber-monitor radius-accounting client-address
client-address
```

To specify the time for refreshing the data:

```
user@host> monitor external-subscriber-monitor radius-accounting client-address
client-address interval interval
```

- Related Documentation**
- [Viewing Statistics for External Subscriber Monitor \(SRC CLI\) on page 359](#)

Viewing Statistics for External Subscriber Monitor Event Notifications (SRC CLI)

Purpose View statistics for the External Subscriber Monitor event notifications.

Action user@host> show external-subscriber-monitor statistics event-notifications

```
Notification Statistics
Number of ipUp events      8
Number of ipDown events   0
Number of ipUp sent       0
Number of ipDown sent     0
Number of ipUp dropped    0
Number of ipDown dropped  4
Number of ipUp queued     0
Number of ipDown queued   0
Number of IpUp retries    0
Number of ipDown retries  0
```

Meaning [Table 17 on page 361](#) describes the output fields for the **show external-subscriber-monitor statistics event-notifications** command. Output fields are listed in the order in which they appear.

Table 17: Output Fields for show external-subscriber-monitor statistics event-notifications

Field Name	Field Description
Number of ipUp events	Total number of ipUp notification events received, including ipUp sent, ipUp dropped, and ipUp queued
Number of ipDown events	Total number of ipDown notification events received, including ipDown sent, ipDown dropped, and ipDown queued
Number of ipUp sent	Total number of ipUp notification events successfully sent
Number of ipDown sent	Total number of ipDown notification events successfully sent
Number of ipUp dropped	Total number of ipUp notification events dropped due to network failure or difficulties locating managed SAE
Number of ipDown dropped	Total number of ipDown notification events dropped due to network failure or difficulties locating managed SAE
Number of ipUp queued	Total number of ipUp notification events queued to send to SAE
Number of ipDown queued	Total number of ipDown notification events queued to send to SAE
Number of IpUp retries	Total number of ipUp notification events resent tries
Number of ipDown retries	Total number of ipDown notification events resent tries
Number of Nic lookup retries	Total number of NIC lookup retries

Related Documentation

- [Configuring Event Notification for External Subscriber Monitor \(SRC CLI\) on page 356](#)
- [Viewing Statistics for External Subscriber Monitor Event Notifications \(C-Web Interface\) on page 366](#)

- [Monitoring Statistics for External Subscriber Monitor Event Notifications \(SRC CLI\) on page 362](#)
- [Viewing Statistics for External Subscriber Monitor \(C-Web Interface\) on page 365](#)
- [Viewing Statistics for the Agent Process \(SRC CLI\) on page 362](#)

Monitoring Statistics for External Subscriber Monitor Event Notifications (SRC CLI)

Purpose Display real-time statistics about event notifications for External Subscriber Monitor.

Action To display real-time statistics about event notifications for External Subscriber Monitor:

```
user@host> monitor external-subscriber-monitor event-notifications
```

To specify the time for refreshing the data:

```
user@host> monitor external-subscriber-monitor event-notifications interval interval
```

Related Documentation • [Viewing Statistics for External Subscriber Monitor Event Notifications \(SRC CLI\) on page 360](#)

Viewing Statistics for the Agent Process (SRC CLI)

Purpose View statistics for the agent process.

Action user@host> show external-subscriber-monitor statistics process

Process Statistics

```
Up Time      Time1147 seconds since Thu Jan 31 15:56:39 EST 2008
Threads      246
Heap In Use   use142343 kilo bytes
Heap Limit    1012672 kilo bytes
```

Meaning [Table 18 on page 362](#) describes the output fields for the **show external-subscriber-monitor statistics process** command. Output fields are listed in the order in which they appear.

Table 18: Output Fields for show external-subscriber-monitor statistics process

Field Name	Field Description
Up time	Length of time the agent has been running on the system. Includes the date and time at which the agent was last started.
Threads	Number of threads in use.
Heap In Use	Heap size allocated by the Java Virtual Machine. The percentage indicates the percentage of the heap in use. We recommend that if the percent in use is more than 90% additional heap be allocated.
Heap Limit	Size of Java heap configured.

**Related
Documentation**

- [Viewing Statistics for External Subscriber Monitor \(C-Web Interface\) on page 365](#)
- [Viewing Statistics for External Subscriber Monitor Event Notifications \(SRC CLI\) on page 360](#)
- [Viewing Statistics for the Agent Process \(C-Web Interface\) on page 366](#)

CHAPTER 26

Monitoring External Subscriber Events with the C-Web Interface

- [Viewing Statistics for External Subscriber Monitor \(C-Web Interface\) on page 365](#)
- [Viewing Statistics for External Subscriber Monitor Event Notifications \(C-Web Interface\) on page 366](#)
- [Viewing Statistics for the Agent Process \(C-Web Interface\) on page 366](#)

Viewing Statistics for External Subscriber Monitor (C-Web Interface)

Purpose View statistics for External Subscriber Monitor.

Action 1. Click **Monitor>Ext Sub Monitor>Statistics>RADIUS Accounting**.

The Statistics/RADIUS Accounting pane appears.

2. In the Client Address box, enter the address of the client for which you want to view statistics.

3. Select an output style from the Style list.

4. Click **OK**.

The Statistics/RADIUS Accounting pane displays the RADIUS statistics for External Subscriber Monitor.

**Related
Documentation**

- [Configuring External Subscriber Monitor \(C-Web Interface\)](#)
- [Viewing Statistics for External Subscriber Monitor \(SRC CLI\) on page 359](#)
- [Viewing Statistics for External Subscriber Monitor Event Notifications \(C-Web Interface\) on page 366](#)
- [Viewing Statistics for the Agent Process \(SRC CLI\) on page 362](#)

Viewing Statistics for External Subscriber Monitor Event Notifications (C-Web Interface)

- Purpose** View statistics for the External Subscriber Monitor notifications.
- Action**
- Click **Monitor>Ext Sub Monitor>Statistics>Event Notification**.
- The Statistics/Event Notification pane displays the event notification statistics for the External Subscriber Monitor.
- Related Documentation**
- [Configuring Event Notification for External Subscriber Monitor \(SRC CLI\) on page 356](#)
 - [Viewing Statistics for External Subscriber Monitor \(SRC CLI\) on page 359](#)

Viewing Statistics for the Agent Process (C-Web Interface)

- Purpose** View statistics for the agent process.
- Action**
- Click **Monitor>Ext Sub Monitor>Statistics>Process**.
- The Statistics/Process pane displays the process statistics for the agent.
- Related Documentation**
- [Viewing Statistics for the Agent Process \(SRC CLI\) on page 362](#)
 - [Viewing Statistics for External Subscriber Monitor \(SRC CLI\) on page 359](#)
 - [Viewing Statistics for External Subscriber Monitor Event Notifications \(C-Web Interface\) on page 366](#)

PART 7

Using Session State Registrar

- [Session State Registrar Overview on page 369](#)
- [Planning Your Session State Registrar Cluster on page 391](#)
- [Configuring the Session State Registrar \(SRC CLI\) on page 399](#)
- [Managing the SSR Cluster on page 419](#)
- [Monitoring the SSR Cluster on page 423](#)

Session State Registrar Overview

- [Session State Registrar Overview on page 369](#)
- [SSR Node Types on page 370](#)
- [SSR Node Groups on page 371](#)
- [C Series Controller Requirements on page 372](#)
- [SSR Cluster Configurations Overview on page 373](#)
- [Scaling the SSR Cluster on page 374](#)
- [SSR Cluster Network Requirements on page 375](#)
- [Supported SSR Cluster Configurations on page 377](#)
- [SSR Database Schema on page 384](#)
- [Making Modifications to the SSR Database Schema Overview on page 388](#)
- [SSR Database Operating Modes on page 389](#)
- [Distributing the SSR Cluster Configuration and Enabling SSR Client Components on page 389](#)

Session State Registrar Overview

The Session State Registrar (SSR) solution implements a stateless, highly reliable and highly available cluster. It separates front end processes from back-end data functions that take place on two or four data servers. Multiple C Series Controllers collaborate and perform different aspects of operation within the cluster to provide a common sessions database in a highly-available, redundant environment. The common shared resources of the cluster can be accessed simultaneously by up to twenty-four C Series Controllers acting as SSR clients.

The front-end SSR client hosts and SSR back-end data servers collaborate to provide:

- High availability
- Session state preservation during failover of front-end client nodes
- Application session awareness

When used in conjunction with an MX Series router running the packet-triggered subscribers and policy control (PTSP) solution, the SSR stores the IP edge attachment sessions learned from IP edge devices in the centralized SSR database. The IP edge

session stored in the SSR database can be used by the SAE to map the sessions received from the MX Series router. An IP edge session is uniquely identified by IP address and VPN ID, and includes subscriber identity information, which is used to locate the subscriber profile for MX sessions that have the same subscriber IP address.

To work efficiently and to provide redundancy, a production SSR cluster must be built on a fast, isolated, redundant network infrastructure. It must include multiple hosts so a single point of failure does not prevent the cluster from operating.

**Related
Documentation**

- [SSR Node Types on page 370](#)
- [SSR Node Groups on page 371](#)
- [C Series Controller Requirements on page 372](#)
- [SSR Cluster Configurations Overview on page 373](#)

SSR Node Types

An SSR cluster has both a physical and logical organization. A C Series Controller that is a member of the cluster is called a *node*. A process that runs on a node and is part of the cluster is called a *component*. These terms are not interchangeable.

Three types of nodes are included in an SSR cluster, each with a specific role within the cluster:

- A *data node* is a C Series Controller residing in the back end of the cluster. It runs the data storage engine component, which cooperatively manages, replicates, and stores data in the SSR storage engine with other data nodes. Each data node has its own memory and permanent storage, and maintains both a portion of the working copy of the SSR database and a portion of one or more replicas of the SSR database. A cluster can contain either two or four data nodes.
- A *client node* is a C Series Controller that resides in the front end of the SSR cluster. Client nodes are responsible for several functions, including hosting SSR client components, hosting the SSR database front-end component, and optionally hosting the *management server* component.

The client components are the SRC components such as the Subscriber Information Collector (SIC), the Network Information Collector (NIC), the Service Activation Engine (SAE), the IMS Services Gateway, Dynamic Service Activator and other SRC components.

In addition to hosting the SSR client components, each client node hosts a front-end database component that reads and writes data into the SSR database and manipulates the cluster's shared data which is hosted by the data nodes.

At least one client node must host a *management server*. The management server controls itself and all data nodes in the cluster. So that there is no single point of failure, you should configure a minimum of two client nodes to host management servers.

- A *data-client-node* is a C Series Controller that hosts a data node component, a client node component, and a management server. Data-client nodes are used only for

small-scale deployments using the **two-shared-data-node** cluster geometry, or for demonstration purposes when the cluster geometry is set to **all-in-one**.

To uniquely identify the various components running in the cluster, the SRC software assigns a node ID number to each component in the node. Node IDs are generated automatically by the SRC software and cannot be modified.

**Related
Documentation**

- [Session State Registrar Overview on page 369](#)
- [SSR Node Groups on page 371](#)
- [C Series Controller Requirements on page 372](#)
- [Configuring the Nodes in the SSR Cluster \(SRC CLI\) on page 404](#)

SSR Node Groups

Each data node participates in a *node group* of two data nodes. A cluster with two data nodes has a single node group; a cluster with four data nodes has two node groups, each with two data nodes. Each node group stores different partitions and replicas.

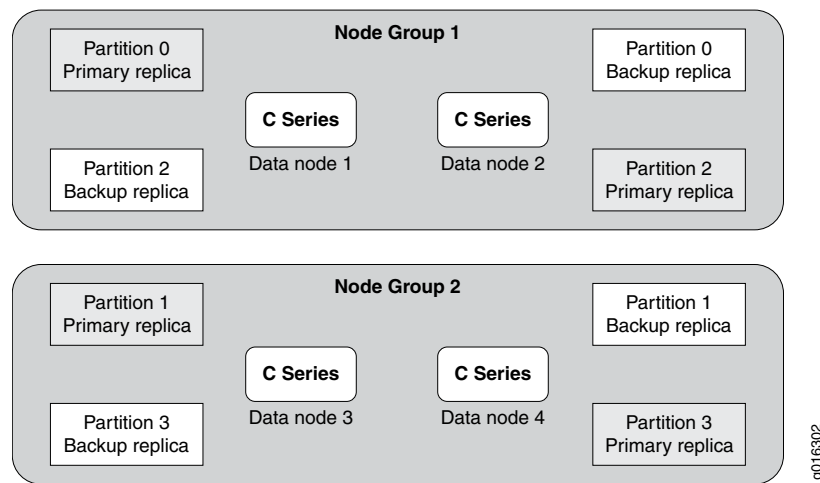
- A *partition* is a portion of all the data stored by the cluster. There are as many cluster partitions as node groups in the cluster. Each node group keeps at least one copy of any partitions assigned to it (that is, at least one replica) available to the cluster.
- A *replica* is a copy of a partition. Each data node in a node group stores a replica of a partition. A replica belongs entirely to a single data node; a node can (and usually does) store several replicas because maintaining two replicas is the fixed setting for the SSR.

[Figure 56 on page 372](#) shows the data elements of an SSR cluster with four data nodes arranged in two node groups of two nodes each. Nodes 1 and 2 belong to Node Group 1. Nodes 3 and 4 belong to Node Group 2.

- Because there are four data nodes, there are four partitions.
- The number of replicas is two, to create two copies of each primary partition.

So long as both nodes in one node group are operating, or one node in each node group is operating, the cluster remains viable.

Figure 56: SSR with Four Data Nodes in Two Groups



The data stored by the cluster in [Figure 56 on page 372](#) is divided into four partitions: 0, 1, 2, and 3. Multiple copies of each partition are stored within the same node group. Partitions are stored on alternate node groups:

- Partition 0 is stored on Node Group 1. A primary replica is stored on Data Node 1 and a backup replica is stored on Data Node 2.
- Partition 1 is stored on the other node group, Node Group 2. The primary replica is on Data Node 3 and its backup replica is on Data Node 4.
- Partition 2 is stored on Node Group 1. The placement of its two replicas is reversed from that of Partition 0; the primary replica is stored on Data Node 2 and the backup on Data Node 1.
- Partition 3 is stored on Node Group 2, and the placement of its two replicas are reversed from those of partition 1: the primary replica is on Data Node 4 and the backup on Data Node 3.

Related Documentation

- [C Series Controller Requirements on page 372](#)
- [SSR Cluster Configurations Overview on page 373](#)
- [Supported SSR Cluster Configurations on page 377](#)
- [Scaling the SSR Cluster on page 374](#)

C Series Controller Requirements

An SSR cluster does not have any single point of failure; each node (C Series Controller) in the cluster has its own memory and disks.

All nodes in the cluster require at least two physical Ethernet ports that provide the same throughput. Bonding the Ethernet interfaces to a single IP address is required. You can accomplish this using SRC group interfaces. Data nodes require 1000Base-T (gigabit Ethernet). For client nodes 100Base-T is sufficient.



NOTE: All data nodes must have equal processor power, memory space, and available bandwidth because they are tightly coupled and share data. If the overall throughput of the data nodes varies from node to node, performance degrades. Therefore, all data nodes must be of the same C Series Controller model—for example, all C4000 models.

Related Documentation

- [SSR Node Types on page 370](#)
- [SSR Cluster Configurations Overview on page 373](#)
- [Supported SSR Cluster Configurations on page 377](#)
- [Planning the SSR Cluster Topology on page 391](#)
- [SSR Cluster Network Requirements on page 375](#)

SSR Cluster Configurations Overview

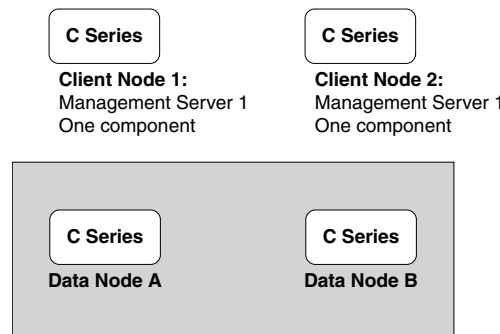
For the highest level of redundancy, each data node must run on its own C Series Controller. Each C Series Controller configured as a client node supports the SSR client components and at least one client node must be configured to host the management server. The management server can run only on a client node. Separation is required so that management arbitration services continue if one of the data node fails. However, for full redundancy, two client nodes should be configured to host management servers.

Using these separation guidelines, we recommend a minimum SSR cluster size of:

- Two client nodes, each running an instance of the management server and a client component
- Two data nodes, each running on its own C Series Controller

This configuration is shown in [Figure 57 on page 373](#):

Figure 57: Basic Session State Registrar Cluster



9016303

Small-Scale Deployments

For small-scale deployments, the minimum SSR configuration requires three C Series Controllers set up in the **two-shared-data-node** geometry. In this configuration, there are two data-client-nodes and two management servers. One management server must be running on the third C Series Controller and the other one can reside on any data-client-node. You also need to run redundant SAEs, SICs, and other components such as NIC to maintain redundancy. Each C Series Controller acts as a backup for the other. This solution requires you to configure the data node memory size because memory is shared with other components.

Related Documentation

- [C Series Controller Requirements on page 372](#)
- [Scaling the SSR Cluster on page 374](#)
- [SSR Cluster Network Requirements on page 375](#)
- [Planning the SSR Cluster Topology on page 391](#)

Scaling the SSR Cluster

You scale your SSR cluster by adding separately licensed Expansion Kits to the Starter Kit. The Starter Kit licenses you for the minimum cluster configuration of two client nodes, each hosting a management server and two data nodes. Expansion Kits are available to scale both the back end and front end of the cluster.

Scaling the Front End of the Cluster

You scale the front end of your SSR cluster by adding licenses for additional client nodes. Optionally, you can add a Management Server Expansion Kit, which allows you to add a third management server component to the on a client node. Each management server component must run on a separate client node.

The service capacity of the SSR cluster grows when you add additional client nodes to the front end. Adding additional client nodes, each of which can host an SSR client component, increases the resiliency of the cluster and the speed of processing a particular transaction because wait time is reduced. Up to twenty four client nodes are supported. At least one of the client nodes must be configured to host the management server component. For redundancy, at least two client nodes must be configured to host the management server component.

The client nodes do not require identical configurations; they can be configured with different components or communications interfaces. For example, one client node might host the Subscriber Information Collector (SIC) component used for the PTSP feature. Another client node might host a different SRC component, such as the SAE or NIC. However, to ensure no single point of failure, we recommend that you configure your cluster with enough client nodes to provide redundancy of the components. For example, for redundancy in a cluster running the SIC component, you would want to have at least:

- Two client nodes, each hosting the SIC component and the management server component

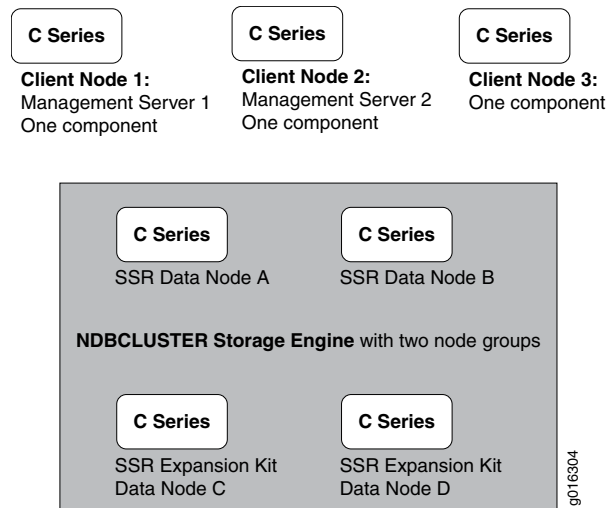
- Two data nodes

Scaling the Back End of the Cluster

You scale the back end of the cluster by adding a data node Expansion Kit which licenses you for two additional data nodes, bringing the total number of data nodes to four, which is the maximum allowed. The additional data nodes form a second node group in the example shown in [Figure 58 on page 375](#), that provides more working memory for the SSR shared database. Each node group manages a partition of the primary SSR database and replicas. The data in each partition is synchronously replicated between the group's data nodes, so if one data node fails, the remaining node can still access all the data. This configuration also provides very quick failover times if a node fails.

Node groupings are managed by the management server. Node groups may not be formed in the same way shown in [Figure 58 on page 375](#). For example, it is possible a new node and an existing node could form one group and the other nodes form another group.

Figure 58: SSR Cluster with Four Data Nodes Forming Two-Node Groups



Related Documentation

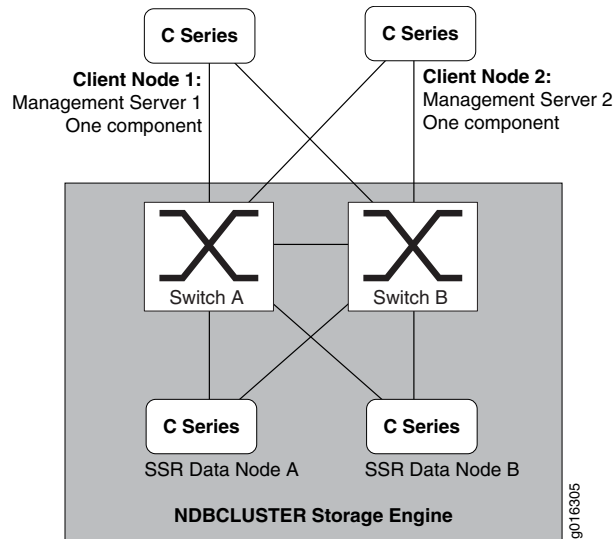
- [SSR Cluster Configurations Overview on page 373](#)
- [SSR Cluster Network Requirements on page 375](#)
- [Supported SSR Cluster Configurations on page 377](#)
- [Planning the SSR Cluster Topology on page 391](#)
- [SSR Cluster Planning Worksheets on page 393](#)

SSR Cluster Network Requirements

A redundant cluster requires a redundant network. We require dual interface cards in each C Series Controller. Use the SRC group interfaces feature to bond the two interfaces to a single IP address.

We recommend that the network be a dedicated subnet with dual switches. This fully duplicates the network, and each C Series Controller in the cluster has at least two routes to all other C Series Controller, as shown in [Figure 59 on page 376](#).

Figure 59: SSR Cluster with Redundant Network



The SSR database schema uses primary key lookups as often as possible during transaction processing, so the database cluster performance scales almost linearly based on the number of data nodes in the cluster.

Do not configure the subnet to be shared beyond the cluster C Series Controllers, because communications between nodes are not encrypted or shielded in any way. The only means of protecting transmissions within a cluster is to run your the cluster on a protected network; do not interpose firewalls between any of the nodes.

Running the cluster on a private or protected network also increases efficiency because the cluster has exclusive use of all bandwidth between cluster nodes. This protects the cluster nodes from interference caused by transmissions between other devices on the network.

The SSR cluster requires Gigabit Ethernet between data nodes and the switch. Client nodes to the switch can use 100Base-T but Gigabit Ethernet is recommended. Network latency can severely degrade performance, as shown in [Table 19 on page 376](#), so we also recommend that all servers be close enough together that latency is always much less than 10 ms.

Table 19: Latency Between Servers and Its Effect on Performance

Latency Times	Performance Degradation
0 ms latency (LAN)	Baseline performance as designed
10 ms latency	Up to 40% performance loss

Table 19: Latency Between Servers and Its Effect on Performance
(continued)

Latency Times	Performance Degradation
20 ms latency	Up to 60% performance loss
More than 20 ms latency	Not supported

Related Documentation

- [C Series Controller Requirements on page 372](#)
- [SSR Cluster Configurations Overview on page 373](#)
- [Scaling the SSR Cluster on page 374](#)
- [Supported SSR Cluster Configurations on page 377](#)
- [Planning the SSR Cluster Topology on page 391](#)

Supported SSR Cluster Configurations

For small-scale deployments, the minimum SSR configuration requires three C series Controllers set up in the **two-shared-data-node** geometry. In this configuration, there are two data-client-nodes and two management servers. One management server must be running on the third C Series Controller and the other one can reside on any data-client-node. Each C Series Controller acts as a backup for the other. This solution requires you to configure the database memory size. This is necessary only under the **two-shared-data-node** cluster geometry. In **two-data-node** and **four-data-node** cluster geometry deployments, all available memory is allocated to the data node process. In the **two-shared-data-node** cluster geometry, memory is shared by all components.

For larger-scale deployments, we recommend you use dedicated C Series Controllers to host each node type. The minimum configuration for a redundant SSR cluster is two client nodes, one of which must host a management server, and two data nodes. A maximum of twenty-four client nodes is supported. For redundancy, at least two client nodes should be configured to host management server components. Data nodes must be added in pairs. The maximum number of data nodes in a cluster is four.

Table 20 on page 378 lists the possible configurations.



CAUTION: Setting up an unsupported configuration can put data and equipment at risk and is not supported by Juniper Networks.

Also, note the latency limitation in [Table 19 on page 376](#). We do not support cluster configurations with latency between nodes that exceeds 20 ms, as can occur if servers are set up to spread a cluster across widely separated locations.

Table 20: Supported Cluster Configurations

Data Nodes	Client Nodes
<p>For small-scale deployments, the minimum SSR configuration requires three C Series Controllers set up in the two-shared-data-node geometry. In this configuration, there are two data-client-nodes and two management servers. One management server must be running on the third C Series Controller and the other one can reside on any data-client-node. Each C Series Controller is configured as a data-client node.</p> <ul style="list-style-type: none"> A data-client node runs a data node component, a client node component, and a management server. You must set the cluster geometry to two-shared-data-node for this configuration. You can add up to 22 additional client nodes. You must configure the database memory when running the two-shared-data-node cluster geometry because all components share the memory. You cannot mix data-client nodes with data nodes. 	
<p>Two</p> <ul style="list-style-type: none"> Each running on its own C Series Controller 	<p>Up to 24</p> <ul style="list-style-type: none"> Each running on its own C Series Controller Up to three configured to run management servers Each hosting SSR client components such as the SIC, SAE and so on Minimum configuration is one client node/management server with one SSR client component (no redundancy)
<p>Four</p> <ul style="list-style-type: none"> Each running on its own C Series Controller 	<p>Up to 24</p> <ul style="list-style-type: none"> Each running on its own C Series Controller Up to three configured to run management servers Each hosting SSR client components such as the SIC, SAE and so on Minimum configuration is one client node/management server with one SSR client component (no redundancy)

Failover Overview

To continue functioning without a service interruption after a component failure, a cluster requires at least 50 percent of its data nodes and client nodes running the management server component to be functional. If more than 50 percent of the data nodes fail, expect a service interruption, but continued operation of the available nodes.

Because SSR client components function as front ends to the back-end data storage portion of the cluster, they are not involved in any failover operations performed by the back-end data components. However, as an administrator, you need to ensure that the front end environment is configured so that it can survive the loss of components.

A data cluster prepares for failover automatically when the cluster starts. During startup, two events occur:

- One of the data nodes (usually the node with the lowest node ID) becomes the *master* of the node group. The master node stores the authoritative copy of the database.

- One data node or management server is elected *arbiter*. The arbiter is responsible for conducting elections among the surviving nodes to determine roles in the event of node failures.

In a cluster, each management server and data node are allocated a vote that is used during this startup election and during failover operations. One management server is selected as the initial arbiter of failover problems and of elections that result from them.

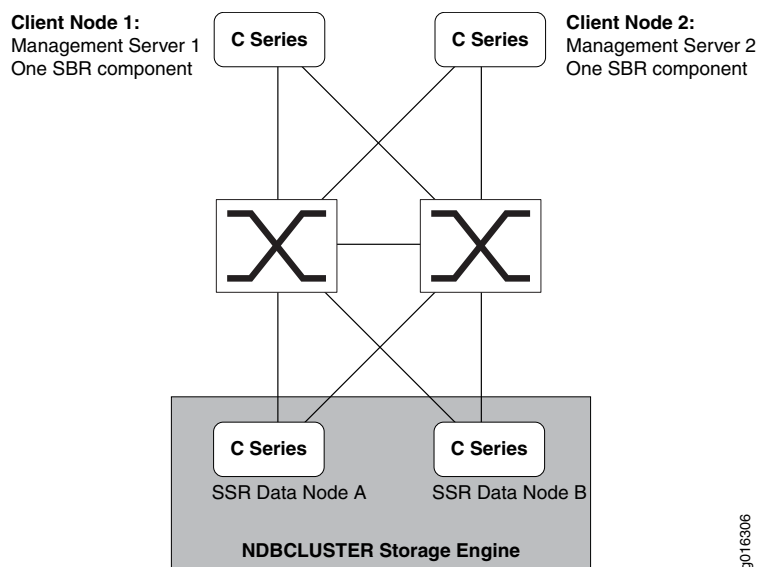
Within the cluster, data nodes and any client nodes hosting management servers monitor each other to detect communications loss. When either type of failure is detected, as long as nodes with more than 50 percent of the votes are operating, there is instantaneous failover and no service interruption. If exactly 50 percent of nodes and votes are lost, and if a data node is one of the lost nodes, the cluster determines which half of the database is to remain in operation. The half with the arbiter (which usually includes the master node) stays up, and the other half shuts down to prevent each node or node group from updating information independently.

When a failed data node (or nodes) returns to service, the working nodes resynchronize the current data with the restored nodes so all data nodes are up to date. How quickly this takes place depends on the current load on the cluster, the length of time the nodes were offline, and other factors.

Failover Examples

The following examples are based on the deployment of two client nodes each hosting a management server, and two data nodes set up with the recommended redundant network as shown in [Figure 60 on page 379](#). Each client node is running a client component. The cluster is set up in a single data center on a fully switched, redundant, layer 2 network. Each of the nodes is connected to two switches using Ethernet bonding for interface failover. The switches have a back-to-back connection.

Figure 60: SSR Cluster with Redundant Network



Possible Failure Scenarios

With this basic configuration, a high level of redundancy is supported. So long as one data node is available to one client node/management server, the cluster is viable and functional.

- If either client node/management server (1 or 2), goes down, the effect on the facility and cluster is:
 - No impact to the SSR client component.
 - Devices (depending on the failover mechanism in the device) switch to their secondary targets—the remaining SSR client node. Recovery of the device when the failed client node returns to service depends on device implementation.
- If either data node A or B goes down, the effect is:
 - No service impact to the SSR client component; both client nodes continue operation using the surviving data node.
 - The management servers running on the client nodes and the surviving data node detect that one data node has gone down, but no action is required, because failover is automatic.
 - When the failed data node returns to service, it synchronizes its data with the surviving data node and resumes operation.
- If both management servers running on client nodes 1 and 2 go down, the effect is:
 - No service impact to the SSR client components, because all client components and data nodes are still available. The data nodes continue to update themselves.
- If both data nodes go down, the effect on:
 - The management servers is minimal. They detect that the data nodes are offline, but can only monitor them.
 - The SSR client components running on the client nodes varies:
 - Sessions that do not require shared resources continue uninterrupted.
 - Sessions that require shared resources are rejected.

The client nodes continue to operate this way until the back-end data cluster comes back online; the cluster resumes normal operation using the data cluster automatically.

- If one half of the cluster (client node 1 and management server 1, and data node A or client node 2 and management server 2, and data node B) go down, the effect is:
 - No service impact, because a client node/management server and a data node are all still in service. Devices using the SSR client component in the failed client node fail over to the other SSR client component in the surviving client node.
 - When the failed data node returns to service, it synchronizes and updates its data with the surviving data node and resumes operation.

- When the failed client node/management server returns to service, the devices assigned to use it as a primary resource return to service depending on the device implementation.

Distributed Cluster Failure and Recovery

You can divide a cluster and separate two equal halves between two data centers. In this case, the interconnection is made by dedicated communications links (shown as red lines in [Figure 61 on page 382](#) and [Figure 62 on page 383](#)) that may be either:

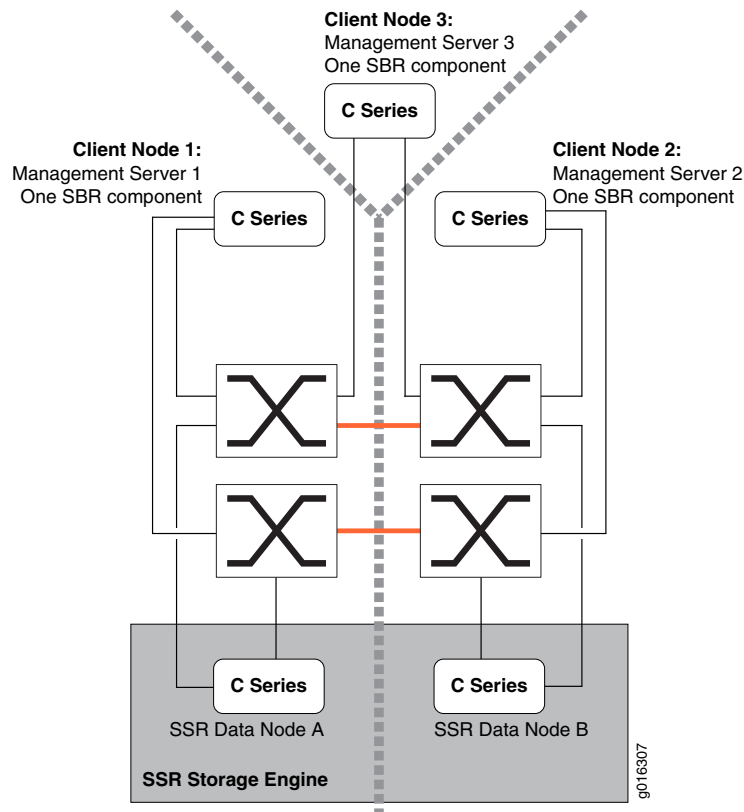
- A switched layer 2 network, just as the single site cluster is set up.
- A routed layer 3 network that uses a routing table with backup routes to route over multiple links between data centers.

However, separating the cluster like this creates a configuration that is vulnerable to a catastrophic failure that severs the two halves of a dispersed cluster. We recommend adding a third client node/management server at a location that has a separate alternative communication route to each half. A third client node/management server:

- Eliminates the possibility of the cluster being evenly split by a communications failure.
- Creates an odd number of votes for elections, which greatly reduces the need for arbitration.

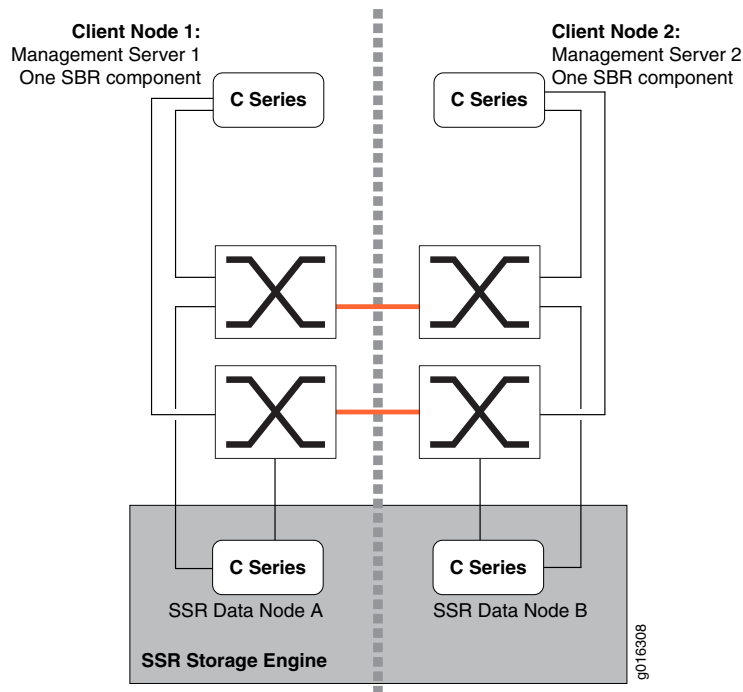
With a third client node/management server in place, failover in the dispersed cluster is well managed because one side of the cluster does not have to determine what role to assume. Recovery is likely to be quicker when the data nodes are reunited because each node's status is more likely to have been monitored by at least one management server that is in communication with each segment.

Figure 61: SSR Cluster Divided Between Two Sites with Tertiary Management Server



Without a third client node/management server, the configuration shown in [Figure 62 on page 383](#) is vulnerable to data loss if both communication links are severed or if the nodes in the master half of the cluster all go offline simultaneously.

Figure 62: SSR Cluster Evenly Divided Between Two Sites



If either of those calamities occur, exactly half the nodes in a side survive. If the master nodes are operating on one or both sides, the cluster continues to function. But the secondary side cannot determine whether the master side is really no longer available, because it only has two votes. It can take 10-15 minutes for the secondary side of the cluster to automatically restart, promote itself to master status, and resume cluster operations.

The SSR client nodes connecting to the secondary side continue to work. However, modifications made to the SSR database may create a divergence between the two copies of the database. The longer the cluster is split, the greater the divergence, and the longer it takes to resolve when recovery takes place.

To eliminate these problems, we recommend a proven alternative: adding another client node with a third management server in a third location that can communicate with each half of the dispersed cluster. Without the tertiary client node/management server, there is a possibility of down time in a dispersed cluster that suffers a catastrophic failure.

If you cannot add a third client node/management server, we recommend that you configure the secondary side of the cluster not to automatically restart, but to go out of service when it instantaneously disconnects from the master side nodes. Then you can determine the best course of action—to keep the cluster offline or to promote the secondary side of the cluster, relink the client node/management servers, and be aware that reconciling the divergence must be part of the recovery procedure.

When the cluster is reunited and goes into recovery mode, the master and secondary data nodes attempt to reconcile the divergence that occurred during separation. The moment they come in contact, transitory failures appear on the client nodes because

the cluster configuration has changed; any transactions that are pending at that moment are aborted. The client nodes retry those transactions because they are classified as temporary failures; in most situations they are accepted on the first retry.

Related Documentation

- [Scaling the SSR Cluster on page 374](#)
- [SSR Cluster Network Requirements on page 375](#)
- [C Series Controller Requirements on page 372](#)
- [Planning the SSR Cluster Topology on page 391](#)
- [SSR Cluster Planning Worksheets on page 393](#)

SSR Database Schema

When used in conjunction with an MX Series router running the packet-triggered subscribers and policy control (PTSP) feature, the Session State Registrar (SSR) stores the IP edge attachment sessions learned from IP edge devices in the SSR centralized database. The IP edge sessions stored in the SSR database can be used by the Service Activation Engine (SAE) to map the sessions received from the MX Series router service node. An IP edge session is uniquely identified by an IP address and VPN ID. It includes subscriber identity information, which is used to locate the subscriber profile for MX Series router sessions that have the same subscriber IP address.

The SSR uses a relational SQL design to store data. The database schema includes multiple tables for storing subscriber identity information for session and service activation requests.



NOTE: If you make changes to the SSR database schema, you must restart all SRC software components.

Subscriber Sessions Table

The format of subscriber identity information stored for each session is managed by the subscriber sessions table, which is controlled by the cluster's client nodes or management servers. The configuration of the subscriber sessions table is copied to all nodes in a node group, so all nodes operate with the same information. The default format of the subscriber sessions table, described in [Table 21 on page 384](#), addresses the needs of most carriers; however, you can modify certain fields to address unique needs and situations.

Table 21: Subscriber Sessions Table Default Fields

Field Name	Field Type	Default Value	Notes
UserIPAddress	binary(4)	NOT NULL	Primary key This column is fixed and is only resizable.

Table 21: Subscriber Sessions Table Default Fields (*continued*)

Field Name	Field Type	Default Value	Notes
VpnID	varchar(16) CHARACTER SET utf8	NOT NULL, DEFAULT ""	Primary key This column is fixed and is only resizable.
UserName	varchar(24) CHARACTER SET utf8	DEFAULT NULL	Indexed by default
IMSI	varchar(15)	DEFAULT NULL	For storing International Mobile Subscriber Identity (IMSI)
CallingStationID	varchar(24) CHARACTER SET utf8	DEFAULT NULL	For storing Mobile Station ISDN (MSISDN) or Calling-Station-ID
CalledStationID	varchar(60) CHARACTER SET utf8	DEFAULT NULL	For storing access point name (APN) or Called-Station-ID
DeviceType	varchar(16) CHARACTER SET utf8	DEFAULT NULL	For storing International Mobile Station Equipment Identity (IMEI)
AccessType	varchar(16) CHARACTER SET utf8	DEFAULT NULL	For storing radio access technology (RAT)
SessionID	varchar(92) CHARACTER SET utf8	NOT NULL	For storing Session ID This column is fixed and is only resizable.
Session State	tinyint unsigned	NOT NULL DEFAULT 1	State of the subscriber session (1 for started, 2 for stopped) This column is fixed and is not resizable.

The primary keys of the subscriber sessions table are the UserIpAddress and VpnID fields. The UserIpAddress field stores the subscriber's IP address in binary format. The default schema uses 4 bytes, which is sufficient for IPv4 addresses. You can modify the length of the UserIpAddress field to 16 bytes if you are using IPv6 addresses. The VpnID field stores the address realm where the user IP address is unique. For non-VPN sessions, the VpnID field must be set to its default value, which is an empty string.

The subscriber sessions table is configurable with some restrictions. You can add new columns, remove existing columns, or modify the column length, type, or index. You cannot remove the UserIpAddress, VpnID, SessionID, or SessionState columns; however, you can modify the lengths of the UserIpAddress and VpnID fields.

Attribute Associations

SSR client components, such as the Service Activation Engine (SAE), Network Information Collector (NIC), Dynamic Service Activator (DSA), and Subscriber Information Collector (SIC), need to read information from the subscriber sessions table and write information to the subscriber sessions table. To support this, you need to specify how SSR client component-specific attributes are translated to subscriber sessions table attributes by defining the mapping between the attributes.

This mapping provides a virtual schema composed of SAE plug-in attributes. The virtual schema is available to all SSR client components— for example, the SAE, NIC, DSA, SIC, and so on. These components use SAE plug-in attributes in the virtual schema to access the subscriber sessions table. The attribute association mapping provides the correlation between the SAE plug-in attributes and the attributes (columns) in the subscriber sessions table.

Use the **shared database cluster primary attribute-associations entity** configuration statement to define the mapping.

Following is a mapping example:

```
shared database cluster primary attribute-associations {
  entity subscriber-sessions field vpn-id{
    sae-plug-in-attribute vpn-id;
  }
}
```

For multivalued dictionary-type SAE plug-in attributes, including PA_PROPERTY and PA_SUBSTITUTION, the suffix can be used to map a specific property or substitution to a field in the SSR subscriber sessions table. For example:

```
shared database cluster primary attribute-associations {
  entity subscriber-sessions{
    field called-station-id{
      sae-plug-in-attribute property.calledStationId;
    }
    field calling-station-id{
      sae-plug-in-attribute property.callingStationId;
    }
  }
}
```

In this example, the PA_PROPERTY plug-in attribute contains two properties: calledStationId and callingStationId. The calledStationId property is mapped to the called-station-id field in the subscriber sessions table, and the callingStationId is mapped to the calling-station-id field. You can also map a multivalued SAE plug-in attribute to a field in the subscriber sessions table. Multiple values are concatenated together using a separator, and stored in the SSR field. When read from the SSR database, the multivalued attribute is restored from the field.

Service Sessions Table

The service sessions table stores service activation requests received through the Application Services Gateway (ASG) used by the PTSP feature of the MX Series router to create a service session in the Session State Registrar (SSR). When the ASG receives a service activation request, it queries the subscriber sessions table by using the subscriber's VpnID and IP address, which are either received from the request directly or obtained through a network information collector (NIC) lookup if the subscriber ID in the request is not a combination of the VpnID and IP address. If the specified combination of VpnID and IP address does not exist, the ASG responds with an error message that the subscriber is unknown. Otherwise, the ASG stores the service activation request with the SessionID attribute, which is taken from the attachment session record, along with SubscriptionName, SessionName, and activation attributes from the request. The SAE is notified through the SSR event interface when the service sessions table is updated—that is, a record is created, updated, or removed. The SAE also queries the service sessions table for service activations when it starts a new PTSP session.

[Table 22 on page 387](#) describes the fields and default values for the service sessions table.

Table 22: Service Sessions Table Default Fields

Field Name	Field Type	Default Value	Notes
SessionID	varchar(92) CHARACTER SET utf8	NOT NULL	Primary key
SubscriptionName	varchar(31) CHARACTER SET utf8	NOT NULL	Primary key
SessionName	varchar(15) CHARACTER SET utf8	NOT NULL, DEFAULT ""	Primary key
ActivationAttributes	varchar(1023) CHARACTER SET utf8	DEFAULT NULL	Serialized activation attributes of the AttrSeq type (defined in the SAE external interface)

The service sessions table contains a composite primary key of SessionID, SubscriptionName, and SessionName. SessionID uniquely identifies an attachment session. SubscriptionName and SessionName identify a service session for a given attachment session.

ActivationAttributes is an attribute of the varchar type that is used to store service activation attributes that are provided to the ASG with the request. ActivationAttributes is the JavaScript Object Notation (JSON) serialized form of the AttrSeq attribute, which is a sequence of structured attributes:

```
struct Attr {
    string name;
    WStringSeq values;
}
```

The AttrSeq attribute that is serialized with JSON should resemble [{name:"name", values:["val1", ...],...}].

The service sessions table is not configurable. If you modify the lengths of the UserIPAddress and VpnID fields in the subscriber sessions table, the SRC software adjusts the lengths of these fields in the service sessions table.

**Related
Documentation**

- [Configuration Changes and Their Impact on the SSR Cluster on page 399](#)
- [Mapping SAE Plug-In Attributes to Fields in the Subscriber Sessions Table \(SRC CLI\) on page 409](#)
- [Configuring the Fields in the Subscriber Sessions Table \(SRC CLI\) on page 407](#)
- [Modifying the SSR Database Schema in an Active Cluster \(SRC CLI\) on page 410](#)

Making Modifications to the SSR Database Schema Overview

Whenever you have modified, added, or deleted fields from the subscriber sessions table, you apply the new database schema by destroying and re-creating the SSR database.



NOTE: This is a destructive process, and must be performed during a schedule maintenance window. Destroying the database causes all data in the database to be lost.

Because the configuration of the database is stored in Juniper Networks database, modifications to the database schema can be made from any node in the SSR cluster. However, destroying and recreating the database can be performed only from a client node (with or without a management server).



NOTE: Changes to the SSR database schema requires you to restart all SRC components.

**Related
Documentation**

- [Modifying the SSR Database Schema in an Active Cluster \(SRC CLI\) on page 410](#)
- [SSR Database Schema on page 384](#)
- [Configuring the Fields in the Subscriber Sessions Table \(SRC CLI\) on page 407](#)
- [Mapping SAE Plug-In Attributes to Fields in the Subscriber Sessions Table \(SRC CLI\) on page 409](#)
- [Creating the SSR Database \(SRC CLI\) on page 420](#)
- [Deleting the SSR Database \(SRC CLI\) on page 420](#)

SSR Database Operating Modes

SSR has both a running configuration and an offline configuration. In addition, the database has two modes of operation: maintenance mode and production mode. The running configuration is the configuration currently running on the SSR database. The offline configuration is used to store the configuration entered through the SRC CLI while the SSR database is running. When you commit your changes through the SRC CLI, the configuration is written to offline configuration. In maintenance mode, database components read the offline configuration. The moment the SSR database is placed into production mode, a snapshot of the offline configuration is taken and saved in to the running configuration. In production mode, SSR database components only read the running configuration. Any configuration changes committed through the SRC CLI are stored in offline configuration and do not affect the SSR database components. Placing the SSR database into maintenance mode causes the SRC components to read from the offline configuration. The previous running configuration is dumped. The purpose of production mode is to protect SSR database components from accidental configuration changes. Maintenance mode allows you to stage configuration changes without impacting your working cluster.

Related Documentation

- [Placing the SSR Database into Maintenance Mode \(SRC CLI\) on page 419](#)
- [Placing the SSR Database into Production Mode \(SRC CLI\) on page 419](#)

Distributing the SSR Cluster Configuration and Enabling SSR Client Components

The SSR is a distributed system. Multiple C Series Controllers participate in the SSR cluster, and each system has a different role and hosts a different type of cluster node. Instead of configuring each C Series Controller individually by its role in the SSR cluster, the configuration is stored in the centralized Juniper Networks database. You can configure the SSR cluster from any C Series Controller. The SSR driver in a C Series Controller distributes the configuration to each node in the cluster. When you enable the SSR component in the system, the SSR driver loads the configuration from Juniper Networks database and finds the role of the system in SSR. Then it loads the relevant configuration for the specific role of the node. Finally, it starts the SSR components corresponding to the role of the node.

Enabling, Restarting and Disabling the SSR Component Database

Although the configuration is distributed to all nodes in the SSR cluster, enabling, disabling and restarting the SSR component database is a local function, which must be performed on each node in the cluster. Enabling or restarting the component database on a node starts or restarts the processes for all SSR components in the node. Disabling the component database on a node stops all processes for all SSR components in the node. After initially configuring your cluster, you must enable the component database on each node in the cluster one by one. In addition, certain configuration changes require you to disable, enable, or restart the component database on each node in the cluster one by one. Be sure to review and carefully follow the procedures described in [“Configuring the Initial SSR Cluster \(SRC CLI\)” on page 401](#). When making changes to an active cluster, be sure to review each maintenance procedure beforehand and follow the steps carefully.

**Related
Documentation**

- *Juniper Networks Database Overview*
- [Configuring the Initial SSR Cluster \(SRC CLI\) on page 401](#)
- [Modifying the SSR Database Schema in an Active Cluster \(SRC CLI\) on page 410](#)
- [Adding Data Nodes to an Active SSR Cluster \(SRC CLI\) on page 412](#)
- [Adding Client Nodes to an Active SSR Cluster \(SRC CLI\) on page 413](#)
- [Adding a Management Server to an Active SSR Cluster on page 414](#)
- [Removing Data Nodes from an Active SSR Cluster on page 415](#)
- [Removing a Client Node from an Active SSR Cluster on page 417](#)
- [Removing a Management Server from an Active SSR Cluster on page 418](#)

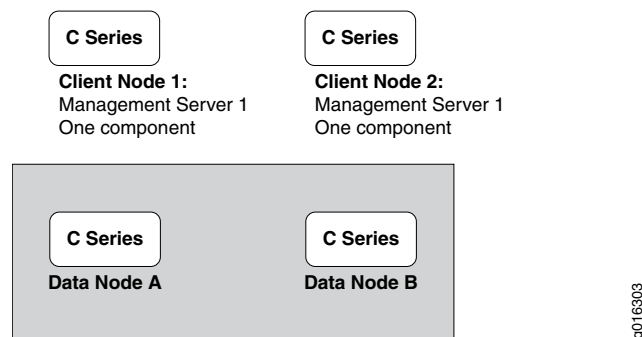
Planning Your Session State Registrar Cluster

- [Planning the SSR Cluster Topology on page 391](#)
- [SSR Cluster Planning Worksheets on page 393](#)

Planning the SSR Cluster Topology

The topology of all SSR clusters begins with the four C Series Controllers required to implement the SSR Starter Kit which licenses you for two C Series Controllers configured as client nodes, each hosting a management server and SSR client components; and two C Series Controllers, each hosting a data node. [Figure 63 on page 391](#) shows the SSR Starter Kit basic configuration.

Figure 63: Basic SSR Starter Kit Cluster



Small-Scale Deployments

For small-scale deployments, the minimum SSR configuration requires three C Series Controllers set up in the **two-shared-data-node** geometry. In this configuration, there are two data-client-nodes and two management servers. One management server must be running on the third C Series Controller and the other one can reside on any data-client-node. You also need to run redundant SAEs, SICs, and other SRC components such as NIC to maintain redundancy. This solution requires you to configure the data node memory size because memory is shared among all components.

Identifying the C Series Controllers in the SSR Cluster

Each node in the cluster is identified by its IP address. The type of node can be either a data node, a client node, or a data client node. Data-client nodes can be used only in **two-shared-data-node** geometry, or for demo purposes. Up to four data nodes and twenty-four client nodes can be configured. The number of data nodes configured must match the template selected with the geometry option (**two-data-node**, **four-data-node**, or **two-shared-data-node**).

- Data node—C Series Controller running an instance of the data node component.
- Client node—C Series Controller running an instance of a Juniper Networks component that is a client of the SSR (for example, SAE or SIC). In addition, each client node hosts an instance of the database front-end component (SQL node component). Up to three client nodes may optionally be configured to run an instance of the management server component.
- Data-client-node—C Series Controller hosting a data node component, a client node component, and a management server. Data-client-nodes are used only for small-scale deployments using the **two-shared-data-node** cluster geometry, or for demonstration purposes when the cluster geometry is set to **all-in-one**.



NOTE: All data nodes must have equal processor power, memory space, and available bandwidth because they are tightly coupled and share data. If the overall throughput of the data nodes varies from node to node, performance degrades. Therefore, all data nodes must be of the same C Series Controller model—for example, all C4000 models.

An example of node ID assignments is shown in [Table 23 on page 392](#). In this example, the cluster consists of two clients nodes, each hosting an SSR client component, and two data nodes.



NOTE: Because Ethernet bonding is recommended, only one IP address is used for each C Series Controller.

Table 23: Example Allocation of Node IDs

IP Address and Type	Bonded IP Address	Component	Node ID Assigned to Component
192.168.0.18 (Data node)	192.168.0.18	data component	18
192.168.0.19 (Data node)	192.168.0.19	data component	19

Table 23: Example Allocation of Node IDs (*continued*)

IP Address and Type	Bonded IP Address	Component	Node ID Assigned to Component
192.168.0.1 (Client node)	192.168.0.1	Management server component	1
		Database front-end component (SQL component)	6
			41
		Client component	
192.168.0.2 (Client node)	192.168.0.2	Management server component	2
		Database front-end component (SQL component)	7
			42
		Client component	

Related Documentation

- [C Series Controller Requirements on page 372](#)
- [SSR Node Types on page 370](#)
- [Supported SSR Cluster Configurations on page 377](#)
- [SSR Cluster Configurations Overview on page 373](#)
- [Scaling the SSR Cluster on page 374](#)
- [SSR Cluster Planning Worksheets on page 393](#)

SSR Cluster Planning Worksheets

Use the worksheet shown in [Table 24 on page 393](#) to plan your SSR Starter Kit cluster. The worksheet allows you to record IP addresses and node IDs for the four C Series Controllers. Record the node IDs after the SRC software has assigned them. If you use more nodes, see the worksheet in [Table 25 on page 394](#) for the Expansion Kit.

Table 24: SSR Starter Kit Cluster Worksheet

Nodes and Hosted Components	Node IP Address	Bonded IP Address	Node ID Assigned to Component
Client node hosting:			
<ul style="list-style-type: none"> • Database front-end component (SQL) • Management server component • Client component 			

Table 24: SSR Starter Kit Cluster Worksheet (*continued*)

Nodes and Hosted Components	Node IP Address	Bonded IP Address	Node ID Assigned to Component
Client node hosting:			
<ul style="list-style-type: none"> Database front-end component (SQL) Management server component Client component 			
Data node hosting:			
<ul style="list-style-type: none"> back-end data component 			
Data node hosting:			
<ul style="list-style-type: none"> back-end data component 			

A cluster can include either two or four data nodes, up to three management servers and up to twenty-four client nodes, each hosting SSR client components such as the SIC, SAE and so on. The worksheet in [Table 25 on page 394](#) provides locations for you to record the IP addresses and node IDs for this maximum configuration.

Table 25: Expanded SSR Cluster Planning Worksheet

Product Name	Nodes and Hosted Components	Node IP Address	Bonded IP Address	Node ID Assigned to Component
Data Node Expansion Kit	Data node hosting:			
	<ul style="list-style-type: none"> Back-end data component 			
	Data node hosting:			
	<ul style="list-style-type: none"> Back-end data component 			
Management Server Expansion Kit	Management server component			

Table 25: Expanded SSR Cluster Planning Worksheet (continued)

[illegible]

Table 25: Expanded SSR Cluster Planning Worksheet (*continued*)

Product Name	Nodes and Hosted Components	Node IP Address	Bonded IP Address	Node ID Assigned to Component
	Client node hosting: <ul style="list-style-type: none"> Database front-end component Client component 			
	Client node hosting: <ul style="list-style-type: none"> Database front-end component Client component 			
	Client node hosting: <ul style="list-style-type: none"> Database front-end component Client component 			
	Client node hosting: <ul style="list-style-type: none"> Database front-end component Client component 			
	Client node hosting: <ul style="list-style-type: none"> Database front-end component Client component 			
	Client node hosting: <ul style="list-style-type: none"> Database front-end component Client component 			
	Client node hosting: <ul style="list-style-type: none"> Database front-end component Client component 			
	Client node hosting: <ul style="list-style-type: none"> Database front-end component Client component 			

Table 25: Expanded SSR Cluster Planning Worksheet *(continued)*

Product Name	Nodes and Hosted Components	Node IP Address	Bonded IP Address	Node ID Assigned to Component
	Client node hosting: <ul style="list-style-type: none">Database front-end componentClient component			
	Client node hosting: <ul style="list-style-type: none">Database front-end componentClient component			
	Client node hosting: <ul style="list-style-type: none">Database front-end componentClient component			
	Client node hosting: <ul style="list-style-type: none">Database front-end componentClient component			
	Client node hosting: <ul style="list-style-type: none">Database front-end componentClient component			
	Client node hosting: <ul style="list-style-type: none">Database front-end componentClient component			

Related Documentation

- [SSR Node Types on page 370](#)
- [Supported SSR Cluster Configurations on page 377](#)
- [SSR Cluster Configurations Overview on page 373](#)
- [Scaling the SSR Cluster on page 374](#)
- [Planning the SSR Cluster Topology on page 391](#)

Configuring the Session State Registrar (SRC CLI)

- [Configuration Changes and Their Impact on the SSR Cluster on page 399](#)
- [Configuration Statements for the SSR Cluster on page 400](#)
- [Configuring the Initial SSR Cluster \(SRC CLI\) on page 401](#)
- [Configuring the SSR Cluster ID \(SRC CLI\) on page 402](#)
- [Configuring the SSR Cluster Geometry \(SRC CLI\) on page 403](#)
- [Configuring the Nodes in the SSR Cluster \(SRC CLI\) on page 404](#)
- [Configuring the Management Servers in the SSR Cluster \(SRC CLI\) on page 406](#)
- [Configuring the Database Memory Size When Running the Two-Shared-Data-Node Geometry \(SRC CLI\) on page 406](#)
- [Configuring the Fields in the Subscriber Sessions Table \(SRC CLI\) on page 407](#)
- [Mapping SAE Plug-In Attributes to Fields in the Subscriber Sessions Table \(SRC CLI\) on page 409](#)
- [Modifying the SSR Database Schema in an Active Cluster \(SRC CLI\) on page 410](#)
- [Modifying Attribute Mapping in an Active SSR Cluster \(SRC CLI\) on page 411](#)
- [Adding Data Nodes to an Active SSR Cluster \(SRC CLI\) on page 412](#)
- [Adding Client Nodes to an Active SSR Cluster \(SRC CLI\) on page 413](#)
- [Adding a Management Server to an Active SSR Cluster on page 414](#)
- [Removing Data Nodes from an Active SSR Cluster on page 415](#)
- [Removing a Client Node from an Active SSR Cluster on page 417](#)
- [Removing a Management Server from an Active SSR Cluster on page 418](#)

Configuration Changes and Their Impact on the SSR Cluster

Certain configuration changes made on an active SSR cluster may require you to shut down and reinitialize the cluster—for example, when the cluster geometry is changed or when modifications are made to the database schema. Here are some examples of configuration changes and their impact on the SSR cluster:

- Dropping or adding of columns in the SSR database schema requires you to re-create the database by executing the **request database delete database** and then **request database create database** commands. It does not require shutdown of the cluster data nodes.



NOTE: Changes to the SSR database schema requires you to restart all SRC components.

- Moving from a two-node geometry to a four-node geometry, or vice versa, requires restarting of all management servers and data nodes in the SSR cluster by executing the **restart component database** command on each of the data nodes and management servers
- Adding or removing client nodes requires restarting all management servers and the new client node (or shutdown the removed client node) in the SSR cluster by executing the **restart component database** command on each management server and the new client node.

To facilitate SSR maintenance, a command is used to set the SSR database into maintenance mode or production mode.

Related Documentation

- [Placing the SSR Database into Maintenance Mode \(SRC CLI\) on page 419](#)
- [Modifying the SSR Database Schema in an Active Cluster \(SRC CLI\) on page 410](#)
- [Modifying Attribute Mapping in an Active SSR Cluster \(SRC CLI\) on page 411](#)
- [Adding Data Nodes to an Active SSR Cluster \(SRC CLI\) on page 412](#)
- [Adding Client Nodes to an Active SSR Cluster \(SRC CLI\) on page 413](#)
- [Adding a Management Server to an Active SSR Cluster on page 414](#)

Configuration Statements for the SSR Cluster

Use the following statements to configure the SSR cluster at the **[edit]** hierarchy level:

```
shared database cluster {
  primary;
}
shared database cluster primary nodes {
  geometry [(all-in-one | two-data-node | four-data-node | two-shared-data-node)];
}
shared database cluster primary nodes {
  node address address;
  platform [(C2000 | C3000 | C4000 | C5000)];
  type [(data-node | client-node | data-client-node)];
}
shared database cluster primary nodes node address client-node {
  management-server;
}
shared database cluster (primary) nodes database-size {
```



```

data-memory data-memory;
index-memory index-memory;
}
shared database cluster primary schema table subscriber-sessions field name {
  type [(int | string | binary)];
  size size;
  require-value
  indexed;
  default default;
  variable-length;
}
shared database cluster primary attribute-associations table name field name name {
  sae-plugin-attribute sae-plugin-attribute;
}

```

Related Documentation

- [Configuring the Initial SSR Cluster \(SRC CLI\) on page 401](#)
- [Session State Registrar Overview on page 369](#)
- [Configuration Changes and Their Impact on the SSR Cluster on page 399](#)

Configuring the Initial SSR Cluster (SRC CLI)

Configuring your initial cluster is a multistep process that requires you to install your C Series Controllers, load the SRC software, and perform the following configuration steps.

For information about supported cluster configurations see [“Supported SSR Cluster Configurations” on page 377](#).



NOTE: This procedure assumes that you are configuring the cluster for the first time and as such, the SSR database is already in maintenance mode.

To configure your initial SSR cluster:

1. Configure the cluster ID.
See [“Configuring the SSR Cluster ID \(SRC CLI\)” on page 402](#).
2. Configure the cluster geometry.
See [“Configuring the SSR Cluster Geometry \(SRC CLI\)” on page 403](#).
3. Configure the role of each node in the cluster
See [“Configuring the Nodes in the SSR Cluster \(SRC CLI\)” on page 404](#).
4. Configure the management servers.
See [“Configuring the Management Servers in the SSR Cluster \(SRC CLI\)” on page 406](#).
5. (Optional) Make desired changes to the default subscriber sessions table.
See [“Configuring the Fields in the Subscriber Sessions Table \(SRC CLI\)” on page 407](#).

6. (Optional) If you are configuring the two-shared-data-node geometry, configure the database memory size.
[See “Configuring the Database Memory Size When Running the Two-Shared-Data-Node Geometry \(SRC CLI\)” on page 406.](#)
7. (Optional) Make desired changes to the default attribute associations table.
[See “Mapping SAE Plug-In Attributes to Fields in the Subscriber Sessions Table \(SRC CLI\)” on page 409.](#)
8. Commit the configuration.
[See *Committing a Configuration and Exiting Configuration Mode*.](#)
9. Enable the component database (must be done on each node in the SSR cluster.)
[See “Enabling the SSR Database \(SRC CLI\)” on page 421.](#)
10. Create the SSR database. (There is no need to destroy the database because it does not exist at initial startup).
[See “Creating the SSR Database \(SRC CLI\)” on page 420.](#)
11. Enable each SSR client component.
[See *Enabling SRC Components*.](#)
12. Verify status of the cluster.
[See “Viewing the Status of the SSR Cluster \(SRC CLI\)” on page 423.](#)
13. Place the cluster into production mode.
[See “Placing the SSR Database into Production Mode \(SRC CLI\)” on page 419.](#)

Related Documentation

- [Configuration Statements for the SSR Cluster on page 400](#)
- [Session State Registrar Overview on page 369](#)
- [Configuration Changes and Their Impact on the SSR Cluster on page 399](#)
- [SSR Node Types on page 370](#)
- [SSR Database Schema on page 384](#)

Configuring the SSR Cluster ID (SRC CLI)

Use the following configuration statements to configure the cluster ID:

```
shared database cluster {  
    primary;  
}
```

Configure the cluster ID from any node in the cluster:



NOTE: In this release of SRC software, the cluster ID is fixed to *primary* and cannot be modified. Only one cluster is supported.

- Set the cluster ID:

```
user@host# edit shared database cluster primary
```

**Related
Documentation**

- [Planning the SSR Cluster Topology on page 391](#)
- [SSR Cluster Planning Worksheets on page 393](#)

Configuring the SSR Cluster Geometry (SRC CLI)

The cluster geometry specifies the number of data nodes in the cluster. In a two-data-node geometry, the two data nodes are configured in the same node group, which means they are backups for each other. In a four-data-node geometry, four data nodes are configured in two node groups. Each node group hosts 50% of the data and contains two data node servers. The two data nodes in each group are backups for each other.



NOTE: All data nodes must have equal processor power, memory space, and available bandwidth because they are tightly coupled and share data. If the overall throughput of the data nodes varies from node to node, performance degrades. Therefore, all data nodes must be of the same C Series Controller model.

Use the following configuration statements to configure the cluster geometry:

```
shared database cluster primary nodes {
  geometry [(all-in-one | two-data-node | four-data-node | two-shared-data-node)];
}
```

Configure the cluster geometry from any node in the cluster:

1. From configuration mode enter the statement to configure the cluster geometry:

```
user@host# edit shared database cluster primary nodes
```

2. Specify the cluster geometry. For example, to set the geometry for two data node:

```
[edit shared database cluster primary nodes]
user@host# set geometry two-data-node
```

The geometry can be set as follows:

- **two-data-node**—In the two-data-node geometry, the two data nodes are configured in the same node group, which means they are backups for each other.
- **four-data-node**—In the four-data-node geometry, four data nodes are configured in two node groups. Each node group hosts 50% of the data and contains two data node servers. The two data nodes in each group are backups for each other.
- **two-shared-data-node**—In the two-shared-data-node geometry, a minimum of three C Series Controllers is required. In this configuration, there are two data-client-nodes

and two management servers. One management server must be running on the third C Series Controller and the other one can reside on any data-client-node. This option is for small-scale deployments. You cannot mix data nodes with data client nodes. Using this geometry, you can have up to twenty-two client nodes.

- **all-in-one**—This option is for demonstration purposes only, and requires the node type to be configured as a **data-client-node**.

**Related
Documentation**

- [SSR Cluster Configurations Overview on page 373](#)
- [SSR Cluster Network Requirements on page 375](#)
- [Configuring the Database Memory Size When Running the Two-Shared-Data-Node Geometry \(SRC CLI\) on page 406](#)
- [Scaling the SSR Cluster on page 374](#)
- [Supported SSR Cluster Configurations on page 377](#)

Configuring the Nodes in the SSR Cluster (SRC CLI)

The node configuration is a list of all nodes in the cluster. Each entry in the list declares either a data node or a client node. For client nodes, an optional keyword specifies that a management server is enabled on the node.

The SSR database contains a node collection. Each node is identified by its IP address. The type of node can be set to either **data-node**, **client-node** or **data-client-node**, which can only be used when the cluster geometry is set to **all-in-one**. If the node type is set to **client-node** or **data-client-node**, a client node object appears under the node for setting the client node related options. You can configure up to four data nodes and twenty-four client nodes.



NOTE: The number of data nodes configured must match the template selected with the geometry setting.



NOTE: All data nodes must have equal processor power, memory space, and available bandwidth because they are tightly coupled and share data. If the overall throughput of the data nodes varies from node to node, performance degrades. Therefore, all data nodes must be of the same C Series Controller model—for example, C4000 models.

Use the following configuration statements to configure at least one client node and two data nodes in the cluster:

```
shared database cluster primary nodes{
  node address address;
  platform [(C2000 | C3000 | C4000 | C5000)];
  type [(data-node | client-node | data-client-node)];
}
```

To configure each node in the cluster (perform steps from any node in the cluster and repeat for each node in the SSR cluster):

1. From configuration mode, access the statement to configure the cluster nodes.

```
user@host# edit shared database cluster primary nodes node
```

2. Specify the node's IP address.

```
[edit shared database cluster primary nodes node]  
user@host# set address address
```

3. Specify the platform type for the node.

```
[edit shared database cluster primary nodes node]  
user@host# set platform [(2000 | C3000 | C4000 | C5000)]
```

4. Configure the node type.

```
[edit shared database cluster primary nodes node]  
user@host# set type [(data-node | client-node | data-client-node)]
```

where the type is one of the following values:

- **data-node**—Configures the machine to host a data node. Data nodes are always configured in pairs, and your cluster can contain either two or four data nodes.
- **client-node**—Configures the machine to host a client node. Client nodes can optionally host a management server.
- **data-client-node**—Configures the machine to host a data node, a client node, and a management server. Use this node type only for small-scale deployments when the cluster geometry is set for **two-shared-data-node**, or for demonstration purposes when the cluster geometry is set to **all-in-one**.

Related Documentation

- [SSR Node Types on page 370](#)
- [Configuring the Database Memory Size When Running the Two-Shared-Data-Node Geometry \(SRC CLI\) on page 406](#)
- [SSR Cluster Configurations Overview on page 373](#)
- [Scaling the SSR Cluster on page 374](#)
- [Supported SSR Cluster Configurations on page 377](#)
- [Configuring the Management Servers in the SSR Cluster \(SRC CLI\) on page 406](#)

Configuring the Management Servers in the SSR Cluster (SRC CLI)

At least one client node must host the management server process. For redundancy at least two client nodes must be configured to host management server processes. Each management server must run on a separate client node.

Use the following configuration statement to configure the management server on a client node:

```
shared database cluster (primary) nodes node address client-node {  
    management-server;  
}
```

To configure the management server, perform these steps from any node in cluster and repeat steps for each client node hosting a management server:

1. From configuration mode, access the statement to configure at least one management server, and specify the IP address of the client node you want to host the management server.

```
user@host# edit shared database cluster primary nodes node address client-node
```

2. Configure the client node to host a management server.

```
[edit shared database cluster primary nodes node address client-node]  
user@host# set management-server
```

Related Documentation

- [SSR Node Types on page 370](#)
- [Configuring the Nodes in the SSR Cluster \(SRC CLI\) on page 404](#)

Configuring the Database Memory Size When Running the Two-Shared-Data-Node Geometry (SRC CLI)

When you run the **two-shared-data-node** geometry, you need to configure the size of the database memory. In this geometry, the memory is shared by all components, unlike the two-data-node and four-data-node geometries, which allocate all available memory to the data node process. The SRC CLI command **show database memory-requirement** can help you estimate the memory size based on the current schema and the maximum number of rows you specify in each table.

Use the following configuration statement to configure the management server on a client node:

```
shared database cluster (primary) nodes database-size {  
    data-memory data-memory;  
    index-memory index-memory;  
}
```

To configure the size of the database memory:

1. (Optional) From configuration mode, access the statement to configure the size of the database memory.

```
user@host# edit shared database cluster primary nodes database-size
```

2. (Optional) Specify data-memory size for each data node.

```
[edit shared database cluster primary nodes database-size]
user@host# set data-memory data-memory
```

3. (Optional) Specify index-memory size for each data node.

```
[edit shared database cluster primary nodes database-size]
user@host# set index-memory index-memory
```

Related Documentation

- [Configuring the Fields in the Subscriber Sessions Table \(SRC CLI\) on page 407](#)
- [Configuring the SSR Cluster Geometry \(SRC CLI\) on page 403](#)
- [SSR Node Types on page 370](#)
- [Configuring the Nodes in the SSR Cluster \(SRC CLI\) on page 404](#)

Configuring the Fields in the Subscriber Sessions Table (SRC CLI)

The format of subscriber identity information stored for each session is managed by the subscriber sessions table, which is controlled by the cluster's client node/management servers. The configuration of the subscriber sessions table is copied to all nodes in the cluster, so all nodes operate with the same information.

The preconfigured schema defines the table's indexes. The default set of database fields in the subscriber sessions table addresses the needs of most carriers, however you can modify certain fields to address unique needs and situations. "[SSR Database Schema](#)" on [page 384](#) describes the default configuration and fields that can be modified.



NOTE: Changes to the SSR database schema requires you to restart all SRC components.

Use the following configuration statements to modify the fields in the subscriber sessions table:

```
shared database cluster primary schema table subscriber-sessions field name {
  type [(int | string | binary)];
  size size;
  require-value;
  indexed;
  default default;
  variable-length;
}
```

To configure the subscriber sessions table:

1. (Optional) From configuration mode, access the statement to configure the fields in the subscriber sessions table. For example, to add a new field called DeviceName:

```
user@host# edit shared database cluster primary schema table subscriber-sessions  
field name
```

2. (Optional) Specify the field type. Values for type correspond to legal SQL data types. For example, to specify the new field as a text string:

```
[edit shared database cluster primary schema table subscriber-sessions field name]  
user@host# set type string
```

3. (Optional) Specify the field size in bytes. For example, to specify the field size as two bytes:

```
[edit shared database cluster primary schema table subscriber-sessions field name]  
user@host# set size 2
```

4. (Optional) Specify whether null values are allowed for the value of the field. For example, to specify that a value is required:

```
[edit shared database cluster primary schema table subscriber-sessions field name]  
user@host# set require-value
```

5. (Optional) Specify whether you want the table to be indexed by this field. For example, to specify that the table is indexed by this field:

```
[edit shared database cluster primary schema table subscriber-sessions field name]  
user@host# set indexed
```

6. (Optional) Specify the default value for the field corresponding to the type. For example, to specify the default as nas123:

```
[edit shared database cluster primary schema table subscriber-sessions field name]  
user@host# set default nas123
```

7. (Optional) If the type is specified as either binary or string, specify whether the value is variable. If set, the SQL data type will be varbinary or varchar, respectively. For example, to specify the type as variable:

```
[edit shared database cluster primary schema table subscriber-sessions field name]  
user@host# set variable-length
```

Related Documentation

- [SSR Database Schema on page 384](#)
- [Making Modifications to the SSR Database Schema Overview on page 388](#)
- [Creating the SSR Database \(SRC CLI\) on page 420](#)
- [Deleting the SSR Database \(SRC CLI\) on page 420](#)

Mapping SAE Plug-In Attributes to Fields in the Subscriber Sessions Table (SRC CLI)



NOTE: Any fields in the subscriber sessions table that have a “not null” requirement must be mapped to either a request attribute or variable.

Use the following configuration statements to map an attribute in the subscriber sessions table to an SAE plugin attribute:

```
shared database cluster (primary) attribute-associations entity name
  shared database cluster primary attribute-associations table name field name {
    sae-plugin-attribute sae-plugin-attribute ;
  }
```

The following procedure can be performed from any node.

To map an attribute in the subscriber sessions table to an SAE plugin attribute:

1. (Optional) From configuration mode, access the configuration statement that configures the name of the table in the SSR database to which you want to make an association. For example, to map the subscriber sessions table attribute VpnID to the SAE plugin attribute vpn-id:

```
user@host# edit shared database cluster primary attribute-associations entity
subscriber-sessions
```

2. Specify the name of the field in the subscriber sessions table and the SAE plugin attribute.

```
[edit shared database cluster primary attribute-associations entity subscriber-sessions]
user@host# set field VpnID sae-plugin-attribute vpn-id
```

3. Commit the configuration.

```
user@host# commit
```

Related Documentation

- [SSR Database Schema on page 384](#)
- [Configuring the Fields in the Subscriber Sessions Table \(SRC CLI\) on page 407](#)

Modifying the SSR Database Schema in an Active Cluster (SRC CLI)

The default database schema is sufficient for most carrier environments. However, if you need to modify the default schema, you need to modify the fields in the subscriber sessions table, re-create the database, and apply the new schema. See [“SSR Database Schema” on page 384](#) for a description of the default database schema and which fields can be modified.



CAUTION: The following procedure requires you to re-create the SSR database. This is a destructive process that deletes all information in the database. This procedure should be performed only during a maintenance window. Review this procedure in full before proceeding. In addition, because the steps required to re-create the SSR database can be performed only from a client node, we recommend that you perform this procedure from a client node.



NOTE: This procedure assumes that you are working with an active cluster and that the SSR database is in production mode. If you are unsure what mode the database is in, see [“Viewing the SSR Database Mode \(SRC CLI\)” on page 423](#).



NOTE: Changes to the SSR database schema requires you to restart all SRC components.

To modify the SSR database schema in an active cluster:

1. Modify the fields in the subscriber sessions table.
See [“Configuring the Fields in the Subscriber Sessions Table \(SRC CLI\)” on page 407](#).
2. Commit the configuration.
See *Committing a Configuration and Exiting Configuration Mode*.
3. Place the database into maintenance mode.
See [“Placing the SSR Database into Maintenance Mode \(SRC CLI\)” on page 419](#).
4. Delete the existing database.
See [“Deleting the SSR Database \(SRC CLI\)” on page 420](#).
5. Create the database with the modified fields.
See [“Creating the SSR Database \(SRC CLI\)” on page 420](#).
6. Verify the cluster configuration changes by viewing the status of the database.
See [“Viewing the Status of the SSR Cluster \(SRC CLI\)” on page 423](#).

7. Restart all SSR client components in the cluster.
See [Enabling SRC Components](#).
8. Place the new database into production mode.
See [“Placing the SSR Database into Production Mode \(SRC CLI\)” on page 419](#).

**Related
Documentation**

- [SSR Database Schema on page 384](#)
- [Configuration Changes and Their Impact on the SSR Cluster on page 399](#)
- [Making Modifications to the SSR Database Schema Overview on page 388](#)

Modifying Attribute Mapping in an Active SSR Cluster (SRC CLI)



NOTE: This procedure assumes that you are working with an active cluster and that the SSR database is in production mode. If you are unsure what mode the database is in, see [“Viewing the SSR Database Mode \(SRC CLI\)” on page 423](#).



NOTE: Any fields in the subscriber sessions table that have a “not null” requirement must be mapped to either a request attribute or variable.

This procedure can be performed from any node in the SSR cluster. To modify the SSR client component attribute associations in an active cluster:

1. Modify the attribute associations.
See [“Mapping SAE Plug-In Attributes to Fields in the Subscriber Sessions Table \(SRC CLI\)” on page 409](#).
2. Commit the configuration.
See [Committing a Configuration and Exiting Configuration Mode](#).
3. Place the database into maintenance mode.
See [“Placing the SSR Database into Maintenance Mode \(SRC CLI\)” on page 419](#).
4. Verify the cluster configuration changes by viewing the status of the database.
See [“Viewing the Status of the SSR Cluster \(SRC CLI\)” on page 423](#).
5. Place the database into production mode.
See [“Placing the SSR Database into Production Mode \(SRC CLI\)” on page 419](#).

**Related
Documentation**

- [SSR Database Schema on page 384](#)
- [Mapping SAE Plug-In Attributes to Fields in the Subscriber Sessions Table \(SRC CLI\) on page 409](#)

Adding Data Nodes to an Active SSR Cluster (SRC CLI)



NOTE: This procedure makes the following assumptions:

- You are working with an active cluster and the SSR database is in production mode. If you are unsure what mode the database is in, see [“Viewing the SSR Database Mode \(SRC CLI\)” on page 423](#)
- Your SSR cluster currently only has two data nodes.
- You have physically installed the two additional C Series Controllers that will be acting as data nodes, made the interface connections to the cluster, and installed the SRC software.



CAUTION: Changing the cluster geometry causes the SSR database to be destroyed. All data will be lost and you need to re-create the database. This procedure should be performed only during a maintenance window. Review this procedure in full before proceeding.

To add data nodes to an active cluster:

1. Configure each of the C Series Controllers as data nodes.
See [“Configuring the Nodes in the SSR Cluster \(SRC CLI\)” on page 404](#).
2. Change the SSR cluster geometry to *four-data-node*.
See [“Configuring the SSR Cluster Geometry \(SRC CLI\)” on page 403](#).
3. Commit the configuration.
See *Committing a Configuration and Exiting Configuration Mode*.
4. Place the database into maintenance mode.
See [“Placing the SSR Database into Maintenance Mode \(SRC CLI\)” on page 419](#).
5. Disable the component database on each management server in the cluster (one by one).
See [“Disabling the SSR Database \(SRC CLI\)” on page 421](#).
6. Enable the component database on each management server in the cluster (one by one).
See [“Enabling the SSR Database \(SRC CLI\)” on page 421](#).
7. Restart the component database on each of the existing data nodes (one by one).
See [“Restarting the SSR Database \(SRC CLI\)” on page 421](#).
8. Enable the component database on each of the new data nodes (one by one).
See [“Enabling the SSR Database \(SRC CLI\)” on page 421](#).

9. Verify the cluster configuration changes by viewing the status of the database.

See [“Viewing the Status of the SSR Cluster \(SRC CLI\)” on page 423.](#)

10. re-create the SSR database.

See [“Creating the SSR Database \(SRC CLI\)” on page 420.](#)

11. Restart all SSR client components.

See *Enabling SRC Components*

12. Place the database into production mode.

See [“Placing the SSR Database into Production Mode \(SRC CLI\)” on page 419.](#)

Adding Client Nodes to an Active SSR Cluster (SRC CLI)



NOTE: This procedure makes the following assumptions:

- You are working with an active cluster and the SSR database is in production mode. If you are unsure what mode the database is in, see [“Viewing the SSR Database Mode \(SRC CLI\)” on page 423.](#)
- You have physically installed the new C Series Controllers that will be acting as a client node, made the interface connections to the cluster, and installed the SRC software.



CAUTION: This procedure requires you to restart the management servers in the cluster, which is a disruptive process. This procedure should be performed only during a maintenance window. Review this procedure in full before proceeding.

To add a client node to an active cluster:

1. Configure the C Series Controller as a client node.

See [“Configuring the Nodes in the SSR Cluster \(SRC CLI\)” on page 404.](#)

2. (Optional) Configure the new client node to host a management server.

See [“Configuring the Management Servers in the SSR Cluster \(SRC CLI\)” on page 406.](#)

3. Commit the configuration.

See *Committing a Configuration and Exiting Configuration Mode.*

4. Place the database into maintenance mode.

See [“Placing the SSR Database into Maintenance Mode \(SRC CLI\)” on page 419.](#)

5. Disable the component database on each management server in the cluster (one by one).

See [“Disabling the SSR Database \(SRC CLI\)” on page 421.](#)

6. Enable the component database on each management server in the cluster (one by one).

See [“Enabling the SSR Database \(SRC CLI\)” on page 421](#).

7. (Optional) If the new client node is hosting a management server, you need to restart the component database and all SSR client components on all client nodes in the cluster (one by one). If the new client node is not hosting a management server, this step is not necessary.

See [“Restarting the SSR Database \(SRC CLI\)” on page 421](#).

See [Enabling SRC Components](#).

8. Enable the component database on the new client node. If you have added more than one client node, perform this step on each new client node (one by one).

See [“Enabling the SSR Database \(SRC CLI\)” on page 421](#).

9. Verify the cluster configuration changes by viewing the status of the database.

See [“Viewing the Status of the SSR Cluster \(SRC CLI\)” on page 423](#).

10. Place the database into production mode.

See [“Placing the SSR Database into Production Mode \(SRC CLI\)” on page 419](#).

Adding a Management Server to an Active SSR Cluster



NOTE: This procedure makes the following assumptions:

- You are working with an active cluster and the SSR database is in production mode. If you are unsure what mode the database is in, see [“Viewing the SSR Database Mode \(SRC CLI\)” on page 423](#).
- The client node is already installed and configured. If you also need to install the client node, use the procedure for [“Adding Client Nodes to an Active SSR Cluster \(SRC CLI\)” on page 413](#).



CAUTION: This procedure requires you to restart the component database on all client nodes and management servers in the cluster, which is a disruptive process. This procedure should be performed only during a maintenance window. Review this procedure in full before proceeding.

To add a management server to a client node in an active cluster:

1. Select the client node you want to host the new management server, and configure the management server.

See [“Configuring the Management Servers in the SSR Cluster \(SRC CLI\)” on page 406](#).

2. Commit the configuration.

See *Committing a Configuration and Exiting Configuration Mode*.

3. Place the database into maintenance mode.

See [“Placing the SSR Database into Maintenance Mode \(SRC CLI\)” on page 419](#).

4. Disable the component database on all management servers in the cluster (one by one).

See [“Disabling the SSR Database \(SRC CLI\)” on page 421](#).

5. Enable the component database on all management servers in the cluster (one by one).

See [“Enabling the SSR Database \(SRC CLI\)” on page 421](#).

6. Restart the component database and all SSR client components on all client nodes in the cluster (one by one).

See [“Restarting the SSR Database \(SRC CLI\)” on page 421](#).

See *Enabling SRC Components*.

7. Verify the cluster configuration changes by viewing the status of the database.

See [“Viewing the Status of the SSR Cluster \(SRC CLI\)” on page 423](#).

8. Place the database into production mode.

See [“Placing the SSR Database into Production Mode \(SRC CLI\)” on page 419](#).

Removing Data Nodes from an Active SSR Cluster



NOTE: This procedure makes the following assumptions:

- You are working with an active cluster and the SSR database is in production mode. If you are unsure what mode the database is in, see [“Viewing the SSR Database Mode \(SRC CLI\)” on page 423](#).
- Your cluster currently contains four data nodes and you want to change to a two-data-node cluster. Data nodes can be added or removed only in pairs.



CAUTION: Changing the SSR cluster geometry (adding or removing data nodes) destroys all data in the SSR database. All data is lost. This procedure should be performed only during a maintenance window. Review this procedure in full before proceeding.

To remove two data nodes from an active cluster:

1. Reconfigure the cluster topology to a two-data-node geometry.
See [“Configuring the SSR Cluster Geometry \(SRC CLI\)” on page 403.](#)
2. Commit the configuration.
See *Committing a Configuration and Exiting Configuration Mode.*
3. Place the database into maintenance mode.
See [“Placing the SSR Database into Maintenance Mode \(SRC CLI\)” on page 419.](#)
4. Disable the component database on all management servers in the cluster (one by one).
See [“Disabling the SSR Database \(SRC CLI\)” on page 421.](#)
5. Disable the component database on the two data nodes you removed from the cluster (one by one).
See [“Disabling the SSR Database \(SRC CLI\)” on page 421.](#)
6. Enable the component database on all management servers in the cluster (one by one).
See [“Enabling the SSR Database \(SRC CLI\)” on page 421.](#)
7. Restart the component database on the two remaining data nodes in the cluster (one by one).
See [“Restarting the SSR Database \(SRC CLI\)” on page 421.](#)
8. re-create the SSR database.
See [“Creating the SSR Database \(SRC CLI\)” on page 420.](#)
9. Restart all SSR client components.
See *Enabling SRC Components*
10. Verify the cluster configuration changes by viewing the status of the database.
See [“Viewing the Status of the SSR Cluster \(SRC CLI\)” on page 423.](#)
11. Place the database into production mode.
See [“Placing the SSR Database into Production Mode \(SRC CLI\)” on page 419.](#)

Removing a Client Node from an Active SSR Cluster



NOTE: This procedure makes the following assumption:

- You are working with an active cluster and the SSR database is in production mode. If you are unsure what mode the database is in, see [“Viewing the SSR Database Mode \(SRC CLI\)” on page 423](#).



CAUTION: This procedure requires you to restart the component database on all management servers in the cluster, which is a disruptive process. This procedure should be performed only during a maintenance window. Review this procedure in full before proceeding.

To remove a client node from an active cluster:

1. Place the database into maintenance mode.
See [“Placing the SSR Database into Maintenance Mode \(SRC CLI\)” on page 419](#).
2. Disable the component database on all management servers in the cluster (one by one).
See [“Disabling the SSR Database \(SRC CLI\)” on page 421](#).
3. Disable the component database on the client node you removed from the cluster.
See [“Disabling the SSR Database \(SRC CLI\)” on page 421](#).
4. Enable the component database on all management servers in the cluster (one by one).
See [“Enabling the SSR Database \(SRC CLI\)” on page 421](#).
5. Restart the SSR component database on all client nodes.
See [“Enabling the SSR Database \(SRC CLI\)” on page 421](#).
6. Restart all SSR client components.
See [Enabling SRC Components](#).
7. Verify the cluster configuration changes by viewing the status of the database.
See [“Viewing the Status of the SSR Cluster \(SRC CLI\)” on page 423](#).
8. Place the database into production mode.
See [“Placing the SSR Database into Production Mode \(SRC CLI\)” on page 419](#).

Removing a Management Server from an Active SSR Cluster



NOTE: This procedure makes the following assumptions:

- You are working with an active cluster and the SSR database is in production mode. If you are unsure what mode the database is in, see [“Viewing the SSR Database Mode \(SRC CLI\)” on page 423](#).
- The client node that is hosting the management server is installed, configured and active, and you want it to remain in the cluster as a client node only.



CAUTION: This procedure requires you to restart the component database on all management servers in the cluster, which is a disruptive process. This procedure should be performed only during a maintenance window. Review this procedure in full before proceeding.

To remove a management server from a client node in an active cluster:

1. Select the client node from which you want to remove the management server, and reconfigure the node as a client node without a management server.

See [“Configuring the Nodes in the SSR Cluster \(SRC CLI\)” on page 404](#).

2. Commit the configuration.

See *Committing a Configuration and Exiting Configuration Mode*.

3. Place the database into maintenance mode.

See [“Placing the SSR Database into Maintenance Mode \(SRC CLI\)” on page 419](#).

4. Disable the component database on all management servers in the cluster (one by one).

See [“Disabling the SSR Database \(SRC CLI\)” on page 421](#).

5. Enable the component database on all management servers in the cluster (one by one).

See [“Enabling the SSR Database \(SRC CLI\)” on page 421](#).

6. Verify the cluster configuration changes by viewing the status of the database.

See [“Viewing the Status of the SSR Cluster \(SRC CLI\)” on page 423](#).

7. Place the database into production mode.

See [“Placing the SSR Database into Production Mode \(SRC CLI\)” on page 419](#).

CHAPTER 30

Managing the SSR Cluster

- [Placing the SSR Database into Maintenance Mode \(SRC CLI\) on page 419](#)
- [Placing the SSR Database into Production Mode \(SRC CLI\) on page 419](#)
- [Deleting the SSR Database \(SRC CLI\) on page 420](#)
- [Creating the SSR Database \(SRC CLI\) on page 420](#)
- [Enabling the SSR Database \(SRC CLI\) on page 421](#)
- [Disabling the SSR Database \(SRC CLI\) on page 421](#)
- [Restarting the SSR Database \(SRC CLI\) on page 421](#)
- [Deleting All Subscriber Sessions in the SSR Database on page 422](#)
- [Deleting Subscriber Sessions in the SSR Database By IP Address on page 422](#)

Placing the SSR Database into Maintenance Mode (SRC CLI)

- To place the SSR database into maintenance mode, from operational mode:
`user@host> request database enter maintenance-mode`

Related Documentation

- [SSR Database Operating Modes on page 389](#)
- [Placing the SSR Database into Production Mode \(SRC CLI\) on page 419](#)

Placing the SSR Database into Production Mode (SRC CLI)

- To place the SSR database into production mode, from operational mode:
`user@host> request database enter production-mode`

Related Documentation

- [SSR Database Operating Modes on page 389](#)
- [Placing the SSR Database into Maintenance Mode \(SRC CLI\) on page 419](#)

Deleting the SSR Database (SRC CLI)



CAUTION: This command destroys the SSR database and all its contents and should be performed only during a maintenance window.



NOTE: This command must be executed from a client node and can be executed only when the database is in maintenance mode.

- From operational mode, destroy the database.

```
user@host> request database delete database
```

Related Documentation

- [SSR Database Schema on page 384](#)
- [Making Modifications to the SSR Database Schema Overview on page 388](#)
- [Creating the SSR Database \(SRC CLI\) on page 420](#)

Creating the SSR Database (SRC CLI)

This procedure generates the database schema using the current configuration and then creates the SSR database and its tables.



NOTE: All cluster components must have been enabled before you create the database.



NOTE: This command must be executed from a client node and can be executed only when the database is in maintenance mode.

- From operational mode, re-create the database using the modified schema.

```
user@host> request database create database
```

Related Documentation

- [Enabling the SSR Database \(SRC CLI\) on page 421](#)
- [Distributing the SSR Cluster Configuration and Enabling SSR Client Components on page 389](#)
- [SSR Database Schema on page 384](#)
- [Making Modifications to the SSR Database Schema Overview on page 388](#)
- [SSR Database Schema on page 384](#)

Enabling the SSR Database (SRC CLI)

This command starts all SSR component processes on the local node. When multiple nodes need to be enabled, you must execute this command on each node, one by one.



NOTE: You must execute this command on each node in the SSR cluster.

- From operational mode, enable the SSR component database.

```
user@host> enable component database
```

Related Documentation

- [Distributing the SSR Cluster Configuration and Enabling SSR Client Components on page 389](#)
- [Juniper Networks Database Overview](#)
- [Creating the SSR Database \(SRC CLI\) on page 420](#)
- [Disabling the SSR Database \(SRC CLI\) on page 421](#)
- [Restarting the SSR Database \(SRC CLI\) on page 421](#)

Disabling the SSR Database (SRC CLI)

This command stops all SSR component processes on the local node. When multiple nodes need to be shut down, you must execute this command on each node one by one.

- From operational mode, disable the SSR component database.

```
user@host> disable component database
```

Related Documentation

- [Distributing the SSR Cluster Configuration and Enabling SSR Client Components on page 389](#)
- [Distributing the SSR Cluster Configuration and Enabling SSR Client Components on page 389](#)
- [Enabling the SSR Database \(SRC CLI\) on page 421](#)
- [Restarting the SSR Database \(SRC CLI\) on page 421](#)

Restarting the SSR Database (SRC CLI)

This command restarts all SSR component processes on the local node. When you need to restart multiple nodes, it must be executed on each node, one by one.

- From operational mode, restart the SSR component database.

```
user@host> restart component database
```

- Related Documentation**
- [Distributing the SSR Cluster Configuration and Enabling SSR Client Components on page 389](#)
 - [Enabling the SSR Database \(SRC CLI\) on page 421](#)
 - [Disabling the SSR Database \(SRC CLI\) on page 421](#)

Deleting All Subscriber Sessions in the SSR Database

- From operational mode, delete all subscriber sessions and associated service sessions.
`user@host> request database delete sessions all`

- Related Documentation**
- [Deleting Subscriber Sessions in the SSR Database By IP Address on page 422](#)
 - [Viewing All Subscriber Sessions in the SSR Database \(SRC CLI\) on page 426](#)
 - [Viewing Subscriber Sessions in the SSR Database by IP Address \(SRC CLI\) on page 427](#)
 - [Viewing Subscriber Sessions in the SSR Database by Indexed Field \(SRC CLI\) on page 428](#)

Deleting Subscriber Sessions in the SSR Database By IP Address

1. To delete subscriber sessions based on IP address, from operational mode:
`user@host>request database delete subscriber-sessions by-address start-address start-address end-address end-address vpn-id vpn-id`
2. Enter the starting address of the IP range. If the end address is not specified, only the session matching this address is deleted.
3. (Optional) Enter the ending address of the IP range. If not specified, only the session matching the start address is deleted.
4. (Optional) Enter the VPN ID that the sessions belong to. If not specified, only sessions with public IP addresses are displayed. If value is either " " or ' ', display sessions from public network or any VPN.

- Related Documentation**
- [Deleting All Subscriber Sessions in the SSR Database on page 422](#)
 - [Viewing All Subscriber Sessions in the SSR Database \(SRC CLI\) on page 426](#)
 - [Viewing Subscriber Sessions in the SSR Database by IP Address \(SRC CLI\) on page 427](#)
 - [Viewing Subscriber Sessions in the SSR Database by Indexed Field \(SRC CLI\) on page 428](#)

CHAPTER 31

Monitoring the SSR Cluster

- [Viewing the SSR Database Mode \(SRC CLI\) on page 423](#)
- [Viewing the Status of the SSR Cluster \(SRC CLI\) on page 423](#)
- [Viewing the Database Memory Requirements \(SRC CLI\) on page 424](#)
- [Viewing the Running Configuration of the SSR Database \(SRC CLI\) on page 425](#)
- [Viewing All Subscriber Sessions in the SSR Database \(SRC CLI\) on page 426](#)
- [Viewing Subscriber Sessions in the SSR Database by IP Address \(SRC CLI\) on page 427](#)
- [Viewing Subscriber Sessions in the SSR Database by Indexed Field \(SRC CLI\) on page 428](#)
- [Viewing the Total Number of Subscriber Sessions in the SSR Database \(SRC CLI\) on page 430](#)

Viewing the SSR Database Mode (SRC CLI)

Purpose	View the current operating mode of the SSR database (maintenance or production mode). This command can be executed from any node in the cluster.
Action	<pre>user@host> show database mode Database is in maintenance-mode</pre>
Related Documentation	<ul style="list-style-type: none">• Placing the SSR Database into Maintenance Mode (SRC CLI) on page 419• Placing the SSR Database into Production Mode (SRC CLI) on page 419

Viewing the Status of the SSR Cluster (SRC CLI)

Purpose	View the status of the SSR database. The IP address and the connection status for each node in the cluster is displayed. This command displays all configured data nodes and management servers, regardless of whether they are connected. However, only client nodes that are connected are displayed.
----------------	---



NOTE: This command must be issued on a client node with a management server.

To interpret the status, make sure that you have a good understanding of the SSR concepts and terminology.

See [“Session State Registrar Overview” on page 369](#) and [“SSR Node Types” on page 370](#).

Action user@host> **show database status**

```
Data Nodes
Data Node
Address 10.227.6.49
Node ID 1
Connected Yes
Status Nodegroup: 0, Master

Management Servers
Management Server
Address 10.227.6.49
Node ID 16
Connected Yes

Connected Client Nodes
Client Node
Address 10.227.6.49
Node ID 17
Component Name mysqld

Client Node
Address 10.227.6.49
Node ID 21
Component Name DSA
```

- Related Documentation**
- [SSR Node Types on page 370](#)
 - [SSR Cluster Configurations Overview on page 373](#)
 - [SSR Node Groups on page 371](#)
 - [Supported SSR Cluster Configurations on page 377](#)
 - [SSR Database Schema on page 384](#)

Viewing the Database Memory Requirements (SRC CLI)

Purpose View the SSR database memory requirements. Use this command to calculate the required memory size based on the defined schema and the projected maximum number of rows to be stored in each table.

Action To view information about the database memory requirements:

```
user@host> show database memory-requirement table (subscriber-sessions |
service-sessions) maximum-number-of-rows maximum-number-of-rows
use-running-configuration use-running-configuration
```

Set the **use-running-configuration** option to use the schema defined in running configuration to calculate the table size. Otherwise, use the schema in the current CLI configuration

To view information about the subscriber sessions table, specify the subscriber sessions table and the maximum projected number of rows to be stored in the table. For example:

```
user@host> show database memory-requirement table subscriber-sessions
maximum-number-of-rows 1000000
```

To view information about the service sessions table, specify the service sessions table and the maximum projected number of rows to be stored in the table. For example:

```
user@host> show database memory-requirement table service-sessions
maximum-number-of-rows 1000000
```

The output displays the minimum data memory, maximum data memory, and index memory. The minimum data memory is the amount of data memory required if all variable length columns are empty. The maximum data memory is the amount of data memory required if all variable length columns are in their full length.

```
root@host> show database memory-requirement table subscriber-sessions
maximum-number-of-rows 1000000
Memory Requirement
Table Name          subscriber-sessions
Minimum Data Memory (MB) 398
Maximum Data Memory (MB) 608
Index Memory (MB)      77
```

Related Documentation

- [Configuring the Database Memory Size When Running the Two-Shared-Data-Node Geometry \(SRC CLI\) on page 406](#)
- [SSR Node Types on page 370](#)
- [Configuring the Fields in the Subscriber Sessions Table \(SRC CLI\) on page 407](#)
- [Supported SSR Cluster Configurations on page 377](#)
- [SSR Database Schema on page 384](#)

Viewing the Running Configuration of the SSR Database (SRC CLI)

Purpose View the database running configuration. When SSR database is in production mode, displays configuration running on SSR database. When SSR database is placed in maintenance mode, the running configuration is discarded and the SSR database running configuration is the configuration you entered in the SRC CLI. This command can be executed only on management node.

Action user@host> **show database running-configuration**

Data Nodes

Data Node

Address 10.227.6.49

Node ID 1

Connected Yes

Status Nodegroup: 0, Master

Management Servers

Management Server

Address 10.227.6.49

Node ID 16

Connected Yes

Connected Client Nodes

Client Node

Address 10.227.6.49

Node ID 17

Component Name mysqld

Client Node

Address 10.227.6.49

Node ID 21

Component Name DSA

- Related Documentation**
- [SSR Node Types on page 370](#)
 - [SSR Cluster Configurations Overview on page 373](#)
 - [SSR Node Groups on page 371](#)
 - [Supported SSR Cluster Configurations on page 377](#)
 - [SSR Database Schema on page 384](#)

Viewing All Subscriber Sessions in the SSR Database (SRC CLI)

Purpose View all the subscriber sessions and associated service sessions stored in the SSR database. The command only shows attributes that are not null.



NOTE: This command must be executed on a client node with or without a management server.

Enter the maximum number of sessions to display. By default the maximum number of sessions displayed is 25.

Action user@host> **show database subscriber-sessions maximum-results** *maximum-result*

```
UserIPAddress  0.0.0.1
VpnID
SessionID 00-12-56-8C-00-CD
UserName      abc
```

```
Service session
Subscription Name test1
SessionName      DEFAULT
```

```
Subscriber session
UserIPAddress  0.0.0.2
VpnID
SessionID 00-23-56-8C-00-CD
UserName      abc
```

```
Service session
Subscription Name test1
SessionName      DEFAULT
```

```
Subscriber session
UserIPAddress  0.0.0.3
VpnID
SessionID 00-43-56-8C-00-CD
UserName      abc
```

```
Subscriber session
UserIPAddress  0.0.0.4
VpnID
SessionID 00-65-56-8C-00-CD
UserName      abc
```

- Related Documentation**
- [Viewing Subscriber Sessions in the SSR Database by IP Address \(SRC CLI\) on page 427](#)
 - [Viewing Subscriber Sessions in the SSR Database by Indexed Field \(SRC CLI\) on page 428](#)
 - [Viewing the Total Number of Subscriber Sessions in the SSR Database \(SRC CLI\) on page 430](#)

Viewing Subscriber Sessions in the SSR Database by IP Address (SRC CLI)

Purpose View subscriber sessions and associated service sessions stored in the SSR database by IP address.



NOTE: This command must be executed on a client node.

Enter the start address of the IP range. If you do not specify the **end-address**, only the session with this IP address is displayed.

(Optional) Enter the end address of an IP range. If not specified, only the session that matches **start-address** is displayed.

(Optional) Enter the VPN ID that the sessions belong to. If not specified, only sessions with public IP addresses are displayed. If value is either " " or ' ', display sessions from public network or any VPN.

Enter the maximum number of sessions to display. By default the maximum number of sessions displayed is 25.

Action `user@host> show database subscriber-sessions by-address start-address start-address end-address end-address vpn-id vpn-id maximum-results maximum-result`

```
UserIPAddress    0.0.0.1
VpnID
SessionID 00-50-56-8C-00-CD
UserName      abc
```

```
Service session
Subscription Name test1
SessionName      DEFAULT
```

```
Subscriber session
UserIPAddress    0.0.0.2
VpnID
SessionID 00-80-67-8D-00-CA
UserName      abc
```

```
Service session
Subscription Name test1
SessionName      DEFAULT
```

```
Subscriber session
UserIPAddress    0.0.0.3
VpnID
SessionID 00-13-56-8C-60-12
UserName      abc
```

```
Subscriber session
UserIPAddress    0.0.0.4
VpnID
SessionID 00-50-56-8C-00-CD
UserName      abc
```

- Related Documentation**
- [Viewing Subscriber Sessions in the SSR Database by Indexed Field \(SRC CLI\) on page 428](#)
 - [Viewing All Subscriber Sessions in the SSR Database \(SRC CLI\) on page 426](#)
 - [Viewing the Total Number of Subscriber Sessions in the SSR Database \(SRC CLI\) on page 430](#)

Viewing Subscriber Sessions in the SSR Database by Indexed Field (SRC CLI)

Purpose View subscriber sessions and associated service sessions stored in the SSR database based on an indexed field in the subscriber sessions table.

Enter the name of the indexed field in the subscriber sessions table.

Enter the value for the indexed field. For fields of integer or binary type, sessions that match the selected indexed field specified value are displayed. For fields of string type, the wildcard '*' can be used as the value. "*" matches any number of characters. However, the value cannot start with a wildcard, such as "*", or "*abc".

Enter the maximum number of sessions to display. By default the maximum number of sessions displayed is 25.



NOTE: This command can be executed from any client node.

Action `user@host> show database subscriber-sessions by-indexed-field name indexed-field-name value value maximum-results maximum-result`

```
UserIPAddress    0.0.0.1
VpnID
SessionID 11-34-74-8A-66-7B
UserName         abc
```

```
Service session
Subscription Name test1
SessionName      DEFAULT
```

```
Subscriber session
UserIPAddress    0.0.0.2
VpnID
SessionID 00-76-42-AA-87-8C
UserName         abc
```

```
Service session
Subscription Name test1
SessionName      DEFAULT
```

```
Subscriber session
UserIPAddress    0.0.0.3
VpnID
SessionID 54-34-12-8A-7B-CD
UserName         abc
```

```
Subscriber session
UserIPAddress    0.0.0.4
VpnID
SessionID 21-45-6B-67-4A-6C
UserName         abc
```

- Related Documentation**
- [Viewing Subscriber Sessions in the SSR Database by IP Address \(SRC CLI\) on page 427](#)
 - [Viewing All Subscriber Sessions in the SSR Database \(SRC CLI\) on page 426](#)
 - [Viewing the Total Number of Subscriber Sessions in the SSR Database \(SRC CLI\) on page 430](#)

Viewing the Total Number of Subscriber Sessions in the SSR Database (SRC CLI)

Purpose View the total number of subscriber sessions currently in the SSR database.



NOTE: This command must be executed on a client node.

Action user@host> show database subscriber-sessions count

```
root@igor> show database subscriber-sessions count
Number of Subscriber Sessions: 4
```

- Related Documentation**
- [Viewing Subscriber Sessions in the SSR Database by IP Address \(SRC CLI\) on page 427](#)
 - [Viewing Subscriber Sessions in the SSR Database by Indexed Field \(SRC CLI\) on page 428](#)
 - [Viewing All Subscriber Sessions in the SSR Database \(SRC CLI\) on page 426](#)

PART 8

Using the Subscriber Information Collector

- [Overview of the Subscriber Information Collector on page 433](#)
- [Configuring the Subscriber Information Collector with the SRC CLI on page 471](#)
- [Device and Service Templates for Dynamic Authorization \(SRC CLI\) on page 539](#)
- [Monitoring the Subscriber Information Collector with the SRC CLI on page 585](#)

CHAPTER 32

Overview of the Subscriber Information Collector

- [Subscriber Information Collector Overview on page 433](#)
- [Managing Dynamic Services on page 434](#)
- [SIC Dynamic Authorization Support Overview on page 435](#)
- [How the Dynamic Authorization Process Works in the SIC on page 437](#)
- [Dynamic Authorization Targets \(SRC CLI\) on page 441](#)
- [RADIUS Authentication/Authorization and Accounting Data Flow on page 441](#)
- [Local and Shared Configurations for the SIC \(SRC CLI\) on page 445](#)
- [Accounting Methods and Targets \(SRC CLI\) on page 446](#)
- [Authentication Route Targets \(SRC CLI\) on page 450](#)
- [Request Routing \(SRC CLI\) on page 450](#)
- [SIC Editing Rules \(SRC CLI\) on page 453](#)
- [RADIUS and Diameter Configuration for the SIC Overview \(SRC CLI\) on page 456](#)
- [Failover Policy on page 460](#)
- [RADIUS and Diameter Transports on page 462](#)
- [SIC Dictionaries and Device Models Overview \(SRC CLI\) on page 463](#)
- [SIC Local Realms Overview on page 464](#)
- [SIC Event Logging Overview \(SRC CLI\) on page 464](#)
- [SNMP Support for the SIC Overview \(SRC CLI\) on page 467](#)

Subscriber Information Collector Overview

The subscriber information collector (SIC) is used in conjunction with the MX Series Ethernet Services Router running the packet-triggered subscribers and policy control (PTSP) solution. The SIC listens for RADIUS accounting and authentication messages from IP edge devices (clients), either directly or indirectly through an authentication, authorization, and accounting (AAA) proxy server, allowing the SRC software to gain increased subscriber awareness.

The role of the SIC is to listen for RADIUS accounting and authentication messages and filter undesired events based on attachment session attributes. The SIC is also responsible for sending RADIUS requests to the correct SRC that is managing the MX Series router.

The major components of the SIC are:

- RADIUS accounting and authentication listeners, which are configured with port numbers and parameters controlling receipt of UDP packets.
- A collection of RADIUS dictionaries.
- A collection of network access server (NAS) accounting and authentication clients.
- A collection of RADIUS accounting, authentication, and dynamic authorization targets.
- A collection of accounting and authentication routing rules.
- A collection of editing rules.
- A collection of RADIUS network elements. A RADIUS network element contains an ordered list of RADIUS accounting and authentication clients, RADIUS accounting, authentication, and dynamic authorization targets, along with a failover policy for targets.
- A collection of accounting methods including storing accounting events in the SRC session state registrar (SSR) database or forwarding them to a downstream AAA server (network element).
- A collection of authentication and dynamic authorization routing targets. Authentication requests are routed to a downstream AAA RADIUS server for authentication. Dynamic authorization requests are routed to the NAS device in the upstream network element.
- An SIC Diameter server that translates dynamic authorization requests (Change of Authorization [COA] and Disconnect Message [DM] requests) into vendor-specific attributes so that they can be understood by the NAS. The SIC Diameter server communicates with the SRC Diameter server for NAS routing information.
- Components supporting SNMP, statistics, and event logging.

**Related
Documentation**

- [Accounting Methods and Targets \(SRC CLI\) on page 446](#)
- [Local and Shared Configurations for the SIC \(SRC CLI\) on page 445](#)
- [Request Routing \(SRC CLI\) on page 450](#)

Managing Dynamic Services

When you integrate the SIC, you can manage services on RADIUS-enabled devices in an SRC network. The SIC processes messages between the NAS device and the RADIUS server. You can configure the services, policies, and parameters with the SRC software independent of the NAS device. The SRC Diameter server communicates with the SIC Diameter server by using Diameter messages to dynamically manage services for a subscriber session. The SIC Diameter server converts the Diameter messages to RADIUS messages and routes dynamic RADIUS requests to the NAS device (client or target), or to the accounting or authentication target.

The SIC Diameter server forwards messages to the SRC Diameter server, which then forwards them to the AAA device driver in the SAE. These Diameter messages perform the following functions:

- AAR—Attach the subscriber to the access network.
- ACR—Provide accounting information.
- ASR—Disconnect the subscriber.
- PPR—Start, modify, or stop the service session; send message routing configuration.
- STR—Detach the subscriber from the access network.

You must configure NAS groups and an AAA device driver for each NAS group hosted by the SAE. You also need to configure the services, policies, and parameters that the SIC uses for service activation on the NAS device. You need to provide specific information for the service templates used by the SIC.

Service templates list the parameters needed for service activation on a NAS device. The SIC has detailed knowledge about the specific NAS device so that it can use the services, policies, and parameters configured by the SRC software for managing services on the NAS device.

Tasks to set up the management of services on RADIUS-enabled devices are:

- Configure the SIC. See [“SIC RADIUS Configuration Summary \(SRC CLI\)” on page 472](#).
- Configure the SRC Diameter application. See *Configuring the Diameter Application (SRC CLI)*.
- Configure the NAS groups. See [“Configuring the NAS Groups \(SRC CLI\)” on page 531](#).
- Configure the SAE to manage AAA devices. See [“Configuring the SAE to Manage AAA Devices” on page 535](#).
- Configure AAA policies. See [“Configuring AAA Policies \(SRC CLI\)” on page 537](#).

**Related
Documentation**

- [SIC Dynamic Authorization Support Overview on page 435](#)
- [How the Dynamic Authorization Process Works in the SIC on page 437](#)
- [Subscriber Information Collector Overview on page 433](#)

SIC Dynamic Authorization Support Overview

The SIC can dynamically manage services on RADIUS-enabled devices. The RADIUS capabilities of the SIC allow the SRC software to be aware of the subscriber activity and make dynamic RADIUS requests using the following RADIUS features:

- Authentication, authorization, and accounting (AAA)
- Change of Authorization (COA) message
- Disconnect Message (DM)

The SIC uses RADIUS AAA messages to communicate with the RADIUS server and the network access server (NAS). The SIC converts Diameter messages to RADIUS messages and vice versa. The SIC also performs conversion between Diameter attribute-value pairs (AVPs) and RADIUS attributes.

The SIC can provide:

- Device abstraction and shared secrets for the NAS device
- Accounting and authentication support for subscriber sessions and service sessions
- COA and DM support
- Service parameter changes

RADIUS was designed as an AAA protocol in client/server mode. Supporting dynamic authorization requests requires that the SIC communicate Change of Authorization (COA) requests and Disconnect Messages (DM) to the network access server (NAS). However, every NAS vendor implements services by using different sets of vendor-specific attributes (VSAs); there is no universal language for sending requests to a NAS. To translate COA or DM requests into the correct dialect, the SIC uses service templates, which define services that the router activates and deactivates. These service templates translate COA or DM requests into VSAs so that the NAS device can understand and implement them. Service templates are created using the SRC CLI and they specify initial authorization, activation, deactivation, and abort session requests.

We provide device templates for Juniper Networks E Series Broadband Services Routers running JunosE Software release 7.2 or later and for Cisco routers running Cisco IOS Release 12.2SB. These templates include sample global and service templates that you can modify for your specific environment. If you want to add a router from another vendor, you must create a new template so that the SRC can communicate properly with your new router.

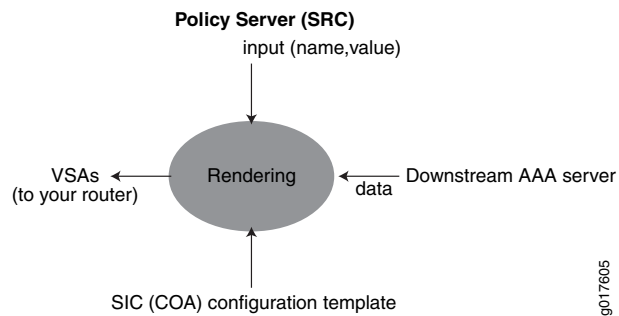
The SIC dynamic authorization function includes:

- RADIUS listeners for authentication and accounting requests.
- RADIUS dynamic authorization interface for sending COA or DM requests to the NAS.
- RADIUS proxy function for forwarding RADIUS authentication and accounting requests to a downstream RADIUS server.
- SIC Diameter server interacts with the SRC Diameter server. User access, accounting requests, and service accounting information are sent to the SAE through this Diameter interface.

Rendering

The SIC generates COA or DM requests on request from the SAE. Translations between SAE, SRC Diameter server, SIC, and your router must take place. This translation process is called rendering. The rendering process is shown in [Figure 64 on page 437](#).

Figure 64: The Rendering Process



The rendering process takes three inputs and produces one output. Inputs are:

- The data the SAE sends (to and from the SRC Diameter server)
- SIC configuration (device and service) templates
- Data that returns with the authentication response from the downstream AAA server (available only for initial authorization process)

**Related
Documentation**

- [How the Dynamic Authorization Process Works in the SIC on page 437](#)
- [Managing Dynamic Services on page 434](#)

How the Dynamic Authorization Process Works in the SIC

This section describes the process of creating device and service templates for dynamic authorization. To understand how service templates interact with service requests, there are three main scenarios that you need to consider:

- Initial Authorization
- Activation and Deactivation
- Abort Session

Each of these has a service template associated with it.



NOTE: In the following discussion and illustrations, the NAS communicates with the SIC through the router.

Introduction

There are two common behaviors that trigger dynamic authorization requests:

- The SIC sends a request to the SAE notifying it about an event, such as authentication success.
- The SAE requests a service, such as activation, deactivation, or abort session.

In the former case, the SAE replies, and the SIC uses this reply as one of the inputs to the rendering process to generate VSAs. In any case, the SAE supplies data that the SIC uses as one of the inputs to the rendering process to generate VSAs. The SIC then sends the VSAs to the NAS so that it can activate or deactivate services.

In the process, requests may go not only from the NAS to the SIC, but also to the downstream AAA server, to the SAE, and, in the case of the initial authorization scenario, from the SIC to the downstream AAA server.

Initial Authorization

Initial authorization of services requires that your NAS support service activation in the Access-Accept message. This capability is called **Initial-Authorization** mode in the service template. This scenario begins when the NAS sends an authorization request to the SIC. The SIC in turn sends a RADIUS access request to the downstream AAA server that handles authorization requests.

If the downstream AAA server approves the request, it sends a RADIUS Access-Accept message to the SIC. Using the global service template configuration, the SIC formats the authorization request to the SAE. At this point, the SAE replies with service activation data used as input to the rendering process. This data contains the service name as specified in the service template along with the attribute values and parameters. For example, if the SAE requests `content_provider_tiered` service, the SIC renders data by using the corresponding mode, as shown in the following example service template configuration:

```
service-template content_provider-tiered {
  mode Initial-Authorization {
    attributes {
      item attr1 {
        parameterized-attribute {
          format
content_provider_tiered($(contentProviderAddress),$(contentProviderMask),
$(subscriberAddress),$(subscriberMask),
$(upstreamBandwidth),$(downstreamBandwidth));
          name Unisphere-Activate-Service;
        }
      }
      item attr2 {
        default-attribute {
          name Unisphere-Service-Stats;
          value 1;
        }
      }
    }
  }
}
```

The SIC renders the Access-Accept message, then informs the SAE about rendering successes and failures in another request. The SAE sends an acknowledgement back to the SIC, which in turn sends a rendered Access-Accept message to the NAS.

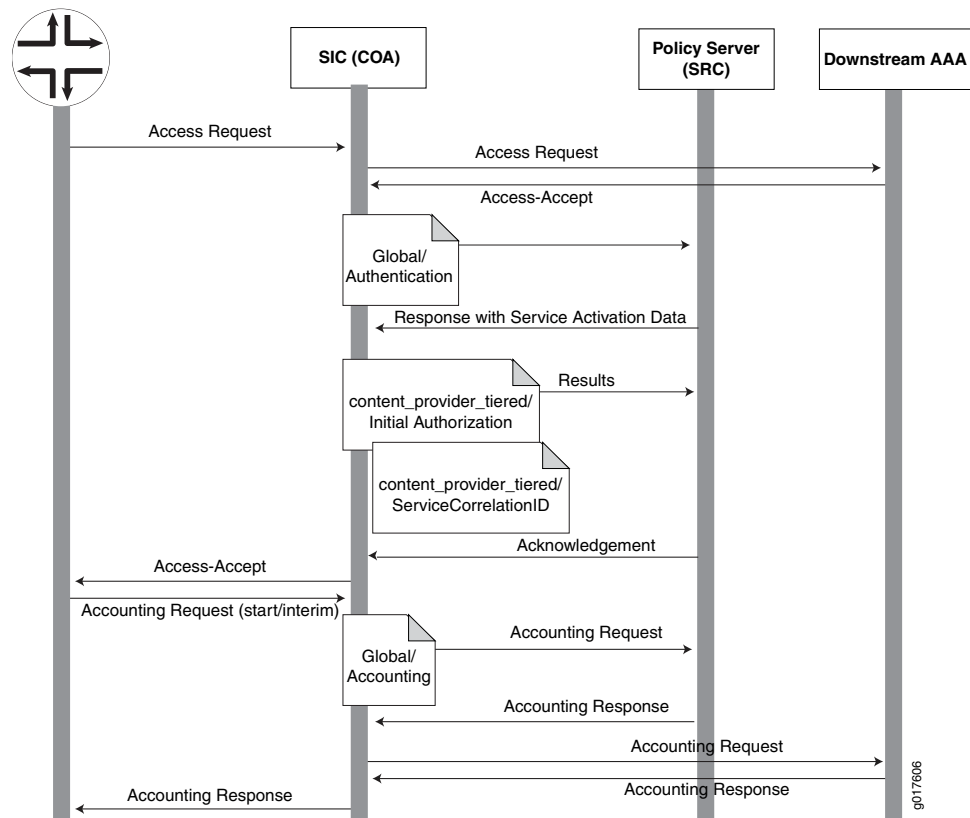
Accounting

As soon as the requested service is active, the next step is sending an accounting (start or interim) request from the NAS to the SIC. Using the rendering process and the

information defined in accounting mode in the global service template, the SIC sends an accounting request to the SAE, which then sends an accounting response. After receiving this response, the SIC sends an accounting request to the downstream AAA server, which sends an accounting response. Finally, the SIC sends an accounting response back to the NAS and service accounting is complete.

Figure 65 on page 439 shows the initial authorization and accounting timing sequence. The rectangles with a folded corner represent pieces of the service or global service templates. For purposes of this illustration, the SIC and SRC are shown in two distinct rectangles.

Figure 65: Initial Authorization and Accounting Timing Sequence



Service Activation and Deactivation

This section describes the service activation and deactivation scenarios. The sequences for activation and deactivation are identical except that the activation sequence uses activation requests and the deactivation sequence uses deactivation requests.

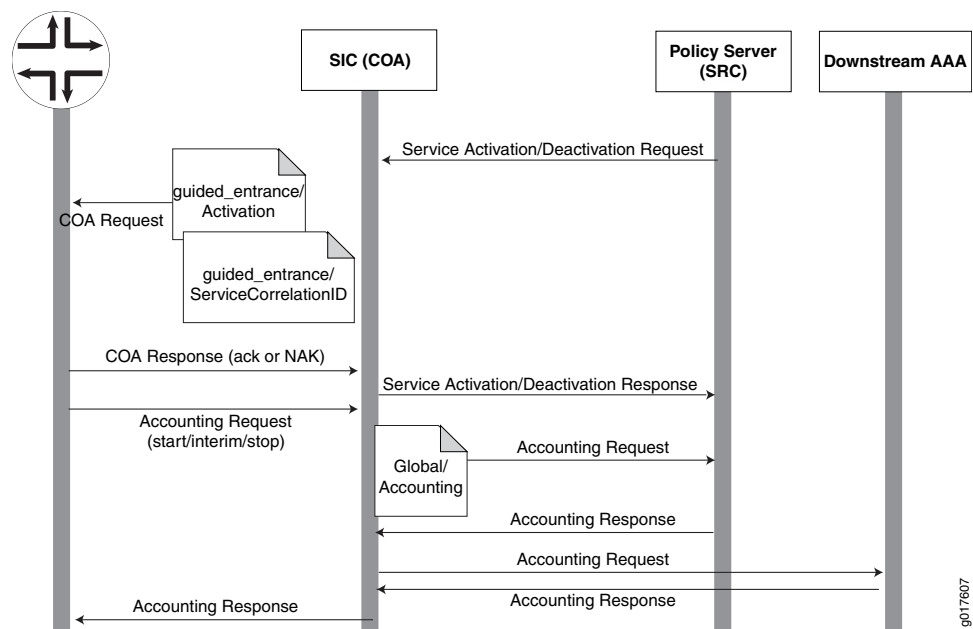
A service activation begins with an activation request from the SAE to the SIC. Figure 66 on page 440 uses the guided_entrance service activation request as an example. This activation request includes all the information needed for the SIC to render the guided_entrance service activation request into RADIUS format for the NAS. The SIC sends the rendered request, along with a service correlation ID, as a COA to the NAS. The

NAS responds with an acknowledgement packet (ack) or negative acknowledgement (NAK). The SIC then sends a service activation response to the SAE.

This completes the service activation. The NAS then initiates an accounting request, the timing sequence of which is identical to the sequence described in [“Accounting” on page 438](#).

[Figure 66 on page 440](#) shows the activation and deactivation timing sequences. For purposes of this illustration, the SIC and SRC are shown in two distinct rectangles.

Figure 66: Activation and Deactivation Timing Sequences



Abort Session Requests

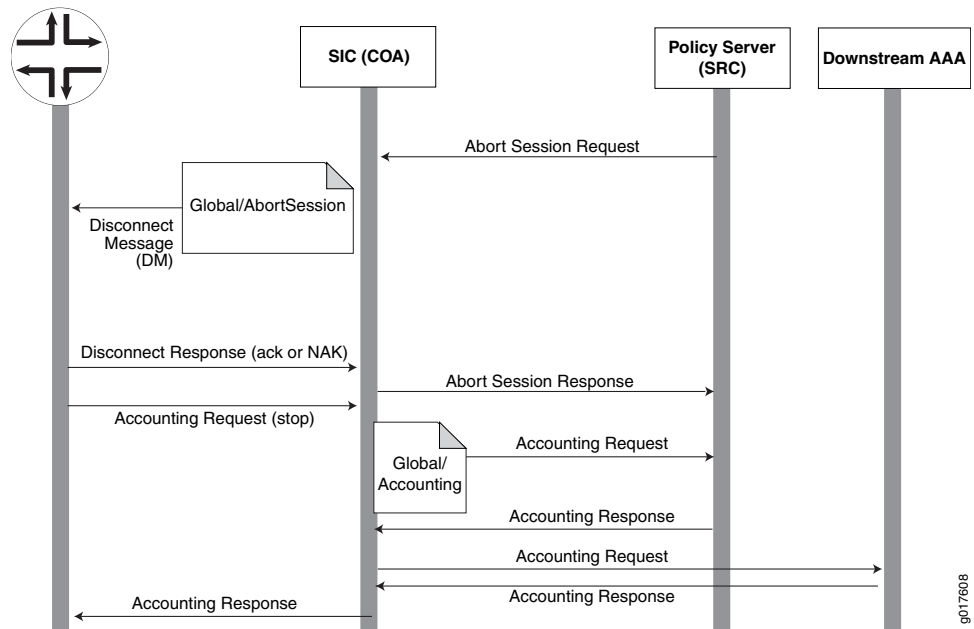
If the SAE receives an abort session request, it sends it to the SIC. The SIC, using the global service template and Abort-Session mode, renders the request and sends it, along with a service correlation ID, as a Disconnect Message (DM) to the NAS. The NAS responds with an ack or NAK. The SIC then sends a response to the SAE.

In all situations, abort session requests follow the same sequence and use the same global service template.

This completes the abort session scenario. The NAS then initiates an accounting request, the timing sequence of which is identical to the sequence described in [“Accounting” on page 438](#).

[Figure 67 on page 441](#) shows the abort session timing sequence. For purposes of this illustration, the SIC and SRC are shown in two distinct rectangles.

Figure 67: Abort Session Timing Sequence



- Related Documentation**
- [SIC Dynamic Authorization Support Overview on page 435](#)
 - [Device and Service Template Configuration Overview \(SRC CLI\) on page 539](#)
 - [Dynamic Authorization Targets \(SRC CLI\) on page 441](#)

Dynamic Authorization Targets (SRC CLI)

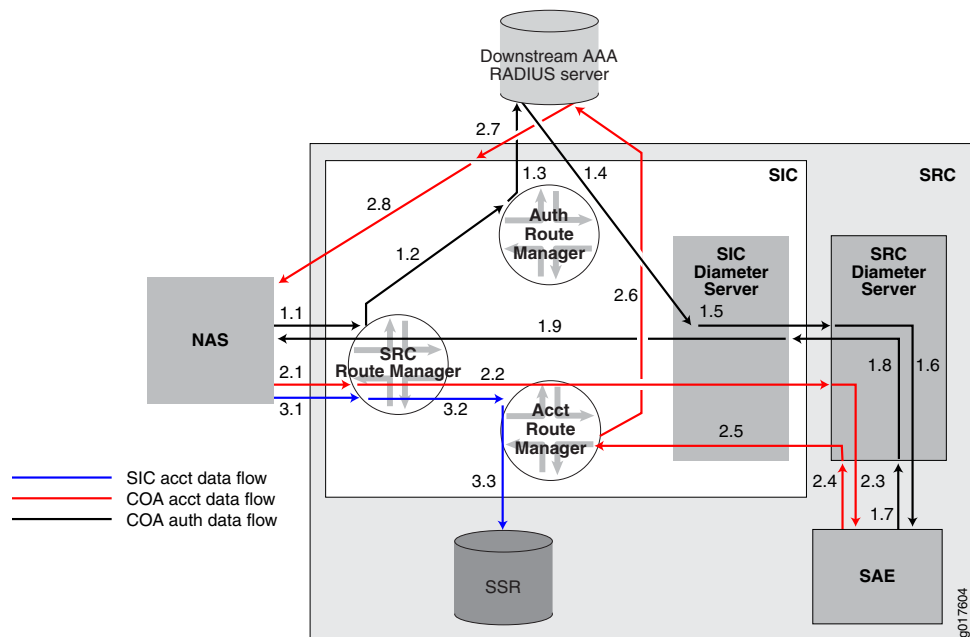
The NAS is considered a dynamic authorization target to the SIC. Dynamic authorization targets are configured in upstream network elements by using the **shared sic group identifier radius network-element id upstream dynamic-authorization-target** statement. When the SIC receives a COA or DM request, it processes the request based on the device and service and global service templates specified in the request.

- Related Documentation**
- [Device and Service Template Configuration Overview \(SRC CLI\) on page 539](#)
 - [SIC Dynamic Authorization Support Overview on page 435](#)
 - [How the Dynamic Authorization Process Works in the SIC on page 437](#)

RADIUS Authentication/Authorization and Accounting Data Flow

Following is an overview of the SIC authentication, dynamic authorization, and accounting data flow processes. [Figure 68 on page 442](#) depicts the various functions involved in these processes.

Figure 68: Data Flow



The SIC internal functions include:

- **Authentication route manager**—The authentication route manager distributes RADIUS access requests to a downstream AAA RADIUS server (authentication target) based on the configured authentication routes. The SIC does not authenticate requests by itself. It proxies access requests to a downstream AAA RADIUS server. You can have multiple downstream AAA RADIUS servers in different realms. The SIC needs to send the access request to the correct RADIUS server based on the configured authentication routes, which are usually based on realm information.
- **Accounting route manager**—The accounting route manager distributes accounting requests to accounting targets, which could be either a downstream AAA RADIUS server or the SSR, based on configured accounting routes. Similar to RADIUS authentication, there may be multiple RADIUS accounting servers in different realms. The SIC needs to forward accounting requests to the correct RADIUS server based on the configured accounting routes. The SSR is just another accounting target to which the accounting route manager can direct accounting requests.
- **SRC route manager**—Because accounting requests may be destined to the COA path or the SSR path, the SIC needs a route manager to distribute accounting traffic to the two paths based on some routing information. The route manager receives routing information from the SRC Diameter server after the Diameter connection is established with the SIC. The routing information is configured in the SRC CLI under **[shared network nas-group name routes]**. When the SIC receives a RADIUS access or accounting request, the request is sent to the SRC route manager, which matches the request against each route received from SRC Diameter servers. If a route is matched, the request is sent to the COA path. If the request is an access request but no route is matched, the SIC still sends the request to the downstream AAA servers. However, the access response from

the AAA servers is returned to the NAS. If the request is an accounting request and no route is matched, the request is distributed to the SSR path.

COA Authentication Data Flow

The SIC needs to be in the RADIUS authentication path to insert the RADIUS class attribute in the Access-Accept response. The class attribute contains the encoded Diameter session ID as well as other information. The Diameter session ID is used to correlate service accounting requests to the SAE user session.

The numbers in the following procedure correlate to the numbers in [Figure 68 on page 442](#).

- 1.1 A RADIUS access request is received by the SIC.
- 1.2 The SRC route manager locates the responsible SRC Diameter server by using the routes configured under **[shared network nas-group name routes]**. Regardless of whether the SRC route manager finds a matching route, the SIC always sends the authentication request to a downstream AAA RADIUS server. The request is sent to the SIC authentication route manager to find the correct downstream AAA RADIUS server responsible for the request.
- 1.3 The authentication route manager locates the downstream AAA RADIUS server by using configured authentication routes and sends the request to the RADIUS server for authentication.
- 1.4 The SIC receives the authentication response from the downstream AAA RADIUS server. If no matching route is found in step 1.2 or the response is Access-Reject, the SIC sends the response to the NAS.
- 1.5 If an Access-Accept message is received from the downstream RADIUS server, the SIC sends an AA-Request (AAR) to the SRC Diameter server (through the SIC Diameter server), which owns the route matching the request.
- 1.6 The SRC Diameter server forwards the AAR to SAE by using CORBA.
- 1.7 The SAE creates the user session based on the AAR, activates activate-on-login (AOL) services for the user session, and returns AA-Answer (AAA) to the SRC Diameter server. The AAA message contains the service template name and arguments for the AOL services.
- 1.8 The SRC Diameter server sends the AA-Answer message to the SIC in a Diameter message.
- 1.9 The SIC Diameter server translates the service activation requests in the Diameter AA-Answer message to RADIUS attributes based on the configured device model. The SIC sends an Access-Accept RADIUS response to the NAS with the class attribute that contains the encoded Diameter session ID. Depending on the configuration, there may be multiple rounds between the SIC and SAE to exchange service activation information before the SIC sends the Access-Accept response to NAS.

COA Accounting Data Flow

After a subscriber is authenticated, the NAS sends an Accounting-Request (ACR) message for the user session and for every service that is activated. The accounting requests must contain the class attribute returned with the Access-Accept response.

The numbers in the following procedure correlate to the numbers in [Figure 68 on page 442](#).

- 2.1 The SIC receives an ACR message from the NAS.
- 2.2 The SRC route manager locates the responsible SRC Diameter server by using the routes configured under **[shared network nas-group name routes]**. When the SRC route manager locates a match, it sends the request as an ACR message to the SRC Diameter server (through the SIC Diameter server) corresponding to the route. If the SRC route manager does not find a match, the request is sent to the SSR path (see [“SIC Accounting Data Flow \(Accounting Target=SSR\)” on page 444](#)).
- 2.3 The SRC Diameter server forwards the ACR message to the SAE in CORBA.
- 2.4 The SAE updates the user session with accounting information in the ACR message and sends an Accounting-Answer (ACA) message to the SRC Diameter server.
- 2.5 The SRC Diameter server forwards the ACA message to the SIC Diameter server.
- 2.6 The SIC receives the ACA message and needs to send the accounting request to the responsible downstream AAA RADIUS server. The SIC looks up the RADIUS server in the accounting route manager based on the configured accounting routes. The accounting route manager forwards the request to the downstream AAA RADIUS server. If accounting routes are not properly configured, the accounting route manager can forward the accounting request to the SSR. This is typically not desirable.
- 2.7 The downstream AAA RADIUS server sends an accounting response to the SIC.
- 2.8 The SIC sends the accounting response to the NAS.

SIC Accounting Data Flow (Accounting Target=SSR)

- 3.1 The SIC receives an accounting request from the NAS.
- 3.2 The SRC route manager cannot locate a match (either because no SRC Diameter server is connected or because the connected SRC Diameter servers do not have any route matching the request), so it sends the accounting request to the accounting route manager.
- 3.3 The accounting route manager evaluates the request against the configured accounting routes. Depending on the route configuration, the request may be sent to the SSR or to a downstream AAA RADIUS server. In the latter case, the downstream RADIUS server can be another SIC, which then stores the accounting request in the SSR.



NOTE: If SRC routes configured under [shared network nas-group *name* routes] for the SRC nas-group, and the SIC accounting routes are configured properly, the SIC can process accounting requests by using either a downstream AAA server or the SSR.

Related Documentation

- [Configuring AAA Policies \(SRC CLI\) on page 537](#)
- [Configuring the SAE to Manage AAA Devices on page 535](#)
- [Configuring the NAS Groups \(SRC CLI\) on page 531](#)
- [How the Dynamic Authorization Process Works in the SIC on page 437](#)

Local and Shared Configurations for the SIC (SRC CLI)

For the SIC, you need to define both a local and a shared group configuration.

Local Configuration

A local configuration applies to a specific server instance in the SIC group. The local configuration specifies the name of the server and the properties the server uses to connect to the Juniper Networks database where the configuration is stored. You specify the local server name by using the **edit slot *number* sic server** statement. You specify the connection properties for the Juniper Networks database by using the **slot *number* sic initial directory-connection** statement.

Shared Configuration

The SIC shared group configuration contains the configuration used by a group of servers. Each SIC server must belong to a group. The SIC group configuration controls the properties for the accounting methods, authentication route targets, dictionaries, editing rules, and RADIUS and Diameter options.

You create the SIC shared group configuration by using the **slot 0 sic server name /group-name/server-name** statement. The identifier associated with the group is the name of the shared configuration. This statement creates the shared group configuration and populates the server configuration with default data. Use this command to add servers to the group and populate the server with default data.

In addition, certain configuration options applicable to the individual server instances belonging to the group are also stored in the shared group configuration under the individual server name. These configuration options include the accounting and authentication routing rules, the event logging configuration, and the RADIUS and Diameter transport configurations specific to the server instance. You configure these options by using the **edit shared sic group identifier server** statements.

For example, if you want to create an SIC group named **server-group1** that includes a server named **server-bldg5**, from configuration mode:

- Specify *group-name* and *server-name*.

```
[edit]
user@host# edit slot 0 sic server
set name /server-group1/server-bldg5
```

The following rules depict how a new SIC group or server configuration is created on successfully committing the configuration:

- If the **group-name** does not exist in the Juniper Networks database, a new group and server instance as specified in this statement are created and populated with default data.
- If the **group-name** already exists in the Juniper Networks database, a server instance as specified in this statement is created under the group and populated with default data.

If you want to add another server to server-group1 named **server-bldg5a**, execute:

```
[edit]
user@host# edit slot 0 sic server
set name /server-group1/server-bldg5a
```

Creating a server by using this statement populates it with default data. You can also add a new server to an existing group by using the **shared sic group identifier server identifier** statement. However, this statement does not populate the server with default data.

**Related
Documentation**

- [Creating an SIC Server Instance \(SRC CLI\) on page 477](#)
- [Creating an SIC Group and Server \(SRC CLI\) on page 476](#)

Accounting Methods and Targets (SRC CLI)

The available types of *accounting methods* for the subscriber information collector (SIC) include:

- Database—Stores accounting events in the SSR database
- Proxy—Forwards accounting events to a downstream network element that contains a proxy AAA server

You configure *accounting targets* by specifying the accounting method used by the SIC group. The accounting target for explicit accounting routing rules can be either the SSR database, or an AAA server in a downstream network element. The accounting target for implicit routing rules is always a proxy AAA server in a remote network element.

Using the SSR Database as the Accounting Method

You use the **shared sic group identifier accounting-method accounting-method-name database** statement to configure the SSR database as the accounting method. Configure this accounting method as the accounting target by using the **shared sic group identifier server identifier accounting-route id target** statement. In addition, you need to define the mapping between any request attributes, literals, or SIC variables, and the respective SAE plug-in attributes by using the **shared sic group identifier accounting-method**

accounting-method-name database plug-in-attribute statement. You also need to configure the mapping between the SAE plug-in attributes and the columns in the SSR database by using the **shared database cluster (primary) attribute-associations entity name field** statement.

To conserve resource space in the SSR database, you can enable the **remove-stopped-sessions** option. When you enable this option, user session entries are deleted from the SSR database when the SIC receives an Accounting-Stop message. The corresponding service sessions are also deleted. The sequence for this is as follows:

1. The SIC receives an Accounting-Stop message from the NAS.
2. The SIC deletes the corresponding entry in the SSR user session table.
3. The SIC deletes the corresponding service session table entries.
4. The SSR notifies the SAE about the delete operation in SSR.
5. The SAE, upon receiving the delete event notification, stops the services and logs out the user.

If the **remove-stopped-sessions** option is not set, the following sequence occurs:

1. The SIC receives an Accounting-Stop message from the NAS.
2. The SIC updates the state of the corresponding entry in the SSR user session table to STOPPED.
3. The SSR notifies the SAE about the update operation in SSR.
4. The SAE, upon receiving the update event (state of the corresponding entry changed to STOPPED), stops the services and logs out the user.

Following is a basic working configuration for the database accounting method that includes the default configuration:

- Accounting listener—You must specify a name for the accounting listener. You can use the default port 1813.
- Device model—You can use the default model.
- Database accounting method—This is the default accounting method.
- Upstream RADIUS network element and accounting client—You must define the upstream network element and at least one accounting client.
- Accounting route—You must define the accounting route.
- Logger—You can use the default logger.

How the SIC Seamlessly Processes IP Address Changes When the SSR Database Is the Accounting Method

When a subscriber using a mobile device connects to a WiFi network, the subscriber is identified by a session-id (or mac-address) within the WiFi domain. The mobile device is also allocated an IP address. The subscriber logging in to the network is given access to the network. However, if the mobile device is inactive for a certain period of time, the

mobile device releases the IP address and enters idle mode. When the mobile device becomes active again, it is again allocated an IP address, which is most likely not the same IP address allocated for the previous session. This IP address change also occurs when a subscriber moves between buildings within the same WiFi domain. Because of the IP address change, the subscriber needs to log back in to the network, even though the session-id remains the same. However, the user is able to log in and out of the network seamlessly.

To avoid requiring the subscriber to log back in after an IP address change, the SSR tracks subscriber sessions on the basis of both the IP address and session-id. When the SIC receives an accounting message that includes the new IP address and the initial session-id, it removes the record with the old IP address from the SSR database and inserts the record with the new IP address. To use this capability, you must map a unique value such as MAC-Address or IMSI to the Session ID column in the subscriber sessions table.

Mapping Attributes When Using the Database Accounting Method

When you use the SSR database as the accounting method, you need to define the mapping between:

- SIC request attributes and variables and SAE plug-in attributes
- SAE plug-in attributes and fields in the subscriber sessions table in the SSR database

The SIC uses internal variables to store intermediate results of transaction processing, such as editing. Every variable must have a name and a value. You can define variables while configuring editing rules. If a variable is configured in an editing rule and it does not already exist, it is created. Another place where variables are used is in the mapping between the SIC, SAE plug-in attributes, and the fields in the subscriber sessions table in the SSR database. A variable from an editing rule can be used in the mapping, which allows you to store the value of the variable (the result of the editing process) in the subscriber sessions table field. There are some internal SIC variables such as:

- ReceiveTime—This is the timestamp of the accounting event.
- UserStatusType—This is correlated to the RADIUS Acct-Status-Type: 1 for Accounting-Start, 2 for Accounting-Stop.



NOTE: You must configure the SIC group to use the database accounting method. In addition, you must map any fields in the SSR subscriber sessions table that have a not-null requirement to either a request attribute or SIC variable.

The following output shows the attribute mapping between SAE plug-in attributes and the SIC request attributes and variables:

```
plug-in-attribute id="PA_USER_INET_ADDRESS" request-attribute="NAS-IP-Address"
plug-in-attribute id="PA_LOGIN_NAME" request-attribute="User-Name"
plug-in-attribute id="PA_PROPERTY.session-id" variable="ReceiveTime"
```


The following output shows the mapping between SAE plug-in attributes and the fields in the subscriber sessions table in the SSR database:

```

ssrMapping
table name="SubscriberSessions"
attributeMapping attribute="PA_USER_INET_ADDRESS" field="UserIPAddress"
attributeMapping attribute="PA_LOGIN_NAME" field="UserName"
attributeMapping attribute="PA_PROPERTY.session-id" field="Sessionid"
table
ssrMapping

```

This mapping results in attributes and variables mapped as shown in [Table 26 on page 449](#).

Table 26: Example of SSR Database Mapping

SIC Variable or Attribute	SAE Plug-In Attribute	Field in Subscriber Sessions Table
Request-attribute=NAS-IP-Address	PA_USER_INET_ADDRESS	UserIPAddress
Request-attribute=User-Name	PA_LOGIN_NAME	UserName
Request-attribute=Acct-Session-Id	PA_PROPERTY.session-id	Sessionid

Using the Proxy RADIUS Accounting Method

The proxy RADIUS accounting method forwards accounting events to an accounting target (AAA server) located in a downstream network element. You use the **shared sic group identifier accounting-method accounting-method-name proxy radius** statement to configure a proxy RADIUS accounting method as the accounting method. You configure the downstream network element that contains the AAA server by using the **shared sic group identifier radius network-element id downstream** statement. You configure the AAA server as the accounting target by using the **shared sic group identifier radius network-element id downstream accounting-target** statement.

Following is a basic working configuration for the proxy accounting method that includes the default configuration:

- Accounting listener—You must specify a name for the accounting listener. You can use the default port 1813.
- Device model—You can use the default model.
- Proxy accounting method—You must configure the proxy accounting method.
- Outbound transport—You can use the default outbound transport.
- Upstream RADIUS network element and accounting client—You must define the upstream network element and at least one accounting client.
- Downstream RADIUS network element and accounting target—You must define the downstream network element and the accounting target.
- Accounting route—You must define the accounting route.
- Logger—You can use the default logger.

Related Documentation

- [Configuring the SSR Database as the Accounting Method \(SRC CLI\) on page 508](#)
- [Configuring Proxy RADIUS as the Accounting Method \(SRC CLI\) on page 509](#)
- [Request Routing \(SRC CLI\) on page 450](#)
- [Configuring Downstream Network Elements and Accounting and Authentication Targets \(SRC CLI\) on page 496](#)
- [Configuring the Optional Editing Rules Used by the SIC Group \(SRC CLI\) on page 505](#)

Authentication Route Targets (SRC CLI)

A downstream AAA server is responsible for authenticating all authentication requests. When the SIC receives a RADIUS authentication request, it evaluates authentication routes to determine which downstream AAA server is responsible for the request and then routes the request to the server. You configure the downstream AAA server as an authentication target by using the **shared sic group identifier radius network-element id downstream (authentication | accounting) authentication-target name** statement. You can configure multiple authentication targets. You can also define any number of authentication routes. The same routing conditions available for accounting routes can be configured for authentication messages.

Related Documentation

- [RADIUS Authentication/Authorization and Accounting Data Flow on page 441](#)
- [Accounting Methods and Targets \(SRC CLI\) on page 446](#)
- [Request Routing \(SRC CLI\) on page 450](#)

Request Routing (SRC CLI)

SIC routing rules define how the SIC routes accounting and authentication requests. You configure routing rules for each server in the SIC group. There are two types of routing rules:

- Explicit routing rules
- Implicit routing rules

Explicit Routing Rules

An explicit route is a collection of criteria used to select a particular routing target. Explicit routing rules consist of a condition, or set of conditions, and an accounting or authentication target to which the request is to be routed. Routing criteria consist of a list of simple Boolean expressions on RADIUS attributes and transactional variables. The accounting target is either the SSR database or a downstream network element that contains an AAA server. The authentication target is an AAA server in a downstream network element.

You specify explicit routing rules based on the following match conditions:

- Realm name

- User identity
- Request attribute

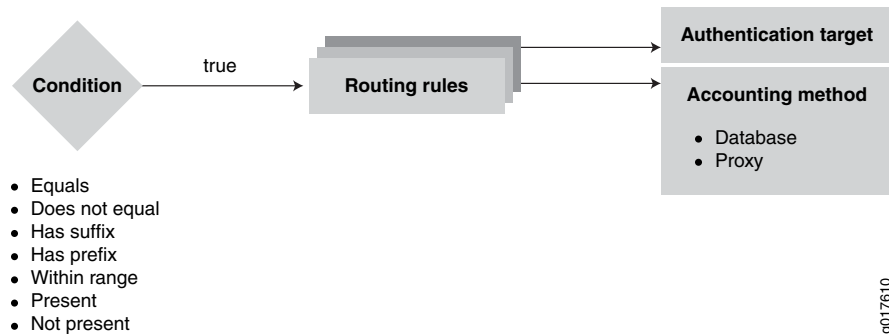
You can test the value of the match condition for the following conditions:

- Present
- Not present
- Equal
- Does not equal
- Has prefix
- Has suffix
- Range

The has prefix and has suffix condition tests work only on the string representation of the value. To test for a range condition, specify a low value and a high value.

[Figure 69 on page 451](#) depicts the explicit routing rule process.

Figure 69: Explicit Routing Rule Process



When the SIC receives an accounting or authentication request, it evaluates any defined explicit routing rules in the order they were configured. When multiple routes are configured, they are evaluated in the order they are displayed by the **show** command. A newly created route is displayed last among the routes and has the lowest priority, so it is evaluated last. You can use the SRC CLI **insert** command to move a route before or after another route to change its evaluation order. The higher a route is displayed on the list, the sooner it is evaluated. For a route to be selected, *all* conditions of the rule must be true. If a match is not found, the next configured rule is examined, and so on. As soon as a rule with matching criteria is encountered, the iteration stops and the accounting or authentication target in that rule is selected as the destination for the request. If a match for all conditions cannot be found in the explicit routing rules, the implicit routing rules are examined.

Before the request is sent to the specified target, you can edit it by optionally specifying an editing rule for the accounting route.

Examples of explicit routing rules are:

Rule 1: If the NAS-Identifier is nas1, then the target is accounting method method1, which is the database accounting method:

```
[edit shared sic group group1 server server1 accounting-route route1]
user@host# show
target method1;
condition {
  attribute nas-identifier;
  equals nas1;
}
```

Rule 2: If the NAS-Identifier is nas2 then the target is accounting method method2, which is the proxy accounting method, pointing to the RADIUS network element rne1:

```
[edit shared sic group group1 accounting-method method2 proxy]
user@host# show
radius {
  network-element rne1;
}
...
[edit shared sic group group1 server server1 accounting-route route2]
user@host# show
target method2;
condition {
  attribute nas-identifier;
  equals nas2;
}
```

This example is an accounting route example. Authentication routes work the same way.

Implicit Routing Rules

Implicit routes are realm based. You configure implicit routes by defining a network element that contains a remote AAA server and assigning it the proxy function. You can then either define a default route used for all requests from all realms or specify that only requests from specific realms are routed to the proxy AAA server. When you specify realms, you have the option to set a condition of either an exact match of the realm string, or a match on the prefix of the realm string.

Implicit routing rules have lower priority and are evaluated only if a match is not found for explicit routing rules. When a request is received, the SIC server evaluates the associated routing rules. First, the server evaluates any explicit routing rules. If no match is found, the server evaluates the implicit routing rules. When a match is found, the server processes the request by routing it to the specified network element that has the proxy function assigned to it.

Related Documentation

- [Configuring Explicit Routing \(SRC CLI\) on page 512](#)
- [Configuring Implicit Routing \(SRC CLI\) on page 515](#)
- [Configuring the Optional Editing Rules Used by the SIC Group \(SRC CLI\) on page 505](#)
- [SIC Editing Rules \(SRC CLI\) on page 453](#)
- [Accounting Methods and Targets \(SRC CLI\) on page 446](#)

SIC Editing Rules (SRC CLI)

Before the SIC sends the request to the specified accounting or authentication target, the request can optionally be edited according to the editing rules associated with the selected routing rule. Editing rules are similar to routing rules, in that the request is examined for matching conditions, and if one is found, the request is edited and then sent to the accounting target.

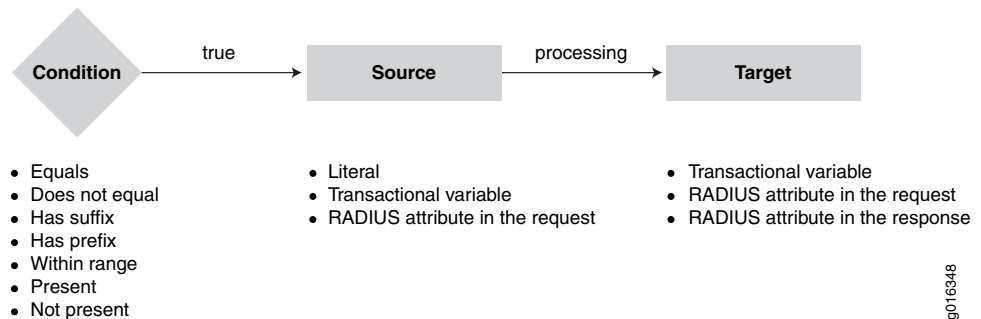
In addition to editing RADIUS attributes, the SIC can edit transactional variables and literals. Editing rules can define new transactional variables, in addition to certain built-in variables such as the result of username parsing, network access server (NAS) client lookup, and so on. Transactional variables are also referenced in the columns of the subscriber sessions table in the SSR database, thus allowing you to store the results of request processing and editing in the subscriber sessions table.



NOTE: You can control the number of transactional variables the SIC supports. The default value is 255. When you change this limit, you need to restart the SIC.

Figure 70 on page 453 depicts the SIC editing rule process.

Figure 70: SIC Editing Rule Process



You configure editing rules by defining the source and its associated match conditions, the editing conditions applied to the source value, and the target in which the edited result is placed. First, the SIC examines the specified source in the request for the defined match conditions. If all conditions are found to be true, the SIC edits the source value based on the defined editing conditions. The result is then placed in the defined target. The edited request, including both the original source and the new target value, is sent to the target.

Each editing rule is a simple assignment of a source (RValue) and a target (LValue). In any assignment the target can be one of the following:

- Transactional variable
- RADIUS attribute in the request
- RADIUS attribute in the response

The source can be one of the following:

- Literal
- Transactional variable
- RADIUS attribute in the request

The match conditions that you can test for in the source include whether a specific realm, user identity, or request attribute is:

- Present
- Not present
- Equals
- Does not equal
- Has suffix
- Has prefix
- Within range

If a match condition is found on the source, you can append or replace the value of the source and place it in the target.

Additionally, if the source is a request attribute, you can edit the value of the source by removing the suffix or prefix, or removing what is before or after the @ and place the result in the target. The remove before @ and remove after @ options can contain wildcards.

For example, if the request contains johnsmith@abcd.net:

- Removing the prefix john results in: smith@abcd.net
- Removing the suffix .net results in: johnsmith@abcd
- Remove the attribute before the @ results in: abcd.net
- Remove the attribute after the @ results in: johnsmith

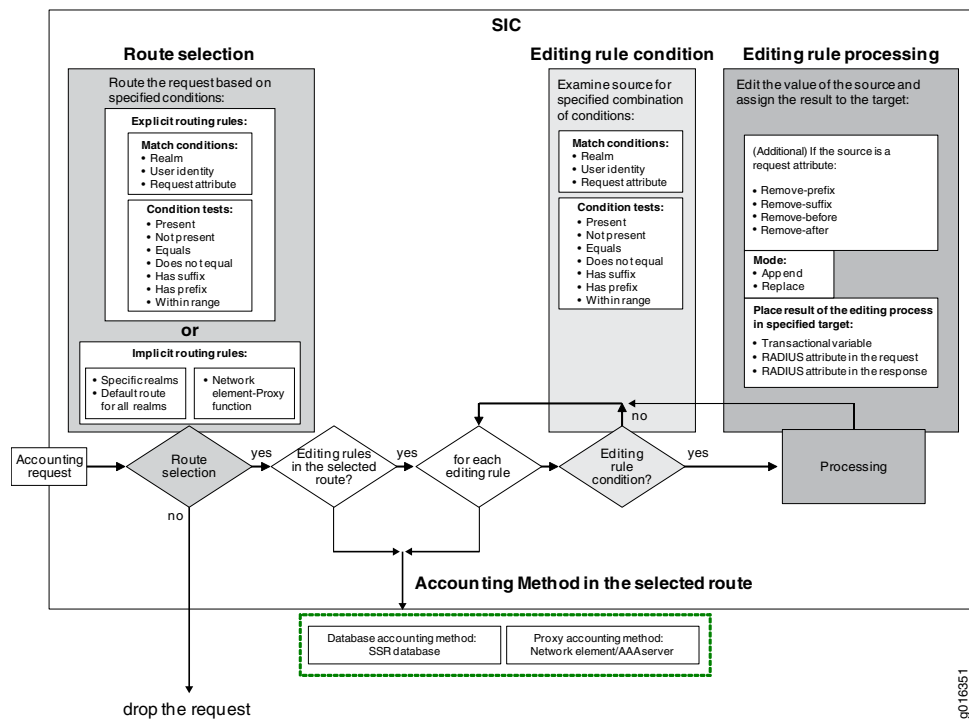
The SIC supports binary-level editing as follows:

- When a source attribute is of an octets type in the dictionary, the value of the attribute is encoded in hexadecimal format with a {hex} prefix. Operations including remove-before, remove-after, remove-prefix, and remove-suffix for such attributes work on their hexadecimal format. The search pattern used by the preceding operations must also use the hexadecimal format. For example, if the source string contains "@" and you want to remove everything after "@", the search pattern is 40, which is the ASCII number of @.
- A remove-before operation on an octets type attribute puts the {hex} prefix after the operation. A remove-prefix operation on an octets type attribute must include {hex} in the search pattern and the {hex} prefix is not added back after the operation. If you want to keep the {hex} prefix, use the remove-before operation.

- When converting an octets type attribute to another type, the SIC first tries to convert the attribute by using the binary representation of the target type. For example, the ip-address type attribute uses four octets. If the source attribute also uses four octets, the conversion is successful.
- If the binary conversion is not successful, the SIC tries the conversion using a printable string representation. For example, the printable format for the ip-address type attribute is "x.x.x.x". The conversion fails if the octets attribute value does not follow the printable format of the target attribute.

Figure 71 on page 455 depicts the editing rule process and the accounting route selection process. Authentication requests can also use this editing process. The target for authentication requests is always a downstream network element that includes an AAA server target.

Figure 71: Editing and Accounting Routing Rule Conditions and Processes



Example of an editing rule:

- If Unisphere-Virtual-Router is present, then the transactional variable vpn-id is the substring after "." in Unisphere-Virtual-Router.
- If NAS-Identifier is nas3, then the transactional variable vpn-id is the realm portion of User-Name (realm transactional variable). Otherwise, the transactional variable vpn-id is the NAS-Identifier.

```
[edit shared sic group group1 editing edit1]
user@host# show

target {
variable vpn-id;
```

```
}
source {
  request-attribute Unisphere-Virtual-Router {
    condition {
      attribute Unisphere-Virtual-Router;
      check-presence;
    }
    remove-before *:
  }
  variable realm {
    condition {
      attribute nas-identifier;
      equals nas1;
    }
  }
}
default {
  request-attribute nas-identifier ;
}
```

**Related
Documentation**

- [Configuring the Optional Editing Rules Used by the SIC Group \(SRC CLI\) on page 505](#)
- [Configuring Explicit Routing \(SRC CLI\) on page 512](#)
- [Accounting Methods and Targets \(SRC CLI\) on page 446](#)
- [Configuring the SSR Database as the Accounting Method \(SRC CLI\) on page 508](#)
- [Configuring Proxy RADIUS as the Accounting Method \(SRC CLI\) on page 509](#)

RADIUS and Diameter Configuration for the SIC Overview (SRC CLI)

The RADIUS and Diameter configuration for the SIC group consists of:

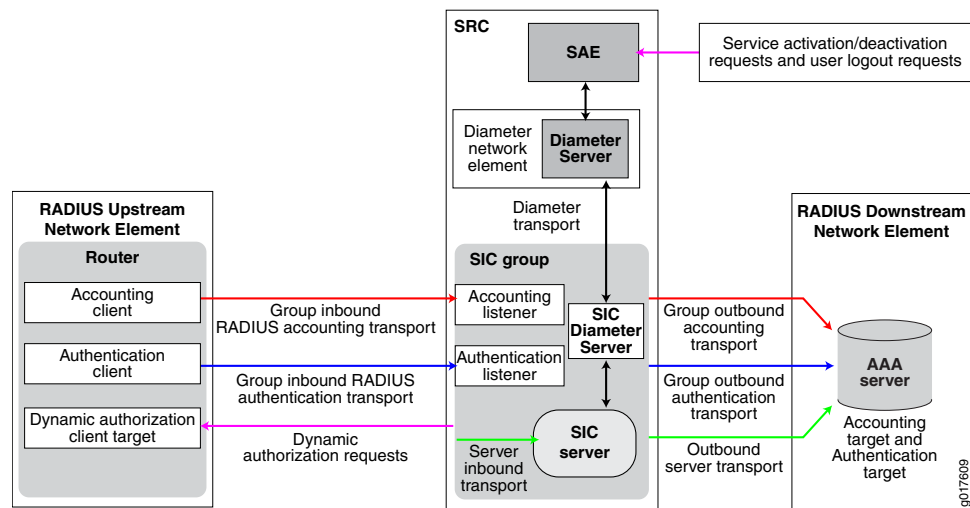
- RADIUS accounting and authentication listeners
- SIC Diameter server
- UDP transports for the group and each SIC server
- At least one RADIUS upstream network element with an accounting client, authentication client, and a dynamic authorization target
- At least one RADIUS downstream network element with an accounting target and an authentication target
- (Optional) A proxy function—Used for defining implicit routing rules



NOTE: Authentication clients and targets and dynamic authorization targets are optional and required only if you are supporting COA and DM requests.

[Figure 72 on page 457](#) depicts the SIC RADIUS and Diameter configurations, which are detailed in the following sections.

Figure 72: SIC RADIUS and Diameter Configuration



RADIUS Accounting and Authentication Listeners

The SIC includes accounting and authentication listeners that listen for RADIUS accounting and authentication messages from the NAS and filter undesired events based on attachment session attributes. You must configure at least one accounting listener for the SIC group. If you are supporting COA or DM requests, you must also configure at least one authentication listener. To configure the listeners, you specify the UDP port that the SIC listens on as well as other parameters that control the receipt of UDP packets. The configuration options associated with the listeners control the RADIUS inbound transport for the SIC group, which is used to communicate with upstream network elements that contain one or more accounting and authentication clients.

SIC Diameter Server

The SIC includes a Diameter server. The SIC Diameter server communicates with the SRC Diameter server, which is a peer to the SIC Diameter server. The SIC Diameter server provides the translation between the SAE and SIC by translating COA or DM into VSAs so that they can be understood by the NAS. The SRC Diameter server also passes routing information from the SAE to the SIC Diameter server. This routing information is configured in the SRC CLI under `[shared network nas-group name routes]`.

Typically, the SIC connects to more than one SRC Diameter server. The Diameter servers may belong to a different redundant group. Each redundant group manages one or more NAS groups. Depending on the configuration, the SIC may connect to the SRC Diameter server in the same C Series Controller.

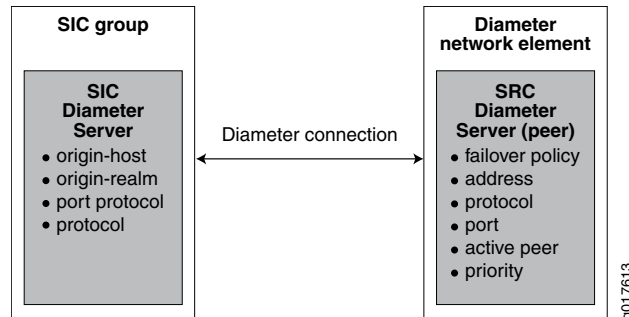


NOTE: Because both the SIC Diameter server and the SRC Diameter server need to listen to Diameter traffic, they should use different ports when both are active in the same C Series Controller.

SIC Diameter Server Configuration Overview

To configure the SIC Diameter server, you need to configure the server identity, port, and protocol. To configure the SRC Diameter server as a peer, configure the Diameter network element, the failover policy information, address, protocol, and active peer information. [Figure 73 on page 458](#) depicts the Diameter configuration for SIC.

Figure 73: SIC Diameter Configuration



RADIUS Network Elements

A network element is an addressable, logical network entity that contains RADIUS clients and targets.

An upstream RADIUS network element contains:

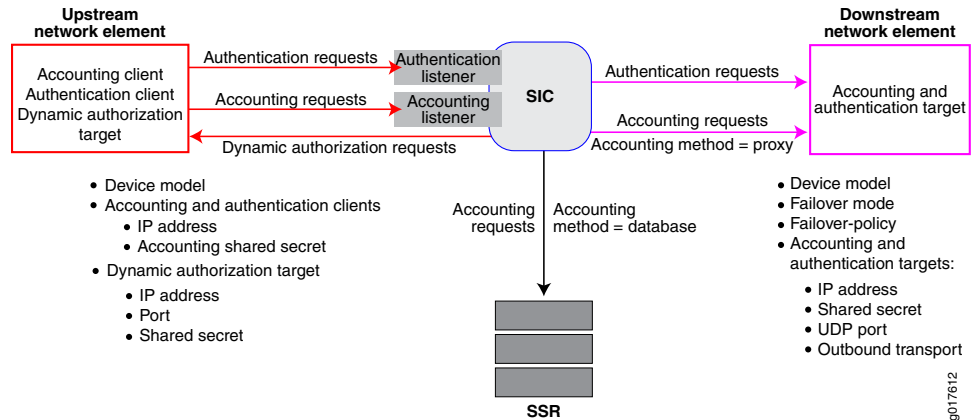
- Accounting clients, which send accounting messages to the SIC accounting listeners
- Authentication clients, which send authentication requests to the SIC authentication listeners
- Dynamic authorization targets, which receive COA/DM requests from the SIC

A downstream RADIUS network element contains an AAA server (target), which receives accounting and authentication messages from the SIC.

Network elements can contain multiple clients and targets.

[Figure 74 on page 459](#) depicts network elements and the various clients and targets they contain.

Figure 74: Upstream and Downstream Network Element Clients and Targets

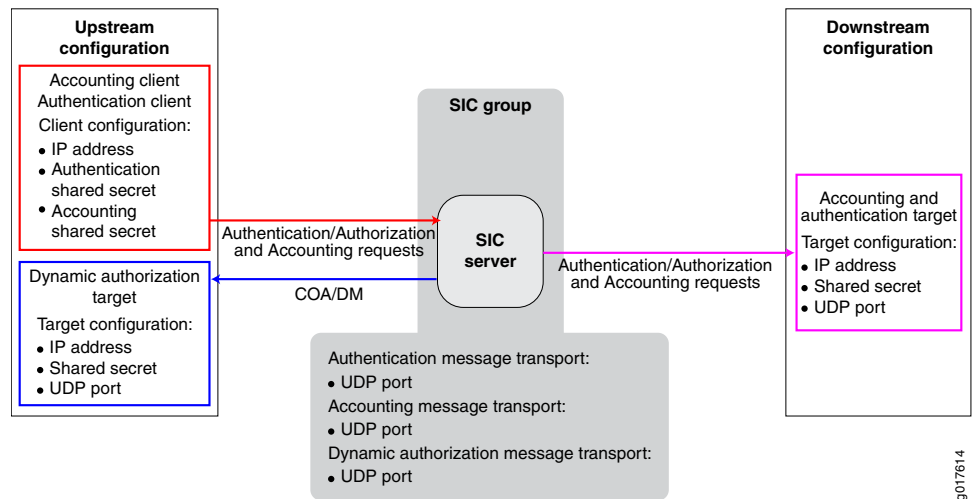


Configuring Upstream RADIUS Network Elements Overview

You need to configure at least one upstream network element containing at least one accounting client and one authentication client. If you are supporting dynamic authorization requests (Change of Authorization [COA] or Disconnect Messages [DMs], you also need to configure a dynamic authorization target in the upstream network element. For dynamic authorization targets, you also need to configure the failover policy and mode. You configure upstream network elements by using the **shared sic group group-name radius network-element upstream** statement and specifying the shared secret, IP address, and device model of the accounting or authentication client.

Figure 75 on page 459 depicts RADIUS network element upstream and downstream client and target configuration options.

Figure 75: Upstream and Downstream Network Element Client and Target Configuration Options



Configuring Downstream RADIUS Network Elements Overview

You need to configure a downstream network element for the accounting and authentication targets. [Figure 75 on page 459](#) depicts RADIUS network element upstream and downstream configurations. You configure downstream network elements by using the **shared sic group *group-name* radius network-element downstream** statement and specifying the outbound transport, UDP port, shared secret, and IP address of the accounting target. You also need to specify the failover policy, failover mode, and the device model of the accounting target (AAA server).

Using the Proxy Function to Define Implicit Routing Rules

You use the proxy function to define implicit routing rules for accounting and authentication requests by specifying a remote AAA server as a proxy and having the SIC forward accounting and authentication requests to it. When the SIC receives a request, it first evaluates any explicit routing rules. If no match is found, it evaluates implicit routing rules. If a match is found, the SIC routes the request to the proxy AAA server.

You configure the proxy function by configuring a network element and specifying it as a proxy. You can then either define a default route used for all requests from all realms, or you can specify that only requests from certain realms are routed to the proxy AAA server. When you specify realms, you have the option to specify a match condition of either an exact match of the realm string or a match on the prefix of the realm string.

Related Documentation

- [Configuring the RADIUS Accounting Listener for the SIC Group \(SRC CLI\) on page 480](#)
- [Failover Policy on page 460](#)
- [Configuring the Outbound RADIUS Transport of the SIC Group \(SRC CLI\) on page 484](#)
- [Configuring Implicit Routing \(SRC CLI\) on page 515](#)
- [Request Routing \(SRC CLI\) on page 450](#)
- [RADIUS and Diameter Transports on page 462](#)

Failover Policy

The failover policy manages how the SIC sends messages over multiple paths to upstream and downstream network elements, and how it responds when it does not receive a response from a target in the network element within a specified amount of time. You configure a failover policy for the following targets:

- Accounting targets in downstream network elements
- Authentication targets in downstream network elements
- Dynamic authorization targets in upstream network elements

When multiple paths are configured to a network element, you need to specify the order in which the SIC uses the paths to send messages to the target. When the SIC sends a message to one of these targets, it expects to receive a reply within a certain amount of time as specified by the fast fail policy. If it does not receive the reply in the specified

time, it places the target into fast fail mode and rejects the request. You configure the failover policy by specifying the fast fail and retry parameters, which control such options as minimum number of times the server retransmits messages to the accounting, delay between sending retransmissions, as well as various timeout and delay settings that control how fast the server goes in and out of fast fail mode.

When the SIC has messages to send to a target, it first examines what failover mode is configured—either round-robin or primary or backup. It then examines whether all paths to the target are operational. It then sends the messages accordingly (over whatever paths are operating). As such, these features inherently manage communication failures by adjusting what paths are used when one or more paths are not working.

Failover Mode

Failover mode manages how the SIC sends messages over multiple paths to a network element target. You can configure failover mode for either round-robin or primary or backup. When the server has a message to send, it first examines whether failover mode is set for round-robin or primary or backup. Next, it examines whether all paths to the network element where the target resides are operational. It then sends the message over whatever path is operational based on failover mode.

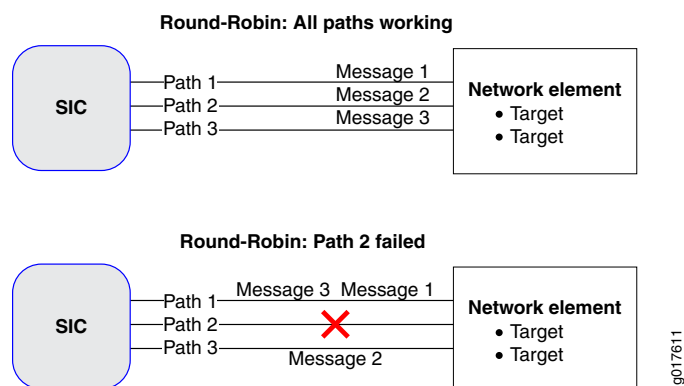
Round-Robin

When failover mode is set to round-robin, the SIC alternates the path it uses to send messages to the target.

Figure 76 on page 461 illustrates how the Round-Robin feature operates when all paths are working properly (top portion), and how it operates when one of the paths has failed (bottom portion).

With all three paths operating properly, if the SIC received three messages, the first message would be sent over path 1, the second message would be sent over path 2, and the third message would be sent over path 3. The next message received would be sent over path 1, and so on. However, if the server received three messages and path 2 had failed, the first message would be sent over path 1, the second message would be sent over path 3, and the third message would be sent over path 1.

Figure 76: Round-Robin



Primary or Backup

When failover mode is set to primary or backup, the SIC sends all messages over the first path defined in the ordered list. If the first path fails, all messages are sent over the next path in the ordered list. When the first path becomes operational again, all messages are again sent over it.

Related Documentation

- [Authentication Route Targets \(SRC CLI\) on page 450](#)
- [Dynamic Authorization Targets \(SRC CLI\) on page 441](#)
- [Configuring Failover Mode and Policy \(SRC CLI\) on page 499](#)

RADIUS and Diameter Transports

To support RADIUS accounting and authentication, you need to configure both inbound and outbound transports for the SIC group and each SIC server within the group.

To support dynamic authorization requests, you need to configure the RADIUS dynamic authorization transport as well as the Diameter transport to the Diameter server in SRC.

Inbound and Outbound RADIUS Accounting Transports for the SIC Group

To support RADIUS accounting in the SIC group, you need to configure the inbound transport by configuring the accounting listeners. You need to configure the group RADIUS outbound accounting transport if you are using the proxy accounting method, or if you are using implicit routing to forward accounting messages to a remote AAA server accounting target. You do not need to configure the group RADIUS outbound accounting transport if you are using the database accounting method.

Inbound and Outbound RADIUS Authentication Transports for the SIC Group

To support RADIUS authentication in the SIC group, you need to configure the inbound transport by configuring the authentication listeners. You also need to configure the outbound RADIUS group authentication transport to the downstream AAA server authentication target. The SSR database cannot be used for authentication.

SIC Server Inbound and Outbound RADIUS Transports

You can configure the inbound and outbound transport for each server in the SIC group. To do this, you specify the names you configured for the group inbound transport (authentication and accounting clients) and the group outbound transport and then specify the IP address the server used to send and receive UDP messages. This is optional. If it is not configured, the address for inbound transport is all IP addresses configured on the C Series Controller. If the address for the outbound transport is not configured, the SIC allows the SRC software to choose one from the list of addresses configured on the local machine.

Diameter Transport

You must configure bidirectional transport (Diameter connection) between an SIC server and the SRC Diameter server.

For the SIC server, you specify the origin-host, origin-realm, transport protocol (TCP or SCTP), and the port the SIC server uses for Diameter messages.

The SRC Diameter server must be defined within a network element. You need to specify the network element ID and the associated failover policy, as well as peer information for the SIC server including IP address, protocol (TCP or SCTP), and port. In addition, you must specify whether or not the peer is an active peer and the priority. These options are used when you have multiple connections from the Diameter server to multiple SIC servers (peers).

Related Documentation

- [RADIUS and Diameter Configuration for the SIC Overview \(SRC CLI\) on page 456](#)
- [Configuring the RADIUS Accounting Listener for the SIC Group \(SRC CLI\) on page 480](#)
- [Configuring the Outbound RADIUS Transport of the SIC Group \(SRC CLI\) on page 484](#)
- [Configuring the RADIUS Transport for an SIC Server \(SRC CLI\) on page 485](#)

SIC Dictionaries and Device Models Overview (SRC CLI)

The SIC uses dictionaries to define RADIUS attributes. Dictionaries identify the attributes the SIC expects when receiving RADIUS requests from a specific type of device—for example, an upstream NAS or downstream AAA server, and the attributes the SIC includes when sending a RADIUS response to a specific type of device. The SIC uses these definitions to parse accounting requests and generate responses.

Dictionaries and the Device Models Supported by the SIC Group

Each SIC group configuration must include a dictionary and a list of device models.. When you configure the device model, you specify an identifier and the associated dictionary that the SIC uses when communicating with the device. The dictionary assigned to the device model identifies the attributes the SIC expects when receiving RADIUS requests from the specific device, and the attributes the SIC needs to include in responses to the device. The SIC uses these definitions to parse accounting, authentication, and dynamic authorization requests and generate responses.

In addition, when you configure an upstream or downstream network element, you need to specify which device models it supports based on the list of device models you have configured for the SIC group. Thereafter, whenever the SIC receives a RADIUS packet from the network element, it consults the associated dictionary for the attributes that it encounters in the packet.

Configuring Device Models and Their Associated Dictionaries for the SIC Group Overview

You need to specify the device models and their associated dictionaries for the SIC group and each network element the SIC needs to communicate with. To specify these for the SIC group, use the **shared sic group identifier model id** statement, and to specify these for network elements, use the **shared sic group identifier radius network-element id upstream** and the **shared sic group identifier radius network-element id downstream** statements.

Modifying a Dictionary

You can add attributes or modify existing attributes in dictionaries. However, you cannot delete the dictionary itself or any of the existing attributes. If you modify a dictionary, you need to restart the SIC for the change to take effect. Use the **shared sic group group name dictionary id** configuration statement to modify an existing dictionary.

Configuring the Dictionaries Used by the SIC Group Overview

The SIC includes standard RADIUS devices. All dictionaries implicitly import RADIUS standard attributes. The RADIUS dictionary is the default dictionary. It is loaded by default when you configure an SIC group and is sufficient for most environments.

- Related Documentation**
- [Configuring the Device Models Supported by the SIC Group \(SRC CLI\) on page 480](#)
 - [Configuring Dictionaries for the SIC Group \(SRC CLI\) on page 478](#)

SIC Local Realms Overview

Defining a realm as local to the SIC group instructs the SIC to use a local server to process the request. The network access identifier (NAI) in the request identifies the intended realm. To properly interpret requests received from intermediate servers, the SIC server must know which realms it is responsible for servicing locally.

When a request is received, the SIC examines the NAI to determine the realm to which the request is to be routed. If the realm is configured as a local realm to the SIC server, the request is processed by the local server. If no realm is present in the NAI, the request is considered to be local.

- Related Documentation**
- [Configuring What Realms Are Local to the SIC Group \(SRC CLI\) on page 501](#)
 - [Local and Shared Configurations for the SIC \(SRC CLI\) on page 445](#)
 - [Creating an SIC Server Instance \(SRC CLI\) on page 477](#)

SIC Event Logging Overview (SRC CLI)

SIC log streams capture different groups of server-related events at various levels of granularity. You can configure the SIC to capture any number of log streams. If you configure multiple log streams, make sure you configure unique names for each log stream by using the **shared sic group identifier server identifier logger identifier** statement. Each log stream you create captures events in a separate log file, which is date stamped, and you can also assign a prefix to it for easy identification.

Log messages are divided into several log groups according to the subject of the log information. You can configure a log stream to display only log messages from particular log groups. Each log group captures different types of server-related events. You configure the level of granularity captured for the log group by setting the event level for the log group.

Log File Options

You use the configuration options described in [Table 27 on page 465](#) to define the properties of the log files.

Table 27: SIC Log File Options

Option	Description
filename	Prefix added to the log file name. This string is prepended to each log file name.
filter	Filter to define which event messages are logged or ignored. The filter specifies the logging level, such as debug. <ul style="list-style-type: none"> Error events are captured for every log group Debug events are captured for every log group
flush-after-writes	If set to true, log messages are immediately written to the log file without buffering. Use this setting for real-time logging. If set to false, SIC log messages are kept in the buffer until the buffer is full and then all messages in the buffer are written to the log file. Use this setting for performance optimization, when real-time logging is not needed.
footer	Footer message added to the end of each log file.
header	Header message added to the beginning of each log file.
high-resolution-timestamps	High-resolution time reporting system functions are used.
maximum-file-size	New log file created after these many bytes. When a log file reaches this size, logging begins in a new log file.
prepend-message-header	Prepend each log message with additional information. Add time, thread, and transaction information to each log message. You can achieve additional fine-tuning by using the work-id-label, work-id-padding, and utc options.
rollover-interval	New log file is created after this amount of time elapses. Specified in seconds.
rollover-on-startup	New log file is created every time the server starts.
utc	Time and date values reflect Universal Time Coordinates (UTC), formerly known as Greenwich Mean Time or (GMT). Otherwise, values reflect local time.
work-id-label	Work data ID prefix added to each log message.
work-id-padding	String added to each log message if work data is not available.

Event Levels

The event level specifies the level of detail captured for the log group. You configure the event level by specifying the log group and then specifying the associated event level. The event level you specify is the highest event level displayed for the log group. You can configure the log stream to display log items from levels at and below a particular event level.

Be careful when using event logging because it consumes server resources while capturing events, and consumes disk space to store the log files. We recommend that event logging be used primarily for troubleshooting purposes, and that you limit the amount of information captured in a log stream to control the consumption of server resources and disk space. Limiting the amount of information in the log stream also makes it easier to interpret the information in the log files. For example, you might configure one log stream to capture only configuration-related events by setting the Configuration log group event level to Detail, and setting all other log group event levels to Error.

In general, each event level includes less verbose event types. For example, if you configure an event level of Warning, then warnings and errors are logged to the specified log stream. The event levels in order of increasing verbosity are shown in [Table 28 on page 466](#).

Table 28: SIC Event Levels

Event Level	Description
None	No events will be logged for the log group.
Error	An error as an event that may cause the system to operate incorrectly. Examples include exceptions being thrown, an inability to continue processing a transaction, or configuration errors that cause a component to fail to start.
Warning	Errors and warnings are logged. Warnings are less severe but more verbose than errors, in that a warning should be logged when the system was able to handle an unexpected input or condition without any threat to the operation of the server. Examples of warnings include invalid packet contents or failures in contacting remote servers.
Standard	Errors, warnings, and standard messages are logged. Standard logging messages show events as a result of normal operation.
Detail	Messages in the log are shown at event levels error, warning, standard, and detail. Detail logging is intended to inform why and how the particular result indicated by standard logging was reached. Server components that perform significant processing on the transaction, such as determining validity of the packet contents, log details about decisions they made. All server components that route the transaction through different processing based on the nature of the transaction log their routing activity at this level. The detail log is allowed to refer to the contents of messages logged at the standard level; that is, it will never be read without the standard messages.
Debug	Messages in the log are shown at event levels error, warning, standard, detail, and debug. Debug logging is provided for engineering troubleshooting only.

Log Groups

Log groups specify the type of server functionality for which you want to log events. The log groups listed in [Table 29 on page 467](#) are available.

Table 29: SIC Log Groups

Log Group	Description
Administration	<p>Reports events related to server administration, such as:</p> <ul style="list-style-type: none"> • A server access log, including identity of the administrator. This is available using the standard event level. • Changes made to the server configuration, including identity of the administrator. This is available using the Detail event level.
Configuration	Reports events related to configuration.
Packet	Reports events related to transaction processing.
PacketTrace	Displays content of a packet in an <attribute name>:<attribute value> format.
PacketTraceRaw	Displays content of a packet in its raw (octets) format.
System	<p>Reports events related to the system, such as:</p> <ul style="list-style-type: none"> • Resource failures (no memory, file not found, disk full, and so on.) • Unknown exceptions • System start • System stop

Related Documentation

- [Configuring Event Logging for an SIC Server \(SRC CLI\) on page 515](#)
- [Configuring SNMP for the SIC Group \(SRC CLI\) on page 519](#)
- [SNMP Support for the SIC Overview \(SRC CLI\) on page 467](#)

SNMP Support for the SIC Overview (SRC CLI)

The Simple Network Management Protocol (SNMP) implementation for the SIC supports alerts and is based on an external SNMP agent that accesses the SIC server agent.

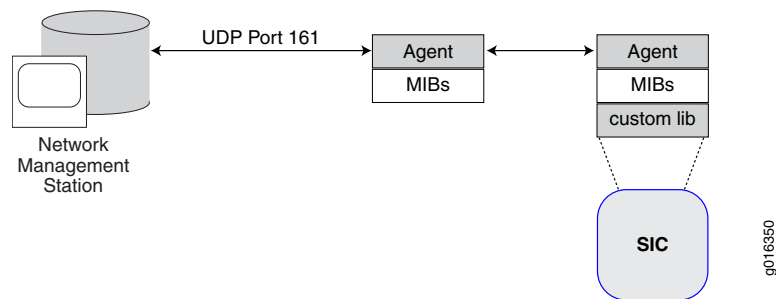


NOTE: You can query status using an SNMP Get command, and you can receive alert notifications through SNMP traps from the agent manager. However, you cannot configure the SIC server using SNMP because the SIC does not support the SNMP Set command.

SNMP support for the SIC functions in the same manner as other SRC components. The SIC has its own subagent, which communicates with the main SRC SNMP daemon using the AgentX protocol. The subagent registers with the main daemon for all relevant object identifiers (OIDs). Traps are communicated from the subagent to the main daemon.

[Figure 77 on page 468](#) depicts the SIC SNMP support.

Figure 77: SNMP Support for the SIC



Use the **show** command to view all statistics counters available to the SNMP daemon. [Table 30 on page 468](#) list the MIBs used by the SIC to maintain accounting statistics.

Table 30: MIBs Used by the SIC for SNMP Statistics

MIB	Description
RFC4670.mib	Maintains accounting client statistics.
RFC4671.mib	Maintains accounting server statistics.

The SIC supports the traps described in [Table 31 on page 468](#).

Table 31: SNMP Traps Supported for the SIC

SNMP Trap	Description
diameter-base-protocol-error	Diameter base protocol error occurred.
diameter-peer-connection-down	Diameter peer connection is down.
diameter-permanent-failure	Diameter permanent failure occurred.
diameter-transient-failure	Diameter transient failure occurred.
sic-server-internal-error	An SIC server implementation—dependent error occurred.
sic-server-log-file-failure	Operation on an SIC server log file such as opening, reading, or writing failed. This is most likely due to the server disk being out of space.
sic-server-resource-failure	An SIC server implementation—dependent resource failure occurred.

Table 31: SNMP Traps Supported for the SIC (*continued*)

SNMP Trap	Description
sic-server-shutdown	The SIC server process stopped successfully.
sic-server-startup	The SIC server process started successfully.
sic-server-unauthorized-administration-request	HTTP/HTTPs requests sent to the SIC server were denied because the user does not have proper permission to access the URL.

**Related
Documentation**

- [Configuring SNMP for the SIC Group \(SRC CLI\) on page 519](#)
- [Configuring Event Logging for an SIC Server \(SRC CLI\) on page 515](#)
- [SIC Event Logging Overview \(SRC CLI\) on page 464](#)

CHAPTER 33

Configuring the Subscriber Information Collector with the SRC CLI

- [SIC Configuration Summary on page 472](#)
- [Configuring the Connection Between the SIC and the Juniper Networks Database \(SRC CLI\) on page 474](#)
- [Creating an SIC Group and Server \(SRC CLI\) on page 476](#)
- [Creating an SIC Server Instance \(SRC CLI\) on page 477](#)
- [Configuring Dictionaries for the SIC Group \(SRC CLI\) on page 478](#)
- [Configuring the Device Models Supported by the SIC Group \(SRC CLI\) on page 480](#)
- [Configuring the RADIUS Accounting Listener for the SIC Group \(SRC CLI\) on page 480](#)
- [Configuring the RADIUS Authentication Listener for the SIC Group \(SRC CLI\) on page 482](#)
- [Configuring the Outbound RADIUS Transport of the SIC Group \(SRC CLI\) on page 484](#)
- [Configuring the RADIUS Transport for an SIC Server \(SRC CLI\) on page 485](#)
- [Configuring the SIC Diameter Server \(SRC CLI\) on page 486](#)
- [Configuring Upstream and Downstream RADIUS Network Elements \(SRC CLI\) on page 490](#)
- [Configuring What Realms Are Local to the SIC Group \(SRC CLI\) on page 501](#)
- [Configuration Statements for SIC Editing Rules \(SRC CLI\) on page 502](#)
- [Configuring the Optional Editing Rules Used by the SIC Group \(SRC CLI\) on page 505](#)
- [Configuring the Accounting Method Used by the SIC Group \(SRC CLI\) on page 507](#)
- [Configuring the Authentication Target Used by the SIC Server \(SRC CLI\) on page 510](#)
- [Configuring Request Routing \(SRC CLI\) on page 510](#)
- [Configuring Event Logging for an SIC Server \(SRC CLI\) on page 515](#)
- [Configuring SNMP for the SIC Group \(SRC CLI\) on page 519](#)
- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 520](#)
- [Configuring the NAS Groups \(SRC CLI\) on page 531](#)
- [Configuring the SAE to Manage AAA Devices on page 535](#)
- [Configuring AAA Policies \(SRC CLI\) on page 537](#)

SIC Configuration Summary

- [SIC RADIUS Configuration Summary \(SRC CLI\) on page 472](#)
- [SIC RADIUS Dynamic Authorization Configuration Summary \(SRC CLI\) on page 473](#)
- [SIC Diameter Configuration Summary \(SRC CLI\) on page 474](#)
- [Configuring Management of RADIUS-Enabled Devices for the SIC \(SRC CLI\) on page 474](#)

SIC RADIUS Configuration Summary (SRC CLI)

The SIC default configuration satisfies the needs of most environments, with minor changes such as accounting and authentication targets, editing rules, routing rules, and RADIUS clients.

To configure the SIC RADIUS options:

1. Use the default settings for the connection between the SIC and the Juniper Networks database, or configure your own connection.
[See “Configuring the Connection Between the SIC and the Juniper Networks Database \(SRC CLI\)” on page 474.](#)
2. Use the default SIC group, or create an SIC group or server.
[See “Creating an SIC Group and Server \(SRC CLI\)” on page 476.](#)
3. Use the default dictionary for the SIC group, or define dictionary changes.
[See “Configuring Dictionaries for the SIC Group \(SRC CLI\)” on page 478.](#)
4. Use the default device model (default-model), or configure the device models used by upstream and downstream network elements.
[See “Configuring the Device Models Supported by the SIC Group \(SRC CLI\)” on page 480.](#)
5. (Optional) Configure editing rules for the SIC group.
[See “Configuring the Optional Editing Rules Used by the SIC Group \(SRC CLI\)” on page 505.](#)
6. Configure any realms you want processed by the local server.
[See “Configuring What Realms Are Local to the SIC Group \(SRC CLI\)” on page 501.](#)
7. Configure the accounting listener for the SIC group.
[See “Configuring the RADIUS Accounting Listener for the SIC Group \(SRC CLI\)” on page 480.](#)
8. Configure the authentication listener for the SIC group.
[See “Configuring the RADIUS Authentication Listener for the SIC Group \(SRC CLI\)” on page 482.](#)
9. Configure outbound transport for the SIC group.
[See “Configuring the Outbound RADIUS Transport of the SIC Group \(SRC CLI\)” on page 484.](#)

10. Configure the upstream and downstream network elements.

For the upstream network element, configure:

- Supported device models
- Accounting client
- Authentication client
- Dynamic authorization target and failover policy

For the downstream network element, configure:

- Supported device models
- Accounting target and failover policy and mode
- Authentication target and failover policy and mode

See [“Configuring Upstream and Downstream RADIUS Network Elements \(SRC CLI\)”](#) on page 490.

11. Configure the accounting method used by the SIC group.

See [“Configuring the Accounting Method Used by the SIC Group \(SRC CLI\)”](#) on page 507.

12. (Optional) Create additional server instances as desired.

See [“Creating an SIC Server Instance \(SRC CLI\)”](#) on page 477.

13. (Optional) Configure request routing.

See [“Configuring Request Routing \(SRC CLI\)”](#) on page 510.

14. Configure transport options for each SIC server in the group.

See [“Configuring the RADIUS Transport for an SIC Server \(SRC CLI\)”](#) on page 485.

15. Configure event logging.

See [“Configuring Event Logging for an SIC Server \(SRC CLI\)”](#) on page 515.

16. (Optional) Configure SNMP options.

See [“Configuring SNMP for the SIC Group \(SRC CLI\)”](#) on page 519.

SIC RADIUS Dynamic Authorization Configuration Summary (SRC CLI)

To configure RADIUS dynamic authorization support, you must configure the device and service templates:

- Review the device and service template configuration overview.

See [“Device and Service Template Configuration Overview \(SRC CLI\)”](#) on page 539.

- Configure the device template used by the SIC group.

See [“Configuring Device Templates \(SRC CLI\)”](#) on page 545.

- Configure the device capabilities.

See [“Configuring the Device Capabilities Supported in the Device Template \(SRC CLI\)”](#) on page 546.

- Configure the service template.

See [“Configuring SIC Service Templates \(SRC CLI\)”](#) on page 549.

- Configure any tagged attributes for the service template.

See [“Configuring Tagged Attributes in SIC Service Templates \(SRC CLI\)”](#) on page 558.

- Configure the global service template.

See [“Configuring Global Service Templates \(SRC CLI\)”](#) on page 565.

SIC Diameter Configuration Summary (SRC CLI)

To configure Diameter support for the SIC:

1. Configure the SIC Diameter server including the Diameter network element failover policy and the Diameter peers.

See [“Configuring the SIC Diameter Server \(SRC CLI\)”](#) on page 486.

2. Configure the Diameter application.

See *Configuring the Diameter Application (SRC CLI)*.

3. Configure the SRC Diameter server.

See *Configuring Diameter Peers (SRC CLI)*.

Configuring Management of RADIUS-Enabled Devices for the SIC (SRC CLI)

To configure management of RADIUS-enabled devices when using the SIC:

1. Configure the NAS group peers, device capabilities, and routes.

See [“Configuring the NAS Groups \(SRC CLI\)”](#) on page 531.

2. Configure the SAE to manage SAE devices.

See [“Configuring the SAE to Manage AAA Devices”](#) on page 535.

3. Configure the AAA policy rules.

See [“Configuring AAA Policies \(SRC CLI\)”](#) on page 537.

Configuring the Connection Between the SIC and the Juniper Networks Database (SRC CLI)

The configuration of the subscriber information collector (SIC) is stored in the Juniper Networks database.

Use the following statements to configure the connection between the SIC and the Juniper Networks database:

```
slot number sic initial directory-connection {
```

```

url url ;
port port ;
principal principal ;
credentials credentials ;
entry-dn entry-dn ;
filter filter ;
}

```

To configure the directory connection properties for the SIC:

1. From configuration mode, access the statement that configures the directory configuration for the SIC in a slot.

```
user@host# edit slot number sic initial directory-connection
```

For example:

```
user@host# edit slot 0 sic initial directory-connection
```

2. Specify the password with which the SIC accesses the directory.

```
[edit slot 0 sic initial directory-connection]
user@host# set credentials credentials
```

3. (Optional) Specify the URL that identifies the location of the primary directory server.

```
[edit slot 0 sic initial directory-connection]
user@host# set url url
```

On a C Series Controller, this value is [ldap://127.0.0.1:389](#).

4. (Optional) Specify the port to use when connecting to the Juniper Networks database.

```
[edit slot 0 sic initial directory-connection]
user@host# set port port
```

5. (Optional) Specify the DN that contains the username that the directory server uses to authenticate the SIC.

```
[edit slot 0 sic initial directory-connection]
user@host# set principal principal
```

6. (Optional) Specify where the root of the SIC configuration is in the directory.

```
[edit slot 0 sic initial directory-connection]
user@host# set entry-dn entry-dn
```

7. (Optional) Specify any query filters you want to use to monitor changes in the Juniper Networks database.

```
[edit slot 0 sic initial directory-connection]
user@host# set filter filter
```

8. (Optional) Verify your configuration.

```
[edit slot 0 sic initial directory-connection]
user@host# show
url ldap://127.0.0.1:389/;
principal cn=conf,o=Operators,<base>;
credentials *****;
```

**Related
Documentation**

- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 520](#)
- *Configuring Initial Directory Eventing Properties for SRC Components*
- *Verifying the Local Configuration for a Component*

Creating an SIC Group and Server (SRC CLI)

The SIC group configuration controls the properties of accounting and authentication targets, dictionaries, editing rules, and RADIUS and Diameter options.

To create an SIC group and the associated server:

- From configuration mode, access the configuration statement that creates an SIC group.

```
[edit]
user@host# edit slot 0 server
```

For example, if you want to create an SIC group named `server-group1` that includes a server named `server-bldg5`, from configuration mode:

- Specify the *group-name* and *server-name*.

```
[edit]
user@host# edit slot 0 sic server
set name /server-group1/server-bldg5
```

The following rules depict how a new SIC group or server configuration is created on successfully committing the configuration:

- If the **group-name** does not exist in the Juniper Networks database, a new group and server instance as specified in this statement are created and populated with sample data.
- If the **group-name** already exists in the Juniper Networks database, a server instance as specified in this statement is created under the group and populated with sample data.

If you want to add another server to `server-group1` named `server-bldg5a`, execute:

```
[edit]
user@host# edit slot 0 sic server
set name /server-group1/server-bldg5a
```

Creating a server by using this statement populates it with sample data. You can also add a new server to an existing group by using the shared SIC group **shared sic group**

identifier server identifier statement. However, this statement does not populate the server with sample data.

**Related
Documentation**

- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 520](#)
- [Local and Shared Configurations for the SIC \(SRC CLI\) on page 445](#)
- [Creating an SIC Server Instance \(SRC CLI\) on page 477](#)

Creating an SIC Server Instance (SRC CLI)

Use either of the following statements to configure an SIC server instance:

```
slot number sic server {  
    name /group-name/server-name;  
}  
shared sic group identifier server identifier
```

To create an instance of an SIC server:

- From configuration mode, access the statement that configures the SIC server.

```
[edit]  
user@host# edit slot 0 sic server  
set name /group-name/server-name
```

For example, if you want to create an SIC group named **server-group1** that includes a server named **server-bldg5**, from configuration mode:

- Specify the *group-name* and *server-name*.

```
[edit]  
user@host# edit slot 0 sic server  
set name /server-group1/server-bldg5
```

If you want to add another server to **server-group1** named **server-bldg5a**, execute:

```
[edit]  
user@host# edit slot 0 sic server  
set name /server-group1/server-bldg5a
```

Creating a server by using this statement populates it with sample data.

You can also add a new server to an existing group by using the **shared sic group identifier server identifier** statement. However, this statement does not populate the server with sample data.

**Related
Documentation**

- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 520](#)
- [Local and Shared Configurations for the SIC \(SRC CLI\) on page 445](#)
- [Creating an SIC Group and Server \(SRC CLI\) on page 476](#)
- [Configuring Event Logging for an SIC Server \(SRC CLI\) on page 515](#)

Configuring Dictionaries for the SIC Group (SRC CLI)

You can add new attributes or modify the current attributes in an SIC dictionary. To add or modify an attribute in a dictionary, specify the unique name of the attribute and configure the RADIUS properties of the attribute.



NOTE: To create a new dictionary, we recommend that you work with Juniper Networks Technical Support.

Use the following statements to configure attributes in an SIC dictionary:

```
shared sic group identifier dictionary id
```

```
shared sic group identifier dictionary id attribute id
```

```
shared sic group identifier dictionary id attribute id radius {  
    type type;  
    format (one-byte-integer | integer | eight-byte-integer | string | ipv4-address |  
        ipv6-address | time | octets);  
    vendor-id vendor-id;  
    encrypt;  
    salt-encrypt;  
    tagged;  
    sensitive;  
}
```

```
shared sic group identifier dictionary id attribute id radius constant constant-name {  
    constant-value;  
}
```

To add or modify attributes in an SIC dictionary:

1. From configuration mode, access the statement that specifies the unique name for the dictionary. This sample procedure uses `group1` as the group identifier and `dic1` as the dictionary identifier.

```
[edit]  
user@host# edit shared sic group group1 dictionary dic1
```

2. Specify the unique name for the attribute you want to add or modify in the dictionary.

```
[edit shared sic group group1 dictionary dic1]  
user@host# edit attribute id
```

3. Specify that the attribute is a RADIUS attribute.

```
[edit shared sic group group1 dictionary dic1 attribute id]  
user@host# edit radius
```

4. Specify the attribute type.

```
[edit shared sic group group1 dictionary dic1 attribute attribute1 radius]
user@host# set type type
```

5. Specify the format of the RADIUS attribute.

```
[edit shared sic group group1 dictionary dic1 attribute attribute1 radius]
user@host# set format (one-byte-integer | integer | eight-byte-integer | string |
  ipv4-address | ipv6-address | time | octets)
```

where:

- one-byte-integer—Attribute value is an 8-bit unsigned integer.
- integer—Attribute value is a 32-bit unsigned integer.
- eight-byte-integer—Attribute value is a 64-bit unsigned integer.
- string—Attribute value is a string.
- ipv4-address—Attribute value is an IPv4 address.
- ipv6-address—Attribute value is an IPv6 address.
- time—Attribute value is a 32-bit unsigned value, with the most significant octet appearing first. The value is equal to the number of seconds since 00:00:00 UTC, January 1, 1970.
- octets—Attribute value consists of raw bytes.

6. (Optional) Specify the vendor ID for the attribute.

```
[edit shared sic group group1 dictionary dic1 attribute attribute1 radius]
user@host# set vendor-id vendor-id
```

7. (Optional) Specify whether the attribute should be encrypted without the salt.

```
[edit shared sic group group1 dictionary dic1 attribute attribute1 radius]
user@host# set encrypt
```

8. (Optional) Specify whether the attribute should be encrypted with the salt.

```
[edit shared sic group group1 dictionary dic1 attribute attribute1 radius]
user@host# set salt-encrypt
```

9. (Optional) Specify whether the RADIUS attribute is tagged.

```
[edit shared sic group group1 dictionary dic1 attribute attribute1 radius]
user@host# set tagged
```

10. (Optional) Specify whether the RADIUS attribute carries sensitive data, so its value will not be logged.

```
[edit shared sic group group1 dictionary dic1 attribute attribute1 radius]
user@host# set sensitive
```

11. (Optional) Specify the name and value of the constant you want to associate with the data contained in the RADIUS attribute.

```
[edit shared sic group group1 dictionary dic1 attribute attribute1 radius constant]
user@host# set constant-name constant-name constant-value
```

12. (Optional) If you modify an existing dictionary, you need to restart the SIC.

```
user@host# restart component sic
```

**Related
Documentation**

- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 520](#)
- [Configuring the Device Models Supported by the SIC Group \(SRC CLI\) on page 480](#)
- [SIC Dictionaries and Device Models Overview \(SRC CLI\) on page 463](#)

Configuring the Device Models Supported by the SIC Group (SRC CLI)

To configure the device models supported by the SIC group:

1. From configuration mode, access the statement that configures the device models supported by the SIC group. For example, to configure the device associated with the model name dm1 for the group group1:

```
[edit]
user@host# edit shared sic group group1 model dm1
```

2. Specify the name of the dictionary used by the device model.

```
[edit shared sic group group1 model dm1]
user@host# set dictionary dictionary
```

**Related
Documentation**

- [SIC Dictionaries and Device Models Overview \(SRC CLI\) on page 463](#)
- [Configuring Dictionaries for the SIC Group \(SRC CLI\) on page 478](#)
- [Configuring Upstream and Downstream RADIUS Network Elements \(SRC CLI\) on page 490](#)
- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 520](#)

Configuring the RADIUS Accounting Listener for the SIC Group (SRC CLI)

The accounting listener listens for RADIUS accounting events and filters undesired events based on attachment session attributes. Complete the following tasks to configure the accounting listener:

1. [Configuring the RADIUS Accounting Listener Queue Limits \(SRC CLI\) on page 480](#)
2. [Configuring the RADIUS Accounting Listener Transport \(SRC CLI\) on page 481](#)

Configuring the RADIUS Accounting Listener Queue Limits (SRC CLI)

Use the following statements to configure the accounting listener queue limits:


```
shared sic group identifier radius accounting-listener limit {
  incoming-queue incoming-queue;
  transaction-queue transaction-queue;
}
```

To configure the RADIUS accounting listener queue limits:

1. From configuration mode, access the statement that configures the RADIUS accounting listener queue limits. For example, to configure the limits for a group called `group1`:

```
[edit]
user@host# edit shared sic group group1 radius accounting-listener limit
```

2. (Optional) Specify the incoming queue limit for the RADIUS accounting listener.

```
[edit shared sic group group1 radius accounting-listener limit]
user@host# set incoming-queue incoming-queue
```

3. (Optional) Specify the transaction queue limit for the RADIUS accounting listener.

```
[edit shared sic group group1 radius accounting-listener limit]
user@host# set transaction-queue transaction-queue
```

Configuring the RADIUS Accounting Listener Transport (SRC CLI)

Use the following statements to configure the RADIUS accounting listener transport:

```
shared sic group identifier radius accounting-listener transport
shared sic group identifier radius accounting-listener transport id {
  port port;
  connections-per-thread connections-per-thread;
  connect-timeout connect-timeout;
  disconnect-timeout disconnect-timeout;
}
```

1. From configuration mode, access the statement that configures the RADIUS accounting listener transport and specify a name for the transport. Each RADIUS accounting transport must have a unique name. For example to configure a transport called `acct-tran1`:

```
[edit]
user@host# edit shared sic group group1 radius accounting-listener transport acct-tran1
```

2. Specify the UDP port number of the accounting listener from which the server listens for RADIUS packets.

```
[edit shared sic group group1 radius accounting-listener transport acct-tran1]
user@host# set port port
```

3. (Optional) Specify the number of UDP connections per thread.

```
[edit shared sic group group1 radius accounting-listener transport acct-tran1]
```

```
user@host# set connections-per-thread connections-per-thread
```

4. (Optional) Specify the UDP connection timeout in milliseconds.

```
[edit shared sic group group1 radius accounting-listener transport acct-tran1]  
user@host# set connect-timeout connect-timeout
```

5. (Optional) Specify the UDP disconnection timeout in milliseconds.

```
[edit shared sic group group1 radius accounting-listener transport acct-tran1]  
user@host# set disconnect-timeout disconnect-timeout
```

Related Documentation

- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 520](#)
- [Configuring the Outbound RADIUS Transport of the SIC Group \(SRC CLI\) on page 484](#)
- [Configuring the RADIUS Transport for an SIC Server \(SRC CLI\) on page 485](#)

Configuring the RADIUS Authentication Listener for the SIC Group (SRC CLI)

The authentication listener listens for RADIUS authentication messages and filters undesired events based on attachment session attributes. Complete the following tasks to configure the authentication listener:

1. [Configuring the RADIUS Authentication Listener Queue Limits \(SRC CLI\) on page 482](#)
2. [Configuring the RADIUS Authentication Listener Transport \(SRC CLI\) on page 483](#)

Configuring the RADIUS Authentication Listener Queue Limits (SRC CLI)

Use the following statements to configure the RADIUS authentication listener queue limit:

```
shared sic group identifier radius authentication-listener limit {  
    incoming-queue incoming-queue;  
    transaction-queue transaction-queue;  
}
```

To configure the RADIUS authentication listener queue limits:

1. From configuration mode, access the statement that configures the RADIUS authentication listener queue limits. For example, to configure the limits for a group called `group1`:

```
[edit]  
user@host# edit shared sic group group1 radius authentication-listener limit
```

2. (Optional) Specify the incoming queue limit for the RADIUS authentication listener.

```
[edit shared sic group group1 radius authentication-listener limit]  
user@host# set incoming-queue incoming-queue
```

3. (Optional) Specify the transaction queue limit for the RADIUS authentication listener.

```
[edit shared sic group group1 radius authentication-listener limit]
user@host# set transaction-queue transaction-queue
```

Configuring the RADIUS Authentication Listener Transport (SRC CLI)

Use the following statements to configure the RADIUS authentication listener transport:

```
shared sic group identifier radius authentication-listener transport
shared sic group identifier radius authentication-listener transport id {
  port port;
  connections-per-thread connections-per-thread;
  connect-timeout connect-timeout;
  disconnect-timeout disconnect-timeout;
}
```

1. From configuration mode, access the statement that configures the RADIUS authentication listener transport and specify a name for the transport. Each RADIUS authentication transport must have a unique name. For example, to configure a transport called `auth-tran1`:

```
[edit]
user@host# edit shared sic group group1 radius authentication-listener transport
auth-tran1
```

2. Specify the UDP port number of the authentication listener from which the server listens for RADIUS packets.

```
[edit shared sic group group1 radius authentication-listener transport auth-tran1]
user@host# set port port
```

3. (Optional) Specify the number of UDP connections per thread.

```
[edit shared sic group group1 radius authentication-listener transport auth-tran1]
user@host# set connections-per-thread connections-per-thread
```

4. (Optional) Specify the UDP connection timeout in milliseconds.

```
[edit shared sic group group1 radius authentication-listener transport auth-tran1]
user@host# set connect-timeout connect-timeout
```

5. (Optional) Specify the UDP disconnection timeout in milliseconds.

```
[edit shared sic group group1 radius authentication-listener transport auth-tran1]
user@host# set disconnect-timeout disconnect-timeout
```

Related Documentation

- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 520](#)
- [Configuring the Outbound RADIUS Transport of the SIC Group \(SRC CLI\) on page 484](#)
- [Configuring the RADIUS Transport for an SIC Server \(SRC CLI\) on page 485](#)

Configuring the Outbound RADIUS Transport of the SIC Group (SRC CLI)

You can use the RADIUS outbound transport to control communication to accounting targets that reside in a downstream network element. You need to specify the UDP port, as well as connect and disconnect related configuration options of the SIC group.

Use the following statements to configure the outbound RADIUS transport of the SIC group:

```
shared sic group identifier radius outbound-transport transport-name
```

```
shared sic group identifier radius outbound-transport transport-name {  
  connections-per-thread connections-per-thread;  
  connect-timeout connect-timeout;  
  disconnect-timeout disconnect-timeout;  
  port port;  
  port-range-size port-range-size;  
}
```

To configure the outbound RADIUS transport of the SIC group:

1. From configuration mode, access the statement that configures the name for the outbound RADIUS transport of the SIC group. For example, to configure the outbound RADIUS transport called outtrp1 for the SIC group group1:

```
[edit]  
user@host# edit shared sic group group1 radius outbound-transport outtrp1
```

2. (Optional) Specify the number of UDP connections per thread.

```
[edit shared sic group group1 radius outbound-transport outtrp1]  
user@host# set connections-per-thread connections-per-thread
```

3. (Optional) Specify the UDP connection timeout in milliseconds.

```
[edit shared sic group group1 radius outbound-transport outtrp1]  
user@host# set connect-timeout connect-timeout
```

4. (Optional) Specify the UDP disconnection timeout in milliseconds.

```
[edit shared sic group group1 radius outbound-transport outtrp1]  
user@host# set disconnect-timeout disconnect-timeout
```

5. Specify the UDP port number starting from which the server sends the RADIUS packets.

```
[edit shared sic group group1 radius outbound-transport outtrp1]  
user@host# set port port
```

6. (Optional) Specify the range of UDP ports that are used to send the RADIUS packets.

```
[edit shared sic group group1 radius outbound-transport outtrp1]  
user@host# set port-range-size port-range-size
```

- Related Documentation**
- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 520](#)
 - [RADIUS and Diameter Transports on page 462](#)
 - [Configuring the RADIUS Transport for an SIC Server \(SRC CLI\) on page 485](#)
 - [Configuring Downstream Network Elements and Accounting and Authentication Targets \(SRC CLI\) on page 496](#)

Configuring the RADIUS Transport for an SIC Server (SRC CLI)

You need to configure both the inbound and outbound RADIUS transport for each server in the SIC group. Servers use the same inbound and outbound transport names that are configured for the SIC group.

Use the following statements to configure the RADIUS transport options for the SIC server:

```
shared sic group identifier server identifier transport transport-name
```

```
shared sic group identifier server identifier transport transport-name {
  address address;
}
```

```
shared sic group identifier server identifier outbound-transport transport-name
```

```
shared sic group identifier server identifier outbound-transport transport-name {
  address address;
}
```

To configure the RADIUS transport options for the SIC server:

1. From configuration mode, access the statement that configures the RADIUS inbound transport options for the server. For example, if the accounting listener transport for the group is configured as `trpin1`, specify the server inbound transport as `trpin1`.

```
[edit]
user@host# edit shared sic group group1 server server1 transport trpin1
```

2. (Optional) Specify the IP address used by the server for receiving UDP packets.

```
[edit shared sic group group1 server server1 transport trpin1]
user@host# set address address
```

3. Specify the RADIUS outbound transport options for the SIC server. For example, if the outbound transport for the SIC group is set to `trpout1`, set the server outbound transport to `trpout1`.

```
[edit]
user@host# edit shared sic group group1 server server1 outbound-transport trpout1
```

4. (Optional) Specify the IP address used by the server when sending outbound requests.

```
[edit shared sic group group1 server server1 outbound-transport trpout1]
user@host# set address address
```

**Related
Documentation**

- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 520](#)
- [RADIUS and Diameter Transports on page 462](#)
- [Managing Dynamic Services on page 434](#)
- [Configuring the Outbound RADIUS Transport of the SIC Group \(SRC CLI\) on page 484](#)
- [Creating an SIC Server Instance \(SRC CLI\) on page 477](#)

Configuring the SIC Diameter Server (SRC CLI)

- [Configuration Statements for the SIC Diameter Server \(SRC CLI\) on page 486](#)
- [Configuring the SIC Diameter Server Identity \(SRC CLI\) on page 487](#)
- [Configuring the SIC Diameter Server Peer \(SRC CLI\) on page 488](#)

Configuration Statements for the SIC Diameter Server (SRC CLI)

Use the following statements to configure the SIC Diameter server:

```
shared sic group identifier server identifier diameter identity {
  origin-host origin-host;
  origin-realm origin-realm;
}
shared sic group identifier server identifier diameter transport id {
  protocol (tcp | sctp);
  port port;
}
shared sic group identifier diameter network-element id {
  description description;
  failover-policy (round-robin | primary-backup);
}
shared sic group identifier diameter network-element id peer name {
  description description;
  address address;
  protocol (tcp | sctp);
  port port;
  active-peer;
  priority priority;
}
shared sic group identifier diameter network-element id peer name {
  enforce-source-address;
}
shared sic group identifier diameter network-element id peer name {
  origin-host origin-host;
}
shared sic group identifier diameter network-element id peer name addresses address
  address
```

Configuring the SIC Diameter Server Identity (SRC CLI)

Configuring the SIC Diameter server identity includes specifying the origin-host, origin-realm, the port the server receives Diameter messages on, and protocol. The SIC Diameter server communicates with the SRC Diameter server. The origin-host and origin-realm identify the SIC Diameter server. This identity is sent in all Diameter requests originating on this server.

The default identity of the SIC Diameter server is set to origin-host="your-host" and the origin-realm="your-realm.net." You must reconfigure these settings for your network environment.

To configure the SRC Diameter server and the Diameter application, see *Configuring the Diameter Application (SRC CLI)* and *Configuring Diameter Peers (SRC CLI)*.

Use the following statements to configure the SIC Diameter server identity:

```
shared sic group identifier server identifier diameter identity {
    origin-host origin-host;
    origin-realm origin-realm;
}
shared sic group identifier server identifier diameter transport id {
    protocol (tcp | sctp);
    port port;
}
```

To configure the SIC Diameter server identity:

1. From configuration mode, access the statement that configures the SIC Diameter server. For example, to configure the SIC Diameter server in an SIC group called g1 that includes an SIC server called svr1:

```
[edit]
user@host# shared sic group g1 server svr1 diameter identity
```

2. Specify the origin-host name of the SIC Diameter server. For example, to specify the origin-host as sic-diam-svr1:

```
[edit shared sic group g1 server svr1 diameter identity]
user@host# set origin-host sic-diam-svr1
```

3. Specify the origin-realm name of the SIC Diameter server. For example, to specify the origin-realm as abc.com:

```
[edit shared sic group g1 server svr1 diameter identity]
user@host# set origin-realm abc.com
```

4. Verify your configuration.

```
[edit shared sic group g1 server svr1 diameter identity]
user@host# show
```

```
user@host# show
origin-host diam-svr1;
origin-realm abc.com;
```

Configuring the SIC Diameter Server Peer (SRC CLI)

The SIC Diameter server handles all communication between the SIC and the SRC Diameter server. This procedure describes how to configure the network element in which the SRC Diameter server logically resides, the failover policy, and the Diameter connection between the SIC Diameter server and the SRC Diameter server.

Use the following statements to configure the SIC Diameter peer:

```
shared sic group identifier diameter network-element id {
  description description;
  failover-policy (round-robin | primary-backup);
}
shared sic group identifier diameter network-element id peer name {
  description description;
  address address;
  protocol (tcp | sctp);
  port port;
  active-peer;
  priority priority;
}
shared sic group identifier diameter network-element id peer name {
  enforce-source-address;
}
shared sic group identifier diameter network-element id peer name {
  origin-host origin-host;
}
shared sic group identifier diameter network-element id peer name addresses address
address
```

To configure the SIC Diameter server peer:

1. From configuration mode, access the statement that configures the SIC Diameter server peer and configure the network element where the SRC Diameter server resides. For example, to configure a Diameter network element called `diam-ne1` for an SIC group called `g1`:

```
[edit]
user@host# shared sic group g1 diameter network-element diam-ne1
```

2. (Optional) Specify a description for the network element.

```
[shared sic group g1 diameter network-element diam-ne1]
user@host# set description description
```

3. (Optional) Configure the failover policy for the network element. For example, to configure the primary or backup failover policy:

```
[shared sic group g1 diameter network-element diam-ne1]
user@host# set primary-backup
```


4. Configure the name of the Diameter peer (SRC Diameter server). For example, to call the peer `src-diam-svr1`:

```
[shared sic group g1 diameter network-element diam-ne1]  
user@host# edit peer src-diam-svr1
```

5. (Optional) Specify a description for the Diameter peer.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]  
user@host# set description description
```

6. Specify the IP address of the remote Diameter peer (SRC Diameter server). For example, `10.1.2.3`.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]  
user@host# set address 10.1.2.3
```

7. Specify the protocol the Diameter peer (SRC Diameter server) uses for Diameter messages (TCP or SCTP).

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]  
user@host# set protocol sctp
```

8. Specify which port the Diameter peer (SRC Diameter server) receives messages on. For example, port `2222`.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]  
user@host# set port 2222
```

9. (Optional) Specify whether the peer is active or not. If the peer is configured to connect actively, the server periodically attempts to connect (or reconnect after a connection has failed) to the remote peer. If this option is not set, a connection is established only after the remote peer attempts to connect to this server.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]  
user@host# set active-peer
```

10. (Optional) Specify the priority of the peer for the failover policy. Peers with lower priority values are the preferred routing targets for Diameter requests. Requests are split equally among peers with the same priority level.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]  
user@host# set priority 1
```

11. (Optional) Specify whether a source IP match is required for the connection. This option determines whether the source IP address of a connection attempt must match one of the configured IP addresses used to connect to this peer. If this option is not set, requests are accepted from any IP address as long as the client presents the correct host name during the capabilities exchange. This functionality allows other peers to exist behind NAS devices.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]  
user@host# set enforce-source-address
```

12. Specify the origin-host name of the Diameter peer (SRC Diameter server). For example, if the origin-host name of the SRC Diameter server is `diam-host1`:

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host# set origin-host diam-host1
```

13. (Optional) Specify an ordered set of IP addresses to use for a multilink connection. An IP address of the remote peer is necessary to establish a Diameter connection with the remote peer (SRC Diameter server). For a Diameter connection over TCP, only one configured address is used. Over SCTP, the connection may be established over multiple addresses.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host# set addresses address 10.1.2.4
user@host# set addresses address 10.1.2.5
user@host# set addresses address 10.1.2.6
```

14. Verify your configuration.

```
[shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host# show
```

```
active-peer;
address 10.1.2.3;
addresses {
  address 10.1.2.4;
  address 10.1.2.5;
  address 10.1.2.6;
}
port 3868;
priority 1;
protocol sctp;
enforce-source-address;
origin-host diam-host1;
```

```
[edit shared sic group g1 diameter network-element diam-ne1 peer src-diam-svr1]
user@host#
```

Configuring Upstream and Downstream RADIUS Network Elements (SRC CLI)

- [Configuration Statements for Downstream Network Elements and Accounting and Authentication Targets \(SRC CLI\) on page 491](#)
- [Configuration Statements for Upstream Network Elements, Accounting and Authentication Clients, and Dynamic Authorization Targets \(SRC CLI\) on page 492](#)
- [Creating a Network Element \(SRC CLI\) on page 492](#)
- [Configuring the Device Models Supported in the Network Element \(SRC CLI\) on page 493](#)
- [Configuring Upstream Network Elements and Accounting and Authentication Clients \(SRC CLI\) on page 494](#)
- [Configuring Upstream Network Elements and Dynamic Authorization Targets \(SRC CLI\) on page 495](#)

- [Configuring Downstream Network Elements and Accounting and Authentication Targets \(SRC CLI\) on page 496](#)
- [Configuration Statements for SIC Group Failover Mode and Policy \(SRC CLI\) on page 498](#)
- [Configuring Failover Mode and Policy \(SRC CLI\) on page 499](#)

Configuration Statements for Downstream Network Elements and Accounting and Authentication Targets (SRC CLI)

Use the following statements to configure downstream RADIUS network elements and accounting and authentication targets for the SIC group:

```
shared sic group identifier radius network-element id
```

```

shared sic group identifier radius network-element id downstream {
  model model;
}
shared sic group identifier radius network-element id downstream (authentication |
  accounting) {
  failover-mode (round-robin | primary-backup);
}
shared sic group identifier radius network-element id downstream (authentication |
  accounting) failover-policy
shared sic group identifier radius network-element id downstream (authentication |
  accounting) failover-policy fast-fail {
  minimum-number minimum-number;
  timeout timeout;
  reset-delay reset-delay;
}
shared sic group identifier radius network-element id downstream (authentication |
  accounting) failover-policy retry {
  number number;
  timeout timeout;
}
shared sic group identifier radius network-element id downstream (authentication |
  accounting) accounting-target name {
  address address;
  priority priority;
}
shared sic group identifier radius network-element id downstream (authentication |
  accounting) accounting-target name {
  secret secret;
  outbound-transport outbound-transport;
  port port;
}
shared sic group identifier radius network-element id downstream (authentication |
  accounting) authentication-target name {
  address address;
  priority priority;
}
shared sic group identifier radius network-element id downstream (authentication |
  accounting) authentication-target name {
  secret secret;
  outbound-transport outbound-transport;
  port port;

```

```
}
```

Configuration Statements for Upstream Network Elements, Accounting and Authentication Clients, and Dynamic Authorization Targets (SRC CLI)

Use the following statements to configure upstream RADIUS network elements, accounting and authentication clients, and dynamic authorization targets for the SIC group:

```
shared sic group identifier radius network-element id
shared sic group identifier radius network-element id upstream {
    model model;
}
shared sic group identifier radius network-element id upstream radius-client id {
    address address;
    accounting-secret accounting-secret;
    authentication-secret authentication-secret;
}
shared sic group identifier radius network-element id upstream dynamic-authorization-target
{
    failover-mode (round-robin | primary-backup);
}
shared sic group identifier radius network-element id upstream dynamic-authorization-target
    failover-policy
shared sic group identifier radius network-element id upstream dynamic-authorization-target
    failover-policy retry {
        number number;
        timeout timeout;
    }
shared sic group identifier radius network-element id upstream dynamic-authorization-target
    failover-policy fast-fail {
        minimum-number minimum-number;
        timeout timeout;
        reset-delay reset-delay;
    }
shared sic group identifier radius network-element id upstream dynamic-authorization-target
    target name {
        address address;
        priority priority;
    }
shared sic group identifier radius network-element id upstream dynamic-authorization-target
    target name {
        secret secret;
        port port;
    }
}
```

Creating a Network Element (SRC CLI)

Network elements are logical entities that are considered either upstream or downstream from the SIC. Upstream network elements contain logical clients and targets for NAS devices. Downstream network elements contain logical targets for the downstream AAA server responsible for accounting and authentication.

Use the following statement to create a network element:

```
shared sic group identifier radius network-element id
```

To create a network element:

- From configuration mode, access the statement that creates a RADIUS network element. For example, to create a network element called `ne1` for the SIC group `group1`:

```
[edit]
user@host# edit shared sic group group1 radius network-element ne1
```

Configuring the Device Models Supported in the Network Element (SRC CLI)

You must configure which device models are supported by the upstream and downstream network elements.



NOTE: To assign a device model to a network element, you must first configure the device models and the associated dictionaries supported by the SIC group using the `shared sic group identifier model id` statement. See “Configuring the Device Models Supported by the SIC Group (SRC CLI)” on page 480.

Use the following statements to configure the device model:

```
shared sic group identifier radius network-element id downstream {
  model model;
}
shared sic group identifier radius network-element id upstream {
  model model;
}
```

To configure the device models supported in the network element:

1. From configuration mode, access the statement that configures the RADIUS network element and specify a name for the network element. This sample procedure uses `group1` for the SIC group and `ne1` for the downstream network element identifier.

```
[edit]
user@host# edit shared sic group group1 radius network-element ne1 downstream
```

2. Specify a device model. The device model must have previously been configured for the SIC group.

```
[edit shared sic group group1 radius network-element ne1 downstream]
user@host# set model model
```

Configuring Upstream Network Elements and Accounting and Authentication Clients (SRC CLI)

Accounting and authentication clients are NAS devices that logically reside in upstream network elements. Accounting clients send RADIUS accounting requests to the SIC accounting listener. Authentication clients send RADIUS authentication requests to the SIC authentication listener. You must configure at least one accounting client and one authentication client. Each client must have a unique name and address.

Use the following statements to configure accounting clients:

```
shared sic group identifier radius network-element id upstream radius-client id {  
    address address;  
    accounting-secret accounting-secret;  
    authentication-secret authentication-secret;  
}
```

To configure RADIUS accounting and authentication clients:

1. From configuration mode, access the statement that configures an upstream network element and RADIUS client. For example, to configure an upstream RADIUS network element called `ne1` and RADIUS client called `rc1` for the SIC group `group1`:

```
[edit]  
user@host# edit shared sic group group1 radius network-element ne1 upstream  
radius-client rc1
```

2. (Optional) Specify the IP address of the RADIUS client.

```
[edit shared sic group group1 radius network-element ne1 upstream radius-client rc1]  
user@host# set address address
```

3. (Optional) Specify the shared secret used by the accounting client.

```
[edit shared sic group group1 radius network-element ne1 upstream radius-client rc1]  
user@host# set accounting-secret authentication-secret
```

4. Specify the shared secret used by the authentication client.

```
[edit shared sic group group1 radius network-element ne1 upstream accounting-client]  
user@host# set accounting-secret accounting-secret
```

Configuring Upstream Network Elements and Dynamic Authorization Targets (SRC CLI)

Dynamic authorization targets are logical entities that represent the NAS device in upstream network elements. The SIC forwards COA/DM requests to dynamic authorization targets.

Use the following statements to configure dynamic authorization targets:

```
shared sic group identifier radius network-element id upstream dynamic-authorization-target
  target name {
    address address;
    priority priority;
  }
shared sic group identifier radius network-element id upstream dynamic-authorization-target
  target name {
    secret secret;
    port port;
  }
shared sic group identifier radius network-element id upstream dynamic-authorization-target
  {
    failover-mode (round-robin | primary-backup);
  }
shared sic group identifier radius network-element id upstream dynamic-authorization-target
  failover-policy {
    priority priority;
  }
shared sic group identifier radius network-element id upstream dynamic-authorization-target
  failover-policy retry {
    number number;
    timeout timeout;
  }
shared sic group identifier radius network-element id upstream dynamic-authorization-target
  failover-policy fast-fail {
    minimum-number minimum-number;
    timeout timeout;
    reset-delay reset-delay;
  }
```

To configure a dynamic authorization target:

1. From configuration mode, access the statement that configures an upstream network element and dynamic authorization target. For example, to configure an upstream RADIUS network element called `ne1` and dynamic authorization target called `dat1` for the SIC group `group1`:

```
[edit]
user@host# edit shared sic group group1 radius network-element ne1 upstream
dynamic-authorization-target target dat1
```

2. Specify the IP address of the target.

```
[edit shared sic group group1 radius network-element ne1 upstream
dynamic-authorization-target target dat1]
user@host# set address address
```

3. Specify the priority of the target. Targets with lower priority values are selected before other targets in a failover policy.

```
[edit shared sic group group1 radius network-element ne1 upstream
dynamic-authorization-target target dat1]
user@host# set priority priority
```

4. Specify the shared secret used by the target.

```
[edit shared sic group group1 radius network-element ne1 upstream
dynamic-authorization-target target dat1]
user@host# set secret secret
```

5. (Optional) Specify the port used by the target to receive dynamic authorization messages.

```
[edit shared sic group group1 radius network-element ne1 upstream
dynamic-authorization-target target dat1]]
user@host# set port port
```

Configuring Downstream Network Elements and Accounting and Authentication Targets (SRC CLI)

Accounting and authentication targets (RADIUS AAA server) receive requests forwarded by the SIC. These targets reside in downstream network elements. You must configure at least one accounting target and one authentication target. Each target must have a unique name and address.

1. [Configuring SIC Accounting Targets \(SRC CLI\) on page 496](#)
2. [Configuring SIC Authentication Targets \(SRC CLI\) on page 497](#)

Configuring SIC Accounting Targets (SRC CLI)

Use the following statements to configure accounting targets:

```
shared sic group identifier radius network-element id downstream (authentication |
accounting) accounting-target name {
  address address;
  priority priority;
}
shared sic group identifier radius network-element id downstream (authentication |
accounting) accounting-target name {
  secret secret;
  outbound-transport outbound-transport;
  port port;
}
```

To configure an accounting target:

1. From configuration mode, access the statement that configures the accounting target. This sample procedure uses group1 for the group identifier, ne1 for the network element identifier, and target1 as the accounting target name.


```
edit shared sic group group1 radius network-element ne1 downstream accounting
accounting-target target1
```

2. Specify the IP address of the RADIUS accounting target contained in the network element.

```
[edit shared sic group group1 radius network-element ne1 downstream accounting
accounting-target target1]
user@host# set address address
```

3. Specify the priority of the target. Targets with lower priority values are selected before other targets in a failover policy.

```
[edit shared sic group group1 radius network-element ne1 downstream accounting
accounting-target target1]
user@host# set priority priority
```

4. Specify the shared secret used by the RADIUS accounting target.

```
[edit shared sic group group1 radius network-element ne1 downstream accounting
accounting-target target1]
user@host# set secret secret
```

5. (Optional) Specify the name of the local transport used to send requests to the accounting target.

```
[edit shared sic group group1 radius network-element ne1 downstream accounting
accounting-target target1]
user@host# set outbound-transport outbound-transport
```

6. (Optional) Specify the UDP port number on which the RADIUS accounting target listens for requests.

```
[edit shared sic group group1 radius network-element ne1 downstream accounting
accounting-target target1]
user@host# set port port
```

Configuring SIC Authentication Targets (SRC CLI)

Use the following statements to configure authentication targets:

```
shared sic group identifier radius network-element id downstream (authentication |
accounting) authentication-target name {
  address address;
  priority priority;
}
shared sic group identifier radius network-element id downstream (authentication |
accounting) authentication-target name {
  secret secret;
  outbound-transport outbound-transport;
  port port;
}
```

To configure an authentication target:

1. From configuration mode, access the statement that configures the authentication target. This sample procedure uses `group1` for the group identifier, `ne1` for the network element identifier, and `target1` as the authentication target name.

```
edit shared sic group group1 radius network-element ne1 downstream authentication  
authentication-target target1
```

2. Specify the IP address of the RADIUS authentication target contained in the network element.

```
[edit shared sic group group1 radius network-element ne1 downstream authentication  
authentication-target target1]  
user@host# set address address
```

3. Specify the priority of the target. Targets with lower priority values are selected before other targets in a failover policy.

```
[edit shared sic group group1 radius network-element ne1 downstream authentication  
authentication-target target1]  
user@host# set priority priority
```

4. Specify the shared secret used by the RADIUS authentication target.

```
[edit shared sic group group1 radius network-element ne1 downstream authentication  
authentication-target target1]  
user@host# set secret secret
```

5. (Optional) Specify the name of the local transport used to send outbound requests to the authentication target.

```
[edit shared sic group group1 radius network-element ne1 downstream authentication  
authentication-target target1]  
user@host# set outbound-transport outbound-transport
```

6. (Optional) Specify the UDP port number on which the RADIUS authentication target listens for requests.

```
[edit shared sic group group1 radius network-element ne1 downstream authentication  
authentication-target target1]  
user@host# set port port
```

Configuration Statements for SIC Group Failover Mode and Policy (SRC CLI)

Use the following statements to configure failover mode and policy:

```
shared sic group identifier radius network-element id downstream (authentication |  
accounting) {  
    failover-mode (round-robin | primary-backup);  
}  
shared sic group identifier radius network-element id downstream (authentication |  
accounting) failover-policy  
shared sic group identifier radius network-element id downstream (authentication |  
accounting) failover-policy fast-fail {  
    minimum-number minimum-number;
```

```

        timeout timeout;
        reset-delay reset-delay;
    }
    shared sic group identifier radius network-element id downstream (authentication |
        accounting) failover-policy retry {
        number number;
        timeout timeout;
    }
    shared sic group identifier radius network-element id upstream dynamic-authorization-target
    {
        failover-mode (round-robin | primary-backup);
    }
    shared sic group identifier radius network-element id upstream dynamic-authorization-target
        failover-policy
    shared sic group identifier radius network-element id upstream dynamic-authorization-target
        failover-policy retry {
        number number;
        timeout timeout;
    }
    shared sic group identifier radius network-element id upstream dynamic-authorization-target
        failover-policy fast-fail {
        minimum-number minimum-number;
        timeout timeout;
        reset-delay reset-delay;
    }
}

```

Configuring Failover Mode and Policy (SRC CLI)

You must configure failover mode and policy for accounting and authentication targets upstream by completing the following tasks:

1. [Configuring Failover Mode \(SRC CLI\) on page 499](#)
2. [Configuring Fast Fail Options for the Failover Policy on page 500](#)
3. [Configuring Retry Options for the Failover Policy on page 501](#)

Configuring Failover Mode (SRC CLI)

You must configure failover mode for both accounting and authentication messages. Use the following statement to configure failover mode:

```

shared sic group identifier radius network-element id downstream (authentication |
    accounting) {
    failover-mode (round-robin | primary-backup);
}

```

To configure failover mode:

1. From configuration mode, access the statement that configures the network element failover mode and specify whether the connection is for authentication or accounting messages.

For example, this sample procedure uses `group1` for the group identifier, `ne1` for the network element identifier, and `accounting` as the connection.

[edit]

```
user@host# edit shared sic group group1 radius network-element ne1 downstream  
accounting
```

2. Specify failover mode used by the network element.

```
[edit shared sic group group1 radius network-element ne1 downstream]  
user@host# set failover-mode (round-robin | primary-backup)
```

Where:

- **round-robin**—When this failover mode is used, messages are sent to the network element over alternating paths.
- **primary-backup**—When this failover mode is used, messages are sent over the primary path unless it is unavailable, in which case messages are sent over the backup path.

Configuring Fast Fail Options for the Failover Policy

You must configure fast fail options for the failover policy for both accounting and authentication messages. Use the following statement to configure fast fail options:

```
shared sic group identifier radius network-element id downstream (authentication |  
accounting) failover-policy  
shared sic group identifier radius network-element id downstream (authentication |  
accounting) failover-policy fast-fail {  
  minimum-number minimum-number;  
  timeout timeout;  
  reset-delay reset-delay;  
}
```

To configure fast fail options for the failover policy:

1. From configuration mode, access the statement that configures fast fail options for the failover policy. For example, this sample procedure uses `group1` for the group identifier, `ne1` for the network element identifier, and `accounting` as the connection type.

```
edit shared sic group group1 radius network-element ne1 downstream accounting  
failover-policy fast-fail
```

2. Specify the minimum number of times the message is retransmitted if an acknowledgment from the target is not received.

```
[edit shared sic group group1 radius network-element ne1 downstream accounting  
failover-policy fast-fail]  
user@host# set minimum-number minimum-number
```

3. Specify the time in seconds before the target is placed into fast fail mode.

```
[edit shared sic group group1 radius network-element ne1 downstream accounting  
failover-policy fast-fail]  
user@host# set timeout timeout
```

- Specify the time in seconds after which the target is taken out of fast fail mode.

```
[edit shared sic group group1 radius network-element ne1 downstream accounting
 failover-policy fast-fail]
user@host# set reset-delay reset-delay
```

Configuring Retry Options for the Failover Policy

You must configure retry options for the failover policy for both accounting and authentication messages. Use the following statement to configure retry options:

```
shared sic group identifier radius network-element id downstream (authentication |
 accounting) failover-policy retry {
  number number;
  timeout timeout;
}
```

To configure retry options for the failover policy:

- From configuration mode, access the statement that configures retry options for the failover policy. For example, this sample procedure uses `group1` for the group identifier, `ne1` for the network element identifier, and `accounting` as the connection type.

```
edit shared sic group group1 radius network-element ne1 downstream accounting
 failover-policy retry
```

- Specify the maximum number of times a message is retransmitted if an acknowledgment from the target is not received.

```
[edit shared sic group group1 radius network-element ne1 downstream accounting
 failover-policy retry]
user@host# set number number
```

- Specify the number of seconds between retry attempts.

```
[edit shared sic group group1 radius network-element ne1 downstream accounting
 failover-policy retry]
user@host# set timeout timeout
```

Configuring What Realms Are Local to the SIC Group (SRC CLI)

To configure what realms are local to the SIC group:

- From configuration mode, access the statement that configures local realms. For example, to configure the local realm called `realm1` for the group `group1`:

```
[edit]
user@host# edit shared sic group group1 local-realm realm1
```

Related Documentation

- [SIC Local Realms Overview on page 464](#)
- [Local and Shared Configurations for the SIC \(SRC CLI\) on page 445](#)
- [Creating an SIC Group and Server \(SRC CLI\) on page 476](#)

- [Creating an SIC Server Instance \(SRC CLI\) on page 477](#)

Configuration Statements for SIC Editing Rules (SRC CLI)

Use the following statements to configure the optional SIC editing rules at the **[edit]** hierarchy level.

Use the following statements to create the editing rule and specify the type of source used in the editing rule:

```
shared sic group identifier editing editing-rule {
    mode (replace | append);
}
shared sic group identifier editing editing-rule default {
    literal literal;
    request-attribute request-attribute;
    variable variable;
}
```

Use the following statements to configure the editing rule when you specify a literal as the source of the editing rule:

```
shared sic group identifier editing editing-rule source literal
shared sic group identifier editing editing-rule source literal identifier condition realm {
    (present | not-present);
}
shared sic group identifier editing editing-rule source literal identifier condition realm
does-not-equal value
shared sic group identifier editing editing-rule source literal identifier condition realm equals
value
shared sic group identifier editing editing-rule source literal identifier condition realm
has-prefix value
shared sic group identifier editing editing-rule source literal identifier condition realm
has-suffix value
shared sic group identifier editing editing-rule source literal identifier condition realm range
{
    low low;
    high high;
}
shared sic group identifier editing editing-rule source literal identifier condition request {
}
shared sic group identifier editing editing-rule source literal identifier condition request
attribute attribute-name {
    (present | not-present);
}
shared sic group identifier editing editing-rule source literal identifier condition request
attribute attribute-name does-not-equal value
shared sic group identifier editing editing-rule source literal identifier condition request
attribute attribute-name equals value
shared sic group identifier editing editing-rule source literal identifier condition request
attribute attribute-name has-prefix value
shared sic group identifier editing editing-rule source literal identifier condition request
attribute attribute-name has-suffix value
```

```

shared sic group identifier editing editing-rule source literal identifier condition request
  attribute attribute-name range {
    low low;
    high high;
  }
shared sic group identifier editing editing-rule source literal identifier condition user-identity
{
  (present | not-present);
}
shared sic group identifier editing editing-rule source literal identifier condition user-identity
  does-not-equal value
shared sic group identifier editing editing-rule source literal identifier condition user-identity
  equals value
shared sic group identifier editing editing-rule source literal identifier condition user-identity
  has-prefix value
shared sic group identifier editing editing-rule source literal identifier condition user-identity
  has-suffix value
shared sic group identifier editing editing-rule source literal identifier condition user-identity
  range {
    low low;
    high high;
  }

```

Use the following statements to configure the editing rule when you specify a request attribute as the source of the editing rule:

```

shared sic group identifier editing editing-rule source request-attribute identifier {
  remove-prefix remove-prefix;
  remove-suffix remove-suffix;
  remove-before remove-before;
  remove-after remove-after;
}
shared sic group identifier editing editing-rule source request-attribute identifier condition
  realm {
    (present | not-present);
  }
shared sic group identifier editing editing-rule source request-attribute identifier condition
  realm does-not-equal value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  realm equals value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  realm has-prefix value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  realm has-suffix value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  realm range {
    low low;
    high high;
  }
shared sic group identifier editing editing-rule source request-attribute identifier condition
  request {
  }
shared sic group identifier editing editing-rule source request-attribute identifier condition
  request attribute attribute-name {
    (present | not-present);
  }
}

```

```
shared sic group identifier editing editing-rule source request-attribute identifier condition
  request attribute attribute-name does-not-equal value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  request attribute attribute-name equals value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  request attribute attribute-name has-prefix value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  request attribute attribute-name has-suffix value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  request attribute attribute-name range {
    low low;
    high high;
  }
shared sic group identifier editing editing-rule source request-attribute identifier condition
  user-identity {
    (present | not-present);
  }
shared sic group identifier editing editing-rule source request-attribute identifier condition
  user-identity does-not-equal value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  user-identity equals value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  user-identity has-prefix value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  user-identity has-suffix value
shared sic group identifier editing editing-rule source request-attribute identifier condition
  user-identity range {
    low low;
    high high;
  }
}
```

Use the following statements to configure the editing rule when you specify an SIC variable as the source of the editing rule:

```
shared sic group identifier editing editing-rule source variable identifier
shared sic group identifier editing editing-rule source variable identifier condition realm {
  (present | not-present);
}
shared sic group identifier editing editing-rule source variable identifier condition realm
  does-not-equal value
shared sic group identifier editing editing-rule source variable identifier condition realm
  equals value
shared sic group identifier editing editing-rule source variable identifier condition realm
  has-prefix value
shared sic group identifier editing editing-rule source variable identifier condition realm
  has-suffix value
shared sic group identifier editing editing-rule source variable identifier condition realm
  range {
    low low;
    high high;
  }
}
shared sic group identifier editing editing-rule source variable identifier condition request {
}
shared sic group identifier editing editing-rule source variable identifier condition request
  attribute attribute-name {
    (present | not-present);
  }
```



```

}
shared sic group identifier editing editing-rule source variable identifier condition request
  attribute attribute-name does-not-equal value
shared sic group identifier editing editing-rule source variable identifier condition request
  attribute attribute-name equals value
shared sic group identifier editing editing-rule source variable identifier condition request
  attribute attribute-name has-prefix value
shared sic group identifier editing editing-rule source variable identifier condition request
  attribute attribute-name has-suffix value
shared sic group identifier editing editing-rule source variable identifier condition request
  attribute attribute-name range {
    low low;
    high high;
  }
}
shared sic group identifier editing editing-rule source variable identifier condition user-identity
{
  (present | not-present);
}
shared sic group identifier editing editing-rule source variable identifier condition user-identity
  does-not-equal value
shared sic group identifier editing editing-rule source variable identifier condition user-identity
  equals value
shared sic group identifier editing editing-rule source variable identifier condition user-identity
  has-prefix value
shared sic group identifier editing editing-rule source variable identifier condition user-identity
  has-suffix value
shared sic group identifier editing editing-rule source variable identifier condition user-identity
  range {
    low low;
    high high;
  }
}

```

Use the following statements to configure the target of the editing rule:

```

shared sic group identifier editing editing-rule target {
  request-attribute request-attribute;
  response-attribute response-attribute;
  variable variable;
}

```

Related Documentation

- [SIC Editing Rules \(SRC CLI\) on page 453](#)
- [Configuring the Optional Editing Rules Used by the SIC Group \(SRC CLI\) on page 505](#)
- [Configuring Explicit Routing \(SRC CLI\) on page 512](#)
- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 520](#)

Configuring the Optional Editing Rules Used by the SIC Group (SRC CLI)

When you use explicit routing for the SIC, you can optionally specify an editing rule you want applied to the accounting or authentication request before SIC sends the request to the target. To configure editing rules, you define a source, conditions, and a target.

Table 32 on page 506 lists the available sources, conditions, and targets you can define in editing rules, and “Configuration Statements for SIC Editing Rules (SRC CLI)” on page 502 provides a complete list of configuration statements used to define editing rules.

Table 32: SIC Editing Rule Options

Source	Conditions	Target
SIC literal	Match conditions:	Transactional variable
Transactional variable	<ul style="list-style-type: none"> • Realm • User identity 	RADIUS attribute in the request
RADIUS attribute in the request	<ul style="list-style-type: none"> • Request attribute 	RADIUS attribute in the response
	Condition tests: <ul style="list-style-type: none"> • Present • Not present • Equals • Does not equal • Has suffix • Has prefix • Within range 	

To configure an editing rule:

1. From configuration mode, access the statement that configures the editing rule, and specify a name for the editing rule. For example, to create an editing rule called `er1`:

```
[edit]
user@host# edit shared sic group identifier editing er1
```

2. Specify the editing mode.

```
[edit shared sic group identifier editing er1]
user@host# set mode (replace | append)
```

Where:

- **replace**—Current target (LValue) is replaced with the new value from the editing process
 - **append**—Current target (LValue) value is concatenated with the new target value from the editing process
3. Define the source of the editing rule. The source can be a literal, transactional variable, or an attribute in the request. For example, to define a literal called `literal1` as the source:

```
[edit]
edit shared sic group identifier editing er1 source literal
user@host# set literal1
```

4. (Optional) If the source is a request attribute, you can also specify whether to remove the prefix, remove the suffix, remove before @, or remove after @.

[edit]

```
user@host# edit shared sic group identifier editing er1 source request-attribute identifier
user@host# set remove-prefix remove-prefix
```

5. Define the editing rule conditions, which include specifying the match conditions and the condition tests. See [Table 32 on page 506](#) and “[Configuration Statements for SIC Editing Rules \(SRC CLI\)](#)” on page 502 for a complete list of configuration statements used to specify SIC editing rules. For example, to specify a condition that examines literals in accounting requests for the realm=abc.com:

[edit]

```
edit shared sic group identifier editing er1 literal literal1 condition realm equals
user@host# set abc.com
```

6. Define the target (where you want the result of the editing process to be placed) of the editing rule. The target can be a transactional variable, a RADIUS attribute in the request, or a RADIUS attribute in the response. For example, to place the results of the editing process in a variable called sic-variable1:

[edit]

```
user@host# edit shared sic group identifier editing er1 target
user@host# set variable sic-variable1
```

7. (Optional) Specify a default editing rule. You can set default editing rules for all three source types (literal, variable, and request attribute).

[edit]

```
user@host# edit shared sic group identifier editing editing-rule default
user@host# set literal literal
```

Related Documentation

- [SIC Editing Rules \(SRC CLI\) on page 453](#)
- [Configuring Explicit Routing \(SRC CLI\) on page 512](#)
- [Accounting Methods and Targets \(SRC CLI\) on page 446](#)
- [Configuration Statements for SIC Editing Rules \(SRC CLI\) on page 502](#)
- [Configuration Statements for SIC Explicit Accounting Routing Rules on page 510](#)
- [Request Routing \(SRC CLI\) on page 450](#)
- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 520](#)

Configuring the Accounting Method Used by the SIC Group (SRC CLI)

- [Configuring the SSR Database as the Accounting Method \(SRC CLI\) on page 508](#)
- [Configuring Proxy RADIUS as the Accounting Method \(SRC CLI\) on page 509](#)

Configuring the SSR Database as the Accounting Method (SRC CLI)

When you use the SSR database as the accounting method for the SIC group, accounting events are stored in the SSR database. Part of configuring this accounting method includes configuring the mapping between any request attributes, literals, or SIC variables, and the respective SAE plug-in attributes.



NOTE: At a minimum, you must configure the mapping between the SIC and the SAE plug-in attributes: `user-inet-address` and `vpin-id`. You also need to map these SAE plug-in attributes to the primary keys: `UserIPAddress` and `VpnID` in the SSR subscriber sessions table.

Use the following statements to configure the SSR database as the accounting method for the SIC group:

```
shared sic group identifier accounting-method accounting-method-name

shared sic group identifier accounting-method accounting-method-name database {
  remove-stopped-sessions
}

shared sic group identifier accounting-method accounting-method-name database
  plug-in-attribute id {
    request-attribute request-attribute;
    variable variable;
    literal literal;
  }
```

To configure the SSR database as the accounting method for the SIC group:

1. From configuration mode, access the statement that configures the SSR database as the accounting method. This sample procedure uses `group1` as the group identifier and `acm1` as the accounting method name.

```
[edit]
user@host# edit shared sic group group1 accounting-method acm1 database
```

2. (Optional) To conserve space in the SSR database, the SIC can delete corresponding entries in the SSR user session table and service session table when it receives an Accounting-Stop message from the NAS. Set this option to enable this capability.

```
[edit]
user@host# set remove-stopped-sessions
```

3. Specify the mapping between the SAE plug-in attribute and the request attribute, SIC variable, or literal. For example, to map the SAE plug-in attribute `login-name` to the request attribute `User-Name`:

```
[edit shared sic group group1 accounting-method acm1 database ]
user@host# edit plug-in-attribute login-name
user@host# set request-attribute User-Name
```

4. (Optional) Verify your configuration.

```
[edit shared sic group g1 accounting-method acm1 database]
user@host# show
plug-in-attribute {
  login-name {
    request-attribute User-Name;
  }
}
```

5. Configure the mapping between the SAE plug-in attribute and the field in the subscriber sessions table in the SSR database. For information about configuring this mapping, see [“Mapping SAE Plug-In Attributes to Fields in the Subscriber Sessions Table \(SRC CLI\)”](#) on page 409.

Configuring Proxy RADIUS as the Accounting Method (SRC CLI)

When you use the proxy RADIUS accounting method, the SIC forwards accounting messages to a remote AAA server (accounting target) located in a downstream network element for processing.

To use the proxy RADIUS accounting method, you need to have previously configured the downstream network element, associated the accounting target, and committed the configuration. For details about configuring downstream network elements and accounting targets, see [“Configuring Downstream Network Elements and Accounting and Authentication Targets \(SRC CLI\)”](#) on page 496.

Use the following statements to configure the proxy accounting method:

```
shared sic group identifier accounting-method accounting-method-name

shared sic group identifier accounting-method accounting-method-name proxy radius {
  network-element network-element;
}
```

To configure proxy RADIUS as the accounting method for the SIC group:

1. From configuration mode, access the statement that configures the accounting method. For example, to configure an accounting method called acm2 for the SIC group group2 and specify proxy RADIUS as the accounting method:

```
[edit]
user@host# edit shared sic group group2 accounting-method acm2 proxy radius
```

2. Specify the name of the previously configured downstream network element that contains the AAA server (accounting target) you want the SIC to forward accounting events to. For example, to forward accounting events to the downstream network element ne2:

```
[edit shared sic group group2 accounting-method acm2 proxy radius]
user@host# set network-element ne2
```

Configuring the Authentication Target Used by the SIC Server (SRC CLI)

The authentication target is a defined network element target that can then be assigned to an authentication route. It refers to a downstream RADIUS AAA server. You must specify a previously configured downstream RADIUS network element name.

Use the following statements to configure the authentication target for the SIC group:

```
shared sic group identifier server identifier authentication-route id target {  
  network-element network-element;
```

To configure the authentication route target used by the SIC server:

1. From configuration mode, access the statement that configures the authentication route target. For example, to configure an authentication route called aaa-route2 for the SIC group group2 and server svr2:

```
[edit]  
user@host# edit shared sic group group2 server svr2 authentication-route aaa-route2
```

2. Specify the name of the previously configured downstream network element you want to use as the authentication target. For example, to configure the preconfigured network element called ne2 as the authentication target:

```
[edit shared sic group group2 server svr2 authentication-route aaa-route2]  
user@host# set network-element ne2
```

Related Documentation

- [Accounting Methods and Targets \(SRC CLI\) on page 446](#)
- [Request Routing \(SRC CLI\) on page 450](#)

Configuring Request Routing (SRC CLI)

- [Configuration Statements for SIC Explicit Accounting Routing Rules on page 510](#)
- [Configuration Statements for SIC Explicit Authentication Routing Rules on page 511](#)
- [Configuring Explicit Routing \(SRC CLI\) on page 512](#)
- [Configuring Implicit Routing \(SRC CLI\) on page 515](#)

Configuration Statements for SIC Explicit Accounting Routing Rules

Use the following statements to configure explicit routing rules for the SIC at the **[edit]** hierarchy level:

```
shared sic group identifier server identifier accounting-route  
shared sic group identifier server identifier accounting-route id condition realm {  
  (present | not-present);  
}  
shared sic group identifier server identifier accounting-route id condition realm  
  does-not-equal value
```

```

shared sic group identifier server identifier accounting-route id condition realm equals value
shared sic group identifier server identifier accounting-route id condition realm has-prefix
    value
shared sic group identifier server identifier accounting-route id condition realm has-suffix
    value
shared sic group identifier server identifier accounting-route id condition realm range {
low low;
high high;
}
shared sic group identifier server identifier accounting-route id condition request
shared sic group identifier server identifier accounting-route id condition request attribute
    attribute-name {
    (present | not-present);
}
shared sic group identifier server identifier accounting-route id condition request attribute
    attribute-name does-not-equal value
shared sic group identifier server identifier accounting-route id condition request attribute
    attribute-name equals value
shared sic group identifier server identifier accounting-route id condition request attribute
    attribute-name has-prefix value
shared sic group identifier server identifier accounting-route id condition request attribute
    attribute-name has-suffix value
shared sic group identifier server identifier accounting-route id condition request attribute
    attribute-name range {
low low;
high high;
}
shared sic group identifier server identifier accounting-route id condition user-identity {
    (present | not-present);
}
shared sic group identifier server identifier accounting-route id condition user-identity
    does-not-equal value
shared sic group identifier server identifier accounting-route id condition user-identity equals
    value
shared sic group identifier server identifier accounting-route id condition user-identity
    has-prefix value
shared sic group identifier server identifier accounting-route id condition user-identity
    has-suffix value
shared sic group identifier server identifier accounting-route id condition user-identity range
    {
low low;
high high;
}

```

Configuration Statements for SIC Explicit Authentication Routing Rules

Use the following statements to configure explicit routing rules for the SIC at the **[edit]** hierarchy level:

```

shared sic group identifier server identifier authentication-route id
shared sic group identifier server identifier authentication-route id editing editing-rule
shared sic group identifier server identifier authentication-route id target {
}
shared sic group identifier server identifier authentication-route id condition user-identity
{
    (present | not-present);
}

```

```
}
shared sic group identifier server identifier authentication-route id condition user-identity
  range {
    low low;
    high high;
  }
shared sic group identifier server identifier authentication-route id condition user-identity
  equals value
shared sic group identifier server identifier authentication-route id condition user-identity
  does-not-equal value
shared sic group identifier server identifier authentication-route id condition user-identity
  has-prefix value
shared sic group identifier server identifier authentication-route id condition user-identity
  has-suffix value
shared sic group identifier server identifier authentication-route id condition realm {
  (present | not-present);
}
shared sic group identifier server identifier authentication-route id condition realm range {
  low low;
  high high;
}
shared sic group identifier server identifier authentication-route id condition realm equals
  value
shared sic group identifier server identifier authentication-route id condition realm
  does-not-equal value
shared sic group identifier server identifier authentication-route id condition realm has-prefix
  value
shared sic group identifier server identifier authentication-route id condition realm has-suffix
  value
shared sic group identifier server identifier authentication-route id condition request {
}
shared sic group identifier server identifier authentication-route id condition request attribute
  attribute-name {
    (present | not-present);
  }
shared sic group identifier server identifier authentication-route id condition request attribute
  attribute-name range {
    low low;
    high high;
  }
shared sic group identifier server identifier authentication-route id condition request attribute
  attribute-name equals value
shared sic group identifier server identifier authentication-route id condition request attribute
  attribute-name does-not-equal value
shared sic group identifier server identifier authentication-route id condition request attribute
  attribute-name has-prefix value
shared sic group identifier server identifier authentication-route id condition request attribute
  attribute-name has-suffix value
```

Configuring Explicit Routing (SRC CLI)

Explicit routing rules can be configured for accounting and authentication requests. When you configure an accounting or authentication route, you specify:

- (Optional) An editing rule you want to apply to the request before it is forwarded to the target.
- A predefined accounting method that is the target for the route.
- A predefined authentication route target (network element).
- (Optional) A set of conditions that must be matched in the request for the route to be selected.

When multiple routes are configured, they are evaluated in the order they are displayed by the **show** command. A newly created route is displayed last among the routes and has the lowest priority, so it is evaluated last. You can use the SRC CLI **insert** command to move a route before or after another route to change its evaluation order. The higher a route is displayed on the list, the sooner it is evaluated.

You can specify any combination of match conditions and condition tests as described in [Table 33 on page 513](#). For a complete list of statements used to configure explicit routing rules, see “[Configuration Statements for SIC Explicit Accounting Routing Rules](#)” on page 510 and “[Configuration Statements for SIC Explicit Authentication Routing Rules](#)” on page 511.

Table 33: Explicit Routing Rule Conditions

Match Condition	Condition Tests
<ul style="list-style-type: none"> • Realm • User identity • Request attribute 	<ul style="list-style-type: none"> • Present • Not present • Equals • Does not equal • Has suffix • Has prefix • Within range

For a complete list of statements you use to configure accounting routes, see “[Configuration Statements for SIC Explicit Accounting Routing Rules](#)” on page 510. For a complete list of statements you use to configure authentication routes, see “[Configuration Statements for SIC Explicit Authentication Routing Rules](#)” on page 511.

To configure explicit routes:

1. From configuration mode, access the configuration statement used to configure explicit routes. For example, to configure an accounting route called route66 for the server svr1, in a group called g1:

```
[edit]
user@host# edit shared sic group g1 server svr1 accounting-route route66
```

2. (Optional) Specify the name of the predefined editing rule you want applied to the request before it is forwarded to the target. For example to apply an editing rule called er1:

```
[edit shared sic group g1 server svr1 accounting-route route66]
user@host# edit editing er1
```

3. Specify a predefined accounting method or authentication routing target to use as the target of the route. If this route is selected, packets are routed to this target. For example, to specify an accounting method called `acctg-meth1` as the target:

```
[edit shared sic group g1 server svr1 accounting-route route66 editing er1]
user@host# up
[edit shared sic group g1 server svr1 accounting-route route66 editing]
user@host# up
[edit shared sic group g1 server svr1 accounting-route route66]
user@host# edit target
[edit shared sic group g1 server svr1 accounting-route route66 target]
user@host# set accounting-method acctg-meth1
```

4. (Optional) Specify the conditions that must be matched for the route to be selected. For example, to specify that the request must contain a `realm=abc.com`:

```
[edit shared sic group g1 server svr1 accounting-route route66 target]
user@host# up
[edit shared sic group g1 server svr1 accounting-route route66]
user@host# edit condition realm equals abc.com
[edit shared sic group g1 server svr1 accounting-route route66 condition realm equals
  abc.com]
user@host#
```

5. Commit the configuration.

```
user@host# commit
commit complete.
```

6. Verify the routing configuration.

```
[edit shared sic group g1 server svr1 accounting-route route66 condition realm equals
  abc.com]
user@host# up
[edit shared sic group g1 server svr1 accounting-route route66 condition realm]
user@host# up
[edit shared sic group g1 server svr1 accounting-route route66 condition]
user@host# up
[edit shared sic group g1 server svr1 accounting-route route66]
user@host# show
```

```
condition {
  realm {
    equals abc.com;
  }
}
editing {
  er1;
}
target {
  accounting-method acctg-meth1;
}
```

```
[edit shared sic group g1 server svr1 accounting-route route66]
user@host#
```

Configuring Implicit Routing (SRC CLI)

You configure implicit accounting and authentication routes by specifying the name of a previously configured network element that has the proxy function assigned to it. You can also define a default route used for all requests from all realms, or you can specify that only requests from specific realms are routed to the proxy AAA server. When you specify specific realms, you have the option to set a condition of either an exact match of the realm string, or a match on the prefix of the realm string.

Use the following statements to configure implicit routes for the SIC:

```
shared sic group identifier radius network-element id proxy {
}
shared sic group identifier radius network-element id proxy realm realmValue {
condition (exact | prefix);
}
```

To configure implicit routes for the SIC:

1. From configuration mode, access the statement that configures the remote AAA server as a proxy. For example, to configure the AAA server in a network element called `ne1` as a proxy:

```
[edit]
user@host# edit shared sic group group1 radius network-element ne1 proxy
```

2. (Optional) Specify that only requests from specific realms are routed to the proxy AAA server by specifying the names of the realms. For example, to specify that all requests from the realm called `abc.com` are routed to the proxy AAA server:

```
[edit shared sic group group1 radius network-element ne1 proxy]
user@host# edit realms abc.com
```

3. (Optional) Specify the match condition for the realm.

```
[edit shared sic group group1 radius network-element ne1 proxy realm abc.com]
user@host# set condition exact | prefix
```

4. (Optional) Specify whether you want this proxy AAA server to be the default route for requests from all realms.

```
[edit shared sic group group1 radius network-element ne1 proxy]
user@host# set default-route-for-all-realms
```

Configuring Event Logging for an SIC Server (SRC CLI)

You can configure the SIC server to capture any number of log streams called loggers. If you configure multiple log streams, make sure you configure unique names for each log stream. You can configure the log stream to display only log messages from particular log groups. To configure the event level for a log group, you first specify the log group and then specify the event level for it.

Use the following statements to configure event logging for the SIC server:

```
shared sic group identifier server identifier
```

```
shared sic group identifier server identifier logger id
```

```
shared sic group identifier server identifier logger id file {  
  filter (/error | /debug-error);  
  filename filename;  
  maximum-file-size maximum-file-size;  
  rollover-interval rollover-interval;  
  rollover-on-startup;  
  flush-after-writes;  
  high-resolution-timestamps;  
  header header;  
  footer footer;  
  prepend-message-header;  
  work-id-label work-id-label;  
  work-id-padding work-id-padding;  
  utc;  
}  
shared sic group identifier server identifier logger id group (administration | audit |  
  configuration | system | packet | packet-trace | packet-trace-raw) {  
  events (error | warning | standard | detail | debug);  
}
```

To configure event logging for the SIC server:

1. From configuration mode, access the statement that configures the server belonging to the SIC group. For example, to configure the server called `sicsr1` for the group `group1`:

```
[edit]  
user@host# edit shared sic group group1 server sicsr1
```

2. Specify the name used by the server to identify the log stream.

```
[edit shared sic group group1 server sicsr1 logger]  
user@host# set id log1
```

3. (Optional) Specify the filter to define which event messages are logged or ignored.

```
[edit shared sic group group1 server sicsr1 logger log1 file]  
user@host# set filter (/error | /debug-error)
```

where:

- `/error`—Error events are captured for every log group
- `/debug-error`—Debug events are captured for every log group

4. Specify the prefix to be added to the log file for easy identification.

```
[edit shared sic group group1 server sicsr1 logger log1 file]  
user@host# set filename filename
```

5. (Optional) Specify the maximum size of the log file and the rollover file.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set maximum-file-size maximum-file-size
```



NOTE: The maximum file size is specified in KB. Maximum size of the log file is 10,000,000 KB.

Do not set the maximum file size to a value greater than the available disk space.

6. (Optional) Specify the time in seconds for the rollover interval after which the new log file is created.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set rollover-interval rollover-interval
```

7. (Optional) Specify whether the new log file is to be created every time the server starts.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set rollover-on-startup
```

8. (Optional) Specify whether or not to buffer log messages.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set flush-after-writes
```

- If set, log messages are immediately written to the log file without buffering. Use this setting for real-time logging.
- If not set, SIC log messages are kept in the buffer until the buffer is full and then all messages in the buffer are written to the log file. Use this setting for performance optimization, when real-time logging is not needed.

9. (Optional) Specify whether the high-resolution-time reporting system functions are used.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set high-resolution-timestamps
```

10. (Optional) Specify the header message to be added to the beginning of each log file.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set header header
```

11. (Optional) Specify the footer message to be added to the end of each log file.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set footer footer
```

12. (Optional) Specify whether to prepend each log message with additional information such as time, thread, and transaction information.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set prepend-message-header
```

13. (Optional) Specify the work data ID prefix to be added to each log message.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set work-id-label work-id-label
```

14. (Optional) Specify the string to be added to each log message if work data is not available.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set work-id-padding work-id-padding
```

15. (Optional) Specify the time and date values to Universal Time Coordinates (UTC, formerly known as Greenwich Mean Time, or GMT). If disabled, time and date reflect local time.

```
[edit shared sic group group1 server sicser1 logger log1 file]
user@host# set utc
```

16. Configure the event level for each log group for which you want to collect events. First, specify the name of the log group, and then specify the event level. Repeat the process for each log group for which you want to collect events.

```
[edit]
user@host# edit shared sic group group1 server sicser1 logger log1 group (administration
| configuration | system | packet | packet-trace | packet-trace-raw)
```

Where:

- **administration**—Log group reports events related to server administration.
- **audit**—Log group reports SSR database operations (insert, update, and delete operations) for both the user session and the corresponding operations in the service session table.
- **configuration**—Log group reports events related to server configuration.
- **system**—Log group reports events related to the system, such as system start and system stop.
- **packet**—Log group reports events related to transaction processing, such as incoming and outgoing packets.
- **packet-trace**—Log group displays contents of a packet. The format is attribute name:attribute value.
- **packet-trace-raw**—Log group displays raw data (octets) of incoming and outgoing packets.



NOTE: To write SSR database operation information to a new file, you must configure a new logger and specify only **audit** in the group.

17. (Optional) Specify the highest event level for the log group.

```
[edit shared sic group group1 server sicser1 logger log1 group]
user@host# set events (error | warning | standard | detail | debug)
```

Where:

- **error**—Messages in log shown at event level error.
- **warning**—Messages in log shown at event levels error and warning.
- **standard**—Messages in log shown at event levels error, warning, and standard.
- **detail**—Messages in log shown at event levels error, warning, standard, and detail.
- **debug**—Messages in log shown at event levels error, warning, standard, detail, and debug.

Related Documentation

- [SIC Event Logging Overview \(SRC CLI\) on page 464](#)
- [Configuring SNMP for the SIC Group \(SRC CLI\) on page 519](#)
- [Local and Shared Configurations for the SIC \(SRC CLI\) on page 445](#)
- [Example: Basic SIC Group Configuration \(SRC CLI\) on page 520](#)

Configuring SNMP for the SIC Group (SRC CLI)

You can configure each SNMP event and associated dilution factor. When an event occurs, an SNMP trap is sent to the SNMP manager.

Use the following statements to configure SNMP for the SIC server:

```
shared sic group identifier snmp event (sic-server-startup | sic-server-shutdown |
  sic-server-unauthorized-administration-request | sic-server-internal-error |
  sic-server-resource-failure | sic-server-log-file-failure) {
  dilution-factor dilution-factor;
}
```

To configure SNMP events for the SIC group:

1. Specify the SNMP trap name for which you want to configure the dilution factor.

```
[edit]
user@host# edit shared sic group group1 snmp event (sic-server-startup |
  sic-server-shutdown | sic-server-unauthorized-administration-request |
  sic-server-internal-error | sic-server-resource-failure | sic-server-log-file-failure)
```

Where:

- **sic-server-startup**—SNMP trap on server startup.
- **sic-server-shutdown**—SNMP trap on server shutdown.
- **sic-server-unauthorized-administration-request**—SNMP trap on unauthorized administration request.

- **sic-server-internal-error**—SNMP trap on server internal error.
 - **sic-server-resource-failure**—SNMP trap on server resource failure.
 - **sic-server-log-file-failure**—SNMP trap on server log file failure.
2. (Optional) Specify the dilution factor. The event is sent to the SNMP manager every *n* occurrences of the condition that generated the alert.

```
[edit shared sic group group1 snmp event]
user@host# set dilution-factor dilution-factor
```

- Related Documentation**
- [SNMP Support for the SIC Overview \(SRC CLI\) on page 467](#)
 - [Configuring Event Logging for an SIC Server \(SRC CLI\) on page 515](#)

Example: Basic SIC Group Configuration (SRC CLI)

This sample configuration uses the default SIC group called default-group, and the default SIC server called default-server.

An editing rule called username specifies that if the source, which is the request attribute User-Name, contains the @test.com suffix, the suffix is to be removed, and the resulting value placed in the target, which is the request attribute User-Name. A second editing rule, called vpnid, specifies that the target, which is the SIC variable vpn-id, should be replaced with the value of the source, which is the request attribute NAS-Identifier.

The SIC group (default-group) includes the default device model called default-model, which are both using the default dictionary called radius.

The accounting listener for the SIC listens on port 1813 for incoming accounting events. An upstream network element called netpc is using the default device model called default-model. The netpc network element contains four accounting clients called netpc13, netpc14, netpc15, and netpc16. The IP addresses and shared secrets of these accounting clients are provided as examples only. The outbound transport uses port 0.

The accounting route called test-route specifies that the editing rule called vpnid is to be applied before the request is routed to the accounting target, which by default is the SSR database (default-method).

[Table 34 on page 520](#) lists the attribute mapping defined between the SIC and the SAE plug-in attributes.

Table 34: Sample Configuration Attribute Associations

SIC Variable or Attribute	SAE Plug-In Attribute
Request-attribute User-Name	Login-name
Request-attribute Calling-Station-Id	Property.calling-station-id
Variable ReceiveTime	Property.session-start-time

Table 34: Sample Configuration Attribute Associations (*continued*)

SIC Variable or Attribute	SAE Plug-In Attribute
Variable UserStatusType	Property.session-state
Request-attribute Framed-IP-Address	User-inet-address

Three log streams are configured, including the default log stream called default-logger, which captures events for the log groups at the event levels listed in [Table 35 on page 521](#).

Table 35: Log Groups and Associated Event Level for Log Stream=default logger

Log Group	Event Level
Administration	Warning
Configuration	Warning
Packet	Debug
PacketTrace	Warning
PacketTraceRaw	Warning
System	Warning

Two additional log streams are configured, called debug-logger and error-logger, which capture events for the log groups at the event levels listed in [Table 36 on page 521](#) and [Table 37 on page 522](#).

Table 36: Log Groups and Associated Event Level for Log Stream=debug-logger

Log Group	Event Level
Administration	Debug
Configuration	Debug
Packet	Debug
PacketTrace	Debug
PacketTraceRaw	Debug
System	Debug

Table 37: Log Groups and Associated Event Level for Log Stream=error-logger

Log Group	Event Level
Administration	Warning
Configuration	Warning
Packet	Warning
PacketTrace	Warning
PacketTraceRaw	Warning
System	Warning

```
user@host# show slot 0 sic
```

```
initial {
  directory-connection {
    credentials *****;
    entry-dn l=SIC,ou=staticConfiguration,ou=Configuration,o=Management,o=umc;
    filter (objectClass=*);
    port 389;
    principal cn=umcadmin,o=umc;
    url 127.0.0.1;
  }
}
server {
  name default-server;
}
```

```
user@host# show shared sic group default-group accounting-method
```

```
default-method database {
  plug-in-attribute {
    login-name {
      request-attribute User-Name;
    }
    property.calling-station-id {
      request-attribute Calling-Station-Id;
    }
    property.session-start-time {
      variable ReceiveTime;
    }
    property.session-state {
      variable UserStatusType;
    }
    user-inet-address {
      request-attribute Framed-IP-Address;
    }
    vpn-id;
  }
}
```

```
[edit]
```

user@host# show shared sic group default-group editing

```
username {
  mode replace;
  source {
    request-attribute {
      User-Name {
        remove-suffix @test.com;
      }
    }
  }
  target {
    request-attribute User-Name;
  }
}
vpnid {
  mode replace;
  source {
    request-attribute {
      NAS-Identifier;
    }
  }
  target {
    variable vpn-id;
  }
}
```

[edit]

user@host# show shared sic group default-group radius

```
accounting-listener {
  transport {
    1813 {
      connect-timeout 1000;
      connections-per-thread 15;
      disconnect-timeout 1000;
      port 1813;
    }
  }
}
network-element netpc {
  upstream {
    model default-model;
    accounting-client {
      netpc13 {
        accounting-secret secret;
        address 10.227.6.213;
      }
      netpc14 {
        accounting-secret secret;
        address 10.227.6.214;
      }
      netpc15 {
        accounting-secret secret;
        address 10.227.6.215;
      }
      netpc16 {
```

```
        accounting-secret secret;
        address 10.227.6.216;
    }
}
}
outbound-transport {
    default-outbound-transport {
        connect-timeout 1000;
        connections-per-thread 15;
        disconnect-timeout 1000;
        port 0;
    }
}

[edit]
user@host# show shared sic group default-group dictionary radius
attribute ARAP-Challenge-Response {
    radius {
        format octets;
        type 84;
    }
}
attribute ARAP-Features {
    radius {
        format octets;
        type 71;
    }
}
attribute ARAP-Password {
    radius {
        format octets;
        type 70;
    }
}
attribute Proxy-State {
    radius {
        format string;
        type 33;
    }
}
attribute Reply-Message {
    radius {
        format string;
        type 18;
    }
}
attribute Service-Type {
    radius {
        constant Administrative {
            6;
        }
        constant Authenticate-Only {
            8;
        }
        constant Authorize-Only {
            17;
        }
        constant Call-Check {
            10;
        }
    }
}
```

```
    }
    constant Callback-Administrative {
        11;
    }
    constant Callback-Framed {
        4;
    }
    constant Callback-Login {
        3;
    }
    constant Callback-NAS-Prompt {
        9;
    }
    constant Fax {
        13;
    }
    constant Framed {
        2;
    }
    constant IAPP-AP-Check {
        16;
    }
    constant IAPP-Register {
        15;
    }
    constant Login {
        1;
    }
    constant Modem-Relay {
        14;
    }
    constant NAS-Prompt {
        7;
    }
    constant Outbound {
        5;
    }
    constant Voice {
        12;
    }
    format integer;
    type 6;
}
}
attribute Session-Timeout {
    radius {
        format integer;
        type 27;
    }
}
attribute State {
    radius {
        format string;
        type 24;
    }
}
}
attribute TeliaSonera-Chargeable-User-Id {
    radius {
        format string;
        type 192;
        vendor-id 15297;
    }
}
```

```
    }  
  }  
  attribute TeliaSonera-Location-Info {  
    radius {  
      format string;  
      type 194;  
      vendor-id 15297;  
    }  
  }  
  attribute TeliaSonera-Location-Name {  
    radius {  
      format string;  
      type 195;  
      vendor-id 15297;  
    }  
  }  
  attribute TeliaSonera-Operator-Name {  
    radius {  
      format string;  
      type 193;  
      vendor-id 15297;  
    }  
  }  
  attribute TeliaSonera-Visited-Operator-ID {  
    radius {  
      format string;  
      type 196;  
      vendor-id 15297;  
    }  
  }  
  attribute Termination-Action {  
    radius {  
      constant Default {  
        0;  
      }  
      constant RADIUS-Request {  
        1;  
      }  
      format integer;  
      type 29;  
    }  
  }  
  attribute Tunnel-Assignment-ID {  
    radius {  
      format string;  
      tagged;  
      type 82;  
    }  
  }  
  attribute Tunnel-Client-Auth-ID {  
    radius {  
      format string;  
      tagged;  
      type 90;  
    }  
  }  
  attribute Tunnel-Client-Endpoint {  
    radius {  
      format string;  
      tagged;  
      type 66;  
    }  
  }  
}
```

```
    }  
  }  
  attribute Tunnel-Medium-Type {  
    radius {  
      constant 802 {  
        6;  
      }  
      constant ATM {  
        3;  
      }  
      constant Appletalk {  
        12;  
      }  
      constant BBN-1822 {  
        5;  
      }  
      constant Banyan-Vines {  
        14;  
      }  
      constant Decnet-IV {  
        13;  
      }  
      constant E.163 {  
        7;  
      }  
      constant E.164 {  
        8;  
      }  
      constant E.164-NSAP-subaddress {  
        15;  
      }  
      constant F.69 {  
        9;  
      }  
      constant Frame-Relay {  
        4;  
      }  
      constant IP {  
        1;  
      }  
      constant IPX {  
        11;  
      }  
      constant X.121 {  
        10;  
      }  
      constant X.25 {  
        2;  
      }  
      format integer;  
      tagged;  
      type 65;  
    }  
  }  
  attribute Tunnel-Password {  
    radius {  
      format string;  
      salt-encrypt;  
      tagged;  
      type 69;  
    }  
  }
```

```
}
attribute Tunnel-Preference {
  radius {
    format integer;
    tagged;
    type 83;
  }
}
attribute Tunnel-Private-Group-ID {
  radius {
    format string;
    tagged;
    type 81;
  }
}
attribute Tunnel-Server-Auth-ID {
  radius {
    format string;
    tagged;
    type 91;
  }
}
attribute Tunnel-Server-Endpoint {
  radius {
    format string;
    tagged;
    type 67;
  }
}
attribute Tunnel-Type {
  radius {
    constant AH {
      6;
    }
    constant ATMP {
      4;
    }
    constant DVS {
      11;
    }
    constant ESP {
      9;
    }
    constant GRE {
      10;
    }
    constant IP-IP {
      7;
    }
    constant IP-IP-Tunneling {
      12;
    }
    constant L2F {
      2;
    }
    constant L2TP {
      3;
    }
    constant MIN-IP-IP {
      8;
    }
  }
}
```



```

        constant PTP {
            1;
        }
        constant VLAN {
            13;
        }
        constant VTP {
            5;
        }
        format integer;
        tagged;
        type 64;
    }
}
attribute User-Name {
    radius {
        format string;
        type 1;
    }
}
attribute User-Password {
    radius {
        format string;
        type 2;
    }
}
}
user@host# show default-model
dictionary radius;

*****

user@host# show shared sic group default-group server
default-server {
    accounting-route {
        test-route {
            editing {
                vpnid;
            }
            target {
                accounting-method default-method;
            }
        }
        default-route {
            target {
                accounting-method default-method;
            }
        }
    }
    logger {
        debug-logger {
            file {
                filename sic_debug;
                filter /debug-error;
                flush-after-writes;
                maximum-file-size 0;
                prepend-message-header;
                rollover-interval 86400;
            }
        }
        group {
            administration events debug;
            configuration events debug;
        }
    }
}

```

```
        packet events debug;
        packet-trace events debug;
        packet-trace-raw events debug;
        system events debug;
    }
}
default-logger {
    file {
        filename sic;
        filter customized;
        flush-after-writes;
        maximum-file-size 0;
        prepend-message-header;
        rollover-interval 86400;
    }
    group {
        administration events warning;
        configuration events warning;
        packet events debug;
        packet-trace events warning;
        packet-trace-raw events warning;
        system events warning;
    }
}
error-logger {
    file {
        filename sic_error;
        filter /error;
        flush-after-writes;
        maximum-file-size 0;
        prepend-message-header;
        rollover-interval 86400;
    }
    group {
        administration events warning;
        configuration events warning;
        packet events warning;
        packet-trace events warning;
        packet-trace-raw events warning;
        system events warning;
    }
}
}
}

[edit]
```

Related Documentation

- [Local and Shared Configurations for the SIC \(SRC CLI\) on page 445](#)
- [Accounting Methods and Targets \(SRC CLI\) on page 446](#)
- [SIC Editing Rules \(SRC CLI\) on page 453](#)
- [RADIUS and Diameter Configuration for the SIC Overview \(SRC CLI\) on page 456](#)

Configuring the NAS Groups (SRC CLI)

Tasks to configure the NAS groups are:

- [Configuring NAS Groups on page 531](#)
- [Configuring the NAS Group Device Capabilities \(SRC CLI\) on page 532](#)
- [Classifying Interfaces on page 532](#)
- [Configuring NAS Group Routes on page 533](#)

Configuring NAS Groups

Use the following configuration statements to configure the NAS groups:

```
shared network nas-group name {
  hosted-by [hosted-by...];
  peers [peers...];
  scope [scope...];
  default-peer default-peer;
  update-grace-period update-grace-period;
  initial-ppr-delay initial-ppr-delay;
}
```

To configure the group of peers:

1. From configuration mode, access the configuration statements for the NAS group.

```
user@host# edit shared network nas-group name
```

2. Specify the hosts that instantiate this peer group. If the peer group is an AAA peer group, the SAEs on the listed hosts create device drivers for this peer group.

```
[edit shared network nas-group name]
user@host# set hosted-by [hosted-by...]
```

3. (Optional) Specify the peers in this NAS group.

```
[edit shared network nas-group name]
user@host# set peers [peers...]
```

4. (Optional) Specify the service scopes available to subscribers connected to this NAS group.

```
[edit shared network nas-group name]
user@host# set scope [scope...]
```

5. (Optional) Specify the default peer.

```
[edit shared network nas-group name]
user@host# set default-peer default-peer
```

6. (Optional) Specify the grace period for interim updates.

```
[edit shared network nas-group name]
```

```
user@host# set update-grace-period update-grace-period
```

7. (Optional) Specify the delay for sending initial Push-Profile-Requests (PPRs) to install policies.

```
[edit shared network nas-group name]  
user@host# set initial-ppr-delay initial-ppr-delay
```

Configuring the NAS Group Device Capabilities (SRC CLI)

The SAE uses user interim accounting requests to keep the user session alive. Some NAS devices do not send user interim accounting requests, which causes the user session to time out in the SAE. To support this type of NAS device, the SAE can use service interim accounting requests to keep the user session alive.

Use the following configuration statements to configure the NAS group device capabilities:

```
shared network nas-group device-capabilities {  
  no-user-interim-update ;  
}
```

To configure the NAS group device capabilities:

1. From configuration mode, access the configuration statements for the NAS group device capabilities.

```
user@host# edit shared network nas-group device-capabilities
```

2. Specify whether to use service interim accounting requests.

```
[edit shared network nas-group device-capabilities]  
user@host# set no-user-interim-update
```

- If this option is set, the SAE uses service interim accounting requests to keep the user session alive in the SAE. The SAE can also send user-tracking events to plug-ins driven by SRC interim update interval.
- If this option is not set, the SAE sends user interim tracking events only when it receives a user interim update from the NAS device.

Classifying Interfaces

Use the following configuration statements to define interface classification scripts:

```
shared network nas-group name interface-classifier rule name {  
  target target;  
}  
  
shared network nas-group name interface-classifier rule name condition name ...  
  
shared network nas-group name interface-classifier rule name script {  
  script-value;  
  include include;  
}
```

A classification script can contain either a target and a condition or a script. If you do not define a script, the classifier must have both a target and a condition.

To define interface classification scripts:

1. From configuration mode, enter the interface classifier configuration for a NAS group.

```
user@host# edit shared network nas-group name interface-classifier
```

2. Create a rule for the classifier. You can create multiple rules for the classifier.

```
[edit shared network nas-group name interface-classifier]
user@host# edit rule name
```

3. Configure either a target or a script for the rule.

- Configure the target for the rule.

```
[edit shared network nas-group name interface-classifier rule name]
user@host# set target target
```

If you configure a target for the rule, you must configure a match condition. You can create multiple conditions for the rule. See *Interface Classification Conditions*.

```
[edit shared network nas-group name interface-classifier rule name]
user@host# set condition name
```

- Configure the script for the rule.

```
[edit shared network nas-group name interface-classifier rule name]
user@host# edit script
```

(Optional) You can specify a script target.

```
[edit shared network nas-group name interface-classifier rule name script]
user@host# set script-value
```

(Optional) You can include a script that has already been created.

```
[edit shared network nas-group name interface-classifier rule name script]
user@host# set include include
```

Where *include* is a reference to an existing script that is included in the script you are configuring.

Configuring NAS Group Routes

Use the following configuration statements to configure the route for messages:

```
shared network nas-group name routes name term name {
  precedence precedence;
}

shared network nas-group name routes name {
  transaction-variable (request-packet | user-name | realm);
```

```

dictionary-attribute (user-name | user-password | chap-password | nas-ip-address |
nas-port | service-type | framed-protocol | framed-ip-address | framed-ip-netmask |
framed-mtu | framed-compression | login-ip-host | callback-number | state |
vendor-specific | called-station-id | calling-station-id | nas-identifier | login-lat-service
| login-lat-node | login-lat-group | chap-challenge | nas-port-type | port-limit |
login-lat-port);
operator (equals | not_equal | present | not_present | prefix | suffix | range);
value value;
low low;
high high;
}

```

To configure route selection for messages from the SRC Diameter server:

1. From configuration mode, access the configuration statements for route selection.

```
user@host# edit shared network nas-group name routes name
```

2. (Optional) Specify the order by which the route is selected. The route that meets all the matching criteria and has the lowest precedence is selected first. Routes without the precedence defined are considered after those that have the precedence defined. The route with precedence of -1 is the default route. The default route is considered after all the other routes, and only one default route can be defined.

```
[edit shared network nas-group name routes name]
user@host# set precedence precedence
```

3. From configuration mode, access the configuration statements for route selection criteria.

```
user@host# edit shared network nas-group name routes name term name
```

All the criteria must match for this route to be selected.

4. Specify the name of the transaction variable used as the matching criterion.

```
[edit shared network nas-group name routes name term name]
user@host# set transaction-variable (request-packet | user-name | realm)
```

5. (Optional) Specify the name of the dictionary attribute contained in the attribute store. This is applicable only if the transaction variable is request-packet.

```
[edit shared network nas-group name routes name term name]
user@host# set dictionary-attribute (user-name | user-password | chap-password |
nas-ip-address | nas-port | service-type | framed-protocol | framed-ip-address |
framed-ip-netmask | framed-mtu | framed-compression | login-ip-host |
callback-number | state | vendor-specific | called-station-id | calling-station-id |
nas-identifier | login-lat-service | login-lat-node | login-lat-group | chap-challenge
| nas-port-type | port-limit | login-lat-port)
```

6. Specify the operator for criterion matching.

```
[edit shared network nas-group name routes name term name]
user@host# set operator (equals | not_equal | present | not_present | prefix | suffix |
range)
```

7. (Optional) Specify the value to be matched by the target.

```
[edit shared network nas-group name routes name term name]
user@host# set value value
```

8. (Optional) Specify the low end of the range criterion.

```
[edit shared network nas-group name routes name term name]
user@host# set low low
```

9. (Optional) Specify the high end of the range criterion.

```
[edit shared network nas-group name routes name term name]
user@host# set high high
```

Configuring the SAE to Manage AAA Devices

Use the following configuration statements to configure the AAA device driver:

```
shared sae configuration driver aaa {
  sae-community-manager sae-community-manager;
  origin-host origin-host;
  origin-realm origin-realm;
  keep-alive-timeout keep-alive-timeout;
  registry-retry-interval registry-retry-interval;
  reply-timeout reply-timeout;
  sequential-message-timeout sequential-message-timeout;
  transient-session-timeout transient-session-timeout;
  max-update-interval max-update-interval;
  update-grace-period update-grace-period;
  resume-unrecovered;
  thread-pool-size thread-pool-size;
  thread-idle-timeout thread-idle-timeout;
}
```

To configure the AAA device driver:

1. From configuration mode, access the configuration statements for the AAA device driver.

```
user@host# edit shared sae configuration driver aaa
```

2. Specify the name of the community manager.

```
[edit shared sae configuration driver aaa]
user@host# set sae-community-manager sae-community-manager
```

3. (Optional) Specify the fully qualified domain name used to identify this host.

```
[edit shared sae configuration driver aaa]
user@host# set origin-host origin-host
```

4. (Optional) Specify the DNS name of the machine used to identify this host.

```
[edit shared sae configuration driver aaa]
user@host# set origin-realm origin-realm
```

5. (Optional) Specify the keepalive timeout before the registry to a Diameter server expires.

```
[edit shared sae configuration driver aaa]
user@host# set keep-alive-timeout keep-alive-timeout
```

6. (Optional) Specify the interval between retrying a failed registry to a Diameter server.

```
[edit shared sae configuration driver aaa]
user@host# set registry-retry-interval registry-retry-interval
```

7. (Optional) Specify the timeout before a request sent to a Diameter server expires.

```
[edit shared sae configuration driver aaa]
user@host# set reply-timeout reply-timeout
```

8. (Optional) Specify the timeout before an expected message expires.

```
[edit shared sae configuration driver aaa]
user@host# set sequential-message-timeout sequential-message-timeout
```

9. (Optional) Specify the timeout before a temporary session expires.

```
[edit shared sae configuration driver aaa]
user@host# set transient-session-timeout transient-session-timeout
```

10. (Optional) Specify the maximum interval between interim updates for a subscriber session.

```
[edit shared sae configuration driver aaa]
user@host# set max-update-interval max-update-interval
```

11. (Optional) Specify the grace period in which to expect an interim update for a subscriber session.

```
[edit shared sae configuration driver aaa]
user@host# set update-grace-period update-grace-period
```

12. (Optional) Specify whether to resume a subscriber session that has failed to recover from a failover.

```
[edit shared sae configuration driver aaa]
user@host# set resume-unrecovered
```

13. (Optional) Specify the number of working threads that process requests.

```
[edit shared sae configuration driver aaa]
user@host# set thread-pool-size thread-pool-size
```

14. (Optional) Specify the timeout for stopping working threads after they become idle.

```
[edit shared sae configuration driver aaa]
user@host# set thread-idle-timeout thread-idle-timeout
```

15. (Optional) Configure the session store parameters for the AAA device driver.

From configuration mode, access the configuration statement that configures the session store for the AAA device driver.

```
user@host# edit shared sae configuration driver aaa session-store
```

For more information about configuring session store parameters, see [“Configuring the Session Store Feature \(SRC CLI\)” on page 31](#).

Configuring AAA Policies (SRC CLI)

Tasks to configure AAA policies are:

- [Configuring AAA Policy Lists on page 537](#)
- [Configuring AAA Policy Rules on page 537](#)
- [Configuring Template Activation Actions on page 537](#)

Configuring AAA Policy Lists

To configure AAA policy lists:

1. From configuration mode, create a policy list. For example, to create a policy list called l1 within a policy group called tiered_aaa:

```
user@host# edit policies group tiered_aaa list l1
```

2. Specify the type of policy list.

```
[edit policies group tiered_aaa list l1]
user@host# set role aaa
```

3. Specify where the policy is applied on the device.

```
[edit policies group tiered_aaa list l1]
user@host# set applicability both
```

Configuring AAA Policy Rules

To configure AAA policy rules:

1. From configuration mode, create a policy rule inside a policy list that has already been created and configured. For example, to create a policy rule called r1 within policy list l1:

```
user@host# edit policies group tiered_aaa list l1 rule r1
```

2. Specify the type of policy rule.

```
[edit policies group tiered_aaa list l1 rule r1]
user@host# set type aaa
```

Configuring Template Activation Actions

Use this action to activate service templates for RADIUS-enabled devices. You can configure template activation actions for AAA policy rules.

The template name and parameters are listed in the SIC service templates.

Use the following configuration statements to configure a template activation action:

```
policies group name list name rule name template-activation name {
  template-name template-name;
  description description;
```

```
}  
policies group name list name rule name template-activation name variables name {  
    value value;  
    type type;  
}
```

To configure a template activation action:

1. From configuration mode, enter the template activation action configuration. For example, in this procedure, ta is the name of the template activation action.

```
user@host# edit policies group tiered_aaa list l1 rule r1 template-activation ta
```

2. Enter the template name to activate.

```
[edit policies group tiered_aaa list l1 rule r1 template-activation ta]  
user@host# set template-name template-name
```

3. (Optional) Enter a description for the template activation action.

```
[edit policies group tiered_aaa list l1 rule r1 template-activation ta]  
user@host# set description description
```

4. From configuration mode, enter the parameters used by the template.

```
user@host# edit policies group tiered_aaa list l1 rule r1 template-activation ta variables  
    name
```

For example:

```
user@host# edit policies group tiered_aaa list l1 rule r1 template-activation ta variables  
    upstreamBandwidth
```

5. (Optional) Configure the value for the variable.

```
[edit policies group tiered_aaa list l1 rule r1 template-activation ta variables name]  
user@host# set value value
```

For example:

```
[edit policies group tiered_aaa list l1 rule r1 template-activation ta variables  
    upstreamBandwidth]  
user@host# set value rateParameter
```

6. (Optional) Configure the variable type. Variable types are mapped to parameter types.

```
[edit policies group tiered_aaa list l1 rule r1 template-activation ta variables name]  
user@host# set type type
```

For example:

```
[edit policies group tiered_aaa list l1 rule r1 template-activation ta variables  
    upstreamBandwidth]  
user@host# set type rate
```

Related Documentation

- [Configuring Template Activation Actions \(SRC CLI\)](#)
- [Before You Configure SRC Policies](#)
- [Managing Dynamic Services on page 434](#)

CHAPTER 34

Device and Service Templates for Dynamic Authorization (SRC CLI)

- [Device and Service Template Configuration Overview \(SRC CLI\) on page 539](#)
- [Configuring Device Templates \(SRC CLI\) on page 545](#)
- [Configuring the Device Capabilities Supported in the Device Template \(SRC CLI\) on page 546](#)
- [Configuration Statements for SIC Service Templates \(SRC CLI\) on page 548](#)
- [Configuring SIC Service Templates \(SRC CLI\) on page 549](#)
- [Configuration Statements for Tagged Attributes in SIC Service Templates \(SRC CLI\) on page 557](#)
- [Configuring Tagged Attributes in SIC Service Templates \(SRC CLI\) on page 558](#)
- [Configuration Statements for SIC Global Service Templates \(SRC CLI\) on page 564](#)
- [Configuring Global Service Templates \(SRC CLI\) on page 565](#)
- [Sample Service Templates on page 573](#)

Device and Service Template Configuration Overview (SRC CLI)

To configure dynamic authorization using the SIC you need to configure:

- **Device template**—Specifies the router make, model and capability.
- **Service template**—Specifies any services that you want to enable for your router. What services are available vary from router to router, so it is important that you understand the properties of your router to successfully implement custom services.
- **Global service template**—Specifies rendering used as part of any mode of any service template. Global service templates are used to control rendering of service-independent requests, such as Abort-Session. A global service template is unique in that its modes, attributes, and variables are available to all services that you define. Global service templates are therefore a mandatory part of any SIC COA configuration.

Device Template Configuration Overview (SRC CLI)

Device templates specify the activation behavior of services and how the router handles multiple requests.

To configure device templates, you specify the capability and its associated value. The associated value is dependent on the specified capability. [Table 38 on page 540](#) describes the available capabilities and associated values.

Table 38: Device Template Capabilities and Associated Values

Capability	Value
Activation —Specify service access/activation behavior.	None (default value)—Indicates that the router is not capable of activating services.
	Access-Accept —Indicates that the router supports activating services only for access-accept messages.
	CoA —Indicates that the router supports activating services in CoA only.
	Both —Enables both Access-Accept and CoA requests.
Modification —Specify service modification behavior.	False (default value)—This attribute must be set to false.
Bundle —Indicates whether and how the router handles multiple service activation/deactivations in one CoA.	None (default value)—Indicates no bundling.
	Single —Indicates the router accepts multiple requests.

Service and Global Service Template Configuration Overview (SRC CLI)

Service templates specify any services that you want to enable for your router. What services are available vary from router to router, so it is important that you understand the properties of your router to successfully implement custom services.

Global service templates specify rendering used as part of any mode of any service template. Global service templates are used to control rendering of service-independent requests, such as Abort-Session. A global service template is unique in that its modes, attributes, and variables are available to all services that you define. Global service templates are therefore a mandatory part of any SIC CoA configuration.

You need to configure the following items for both the service and global service template:

- Mode
- Attributes
- Variable

Mode

Service and global service templates have groups of data called mode that each service must specify. A mode contains attributes and variables, which are explained in the next sections. It is mandatory to configure the mode for each service and global template. You must use the provided modes; you cannot create new modes.

[Table 39 on page 541](#) lists the modes and attributes for global service templates.

Table 39: Service Template Modes

Mode	Description
Activation	Activates services on request from the SAE.
Deactivation	Deactivates services on request from the SAE.
Initial-Authorization	Initial activation of services in the Access-Accept message.
Service-Correlation-Id	Assigns an ID number when any other mode is initiated. The SRC software uses this identification number internally.
Service-Profile-Download	Used for Cisco routers only. See “Caveat (Cisco Only)” on page 541 .

Table 40 on page 541 lists the modes and attributes for global service templates.

Table 40: Global Service Template Modes

Mode	Description
Authentication	<p>Use this mode for optional rendering of the request in the case of an Initial-Authorization. Usually this mode is empty, since no additional rendering is required.</p> <p>Unlike modes in service templates, this mode renders requests to the SRC software and not to the router.</p>
Accounting	Use this mode to control the rendering of the accounting request sent to the SAE. Accounting is a post-authorization service, and it uses the ID numbers and names from the service activation rendering.
Abort-Session	Use this mode for rendering of RADIUS disconnect request (DM) upon abort session request from the SAE.

Caveat (Cisco Only)

Cisco routers require an additional step to complete service activation. When the SIC activates a service on a Cisco router, the router sends an extra Access-Request to the SIC to retrieve the service profile. The SIC then sends back an Access-Accept response with VSAs representing the service profile. In response to the extra Access-Request, the SIC has to send all VSAs generated by the previous rendering process. The router then activates the service. This means that the SIC has to render the activation twice. In the second rendering a special mode, Service-Profile-Download is used.

This activation process is different from the usual scenario. Extra Access-Requests happen prior to the SIC response to an SAE request. Therefore, you can minimize the first rendering and place most of the work on the SAE download mode by doing the following:

The **Service-Profile-Download** mode in the supplied Cisco router configuration template is used to render the answer to the Cisco Profile Download request. The **Initial-Authorization** or **Activation** modes are used to render the first Access-Accept or COA message in the packet. To comply with the Cisco requirement to have only the

service name in the first Access-Accept or COA message, the **Initial-Authorization** or **Activation** modes should contain the attribute for the service name only, and the rest of parameters should be specified using the **shared sic group identifier device-template id service-template name mode service-profile-download** statement.

- In the activation mode, specify only the service name.
- In the service-policy-download mode, specify the rest of the needed parameters.

See your Cisco documentation for more information.

Attributes

All modes have attributes. Attributes define which RADIUS attributes are generated as a result of rendering. All attributes create data that appears in the RADIUS attributes (such as VSAs) generated by the rendering process. It is important to understand that modes are the very core of the rendering process.

[Table 41 on page 542](#) lists the attributes, explains their parameters, and describes their behavior.

Table 41: Attributes for All Modes

Attribute	Description
required	<p>If the renderer finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message for the router. Otherwise, the rendering fails.</p> <p>Options</p> <ul style="list-style-type: none"> • name name—Name of the attribute. The specified name must match a defined RADIUS attribute in the downstream AAA server response. • copy-from copy-from—(Optional) Specify the name of the attribute to copy the value from. If the copy-from option is specified, the renderer looks up the attribute specified by copy-from option in the downstream AAA Server response. In the absence of copy-from option, the renderer looks up the attribute specified by the name option.
override	<p>Whether or not the renderer finds the attribute in the downstream AAA server response, it creates the attribute name with the specified value.</p> <p>Options</p> <ul style="list-style-type: none"> • name name —Name of the attribute. The name must match a defined RADIUS attribute in the downstream AAA server response. • value value —Set the attribute to this value.

Table 41: Attributes for All Modes (*continued*)

Attribute	Description
default	<p>If the renderer finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message. Otherwise, it creates the attribute name with the specified value.</p> <p>Options</p> <ul style="list-style-type: none"> • name <i>name</i> —Name of the attribute. The name must match a defined RADIUS attribute in the downstream AAA server response. • value <i>value</i> —Set the attribute to this value. • copy-from <i>copy-from</i>—(Optional) Specify the name of the attribute to copy the value from. If the copy-from option is specified, the renderer looks up the attribute specified by copy-from option in the downstream AAA Server response. In the absence of copy-from option, the renderer looks up the attribute specified by the name option.
normal	<p>If the renderer finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message for the router. Otherwise, no action occurs. Unlike <i>required-attribute</i>, the rendering does not fail in this case.</p> <p>Options</p> <ul style="list-style-type: none"> • name <i>name</i>—Name of the attribute. The specified name must match a defined RADIUS attribute in the downstream AAA server response. • copy-from <i>copy-from</i>—(Optional) Specify the name of the attribute to copy the value from. If the copy-from option is specified, the renderer looks up the attribute specified by copy-from option in the downstream AAA Server response. In the absence of copy-from option, the renderer looks up the attribute specified by the name option.

Table 41: Attributes for All Modes (*continued*)

Attribute	Description
parameterized	<p>The most powerful and flexible part of the template. It generates attribute values using a format specification, which makes it the most flexible of the attributes.</p> <p>Options</p> <ul style="list-style-type: none"> • name <i>name</i>—Name of the attribute. The specified name must match a defined RADIUS attribute in the downstream AAA server response. • format <i>format</i>—In a form of "\$ (p1) \$ (p2) ... \$ (pn) [\$p(n+1)]". Behaves much like <code>sprintf</code> in C; you can intersperse literal text in between parameter definitions. Unlike <code>sprintf</code>, <code>format</code> supports an optional parameter definition. If the optional parameter is absent, it, and any literal text included in the square brackets, is ignored. All parameters come from the SAE as input to rendering. If you need to use restricted characters in your strings, use the backslash convention: <code>\\$, \', \", \[, \], \[, \]</code>.

Variables

Modes can also have variables, which control the rendering process. Variables are subtags under modes. You can use them to render information that is not part of RADIUS attributes. They provide inner logic for the rendering process. Nothing defined by variables appears in VSAs sent to the router.

Variables have three configuration options, described in [Table 42 on page 544](#).

Table 42: Variables

Option	Description
name	The variable name
value	The value, usually an integer
type	The data type, integer or string

A rule for processing variables: while rendering, when the SIC encounters a variable with a new value, and that variable already has a different value, the rendering stops and sends the results to the SAE. The SAE generates a RADIUS message and resumes rendering with the new value. Thus, it creates two VSAs, one each for the variable values. This correlates with the Bundle capability.

Overriding the Service Correlation ID

You can also use variables to override the **service-correlation-id** mode. For example,

```
variable name= "CreateServiceCorrelationId" value="0"
```


overrides the **service-correlation-id** mode, so no identification number is created.

Tagged Attributes

The SIC supports tagged attributes, which are an extension of the RADIUS protocol. Refer to RFC 2868 (<http://www.ietf.org/rfc/rfc2868>) for a description of this feature.

If you have **bundle=single** and you want to send a single COA activating two services, these activation requests must have the same RADIUS attributes, but with different values. To discriminate between attributes from two separate activation requests, you must use a unique tag for each.

Specify tagged attributes using the **shared sic group *identifier* device-template *id* service-template *name* mode (activation|deactivation|initial-authorization|service-correlation-id|service-profile-download) attributes tagged-group *name*** statement.



NOTE: Each service template is restricted to have only one tagged group; for attributes configured under the tagged-group, only attributes that support tags are affected. Otherwise, it has no effect if the configured attributes does not support tagging.

The attributes described in [Table 41 on page 542](#) are also support for tagged attribute configurations.

Related Documentation

- [Managing Dynamic Services on page 434](#)
- [SIC Dynamic Authorization Support Overview on page 435](#)
- [How the Dynamic Authorization Process Works in the SIC on page 437](#)

Configuring Device Templates (SRC CLI)

Device templates specify the make (vendor), model, and capability of the router. Device models are stored in the Juniper Networks database and can be shared by multiple SICs.



NOTE: When you modify a device template, you must restart the SIC to apply the changes.

Before you configure the device template, you need to configure the device models and dictionaries used by the SIC group. See “[Configuring the Device Models Supported by the SIC Group \(SRC CLI\)](#)” on page 480 and “[Configuring Dictionaries for the SIC Group \(SRC CLI\)](#)” on page 478.

Use the following statements to configure a device template for the SIC:

```
shared sic group identifier device-template id {
  vendor vendor;
  model model;
```

```
}
```

To configure a device template for the SIC:

1. From configuration mode, access the statement that configures the device template and specify a name for the template. For example, to create a device template named *dt1* in an SIC group named *g1*:

```
[edit]
```

```
user@host# edit shared sic group g1 device-template dt1
```

We provide templates for Juniper Networks E Series Broadband Services Routers running JunosE Software release 7.2 or later and for Cisco routers running Cisco IOS Release 12.2SB. These templates include sample global and service templates that you can modify for your specific environment. To specify the Juniper Networks or Cisco template, enter the following device template names:

- *juniper-router-junose-7.2-plus*
- *cisco-router-ios-12.2-sb*

2. (Optional) Specify the vendor supported in the device template.

```
[edit shared sic group g1 device-template dt1]
```

```
user@host# set vendor vendor
```

3. (Optional) Specify the device model name supported in the device template.

```
[edit shared sic group g1 device-template dt1]
```

```
user@host# set model model
```

Related Documentation

- [SIC Dictionaries and Device Models Overview \(SRC CLI\) on page 463](#)
- [Configuring the Device Capabilities Supported in the Device Template \(SRC CLI\) on page 546](#)
- [Device and Service Template Configuration Overview \(SRC CLI\) on page 539](#)
- [Sample Service Templates on page 573](#)

Configuring the Device Capabilities Supported in the Device Template (SRC CLI)

Device capabilities specify access behavior, modification of the existing service, and whether multiple COAs can attach to one VSA.

Use the following statements to configure the device capabilities:

```
shared sic group identifier device-template id capabilities capability (activation |  
modification | bundle) {  
    value;  
}
```

To configure device capabilities, you specify the capability and its associated value. The associated value is dependent on the specified capability. [Table 43 on page 547](#) describes the available capabilities and associated values.

Table 43: Capabilities and Associated Values

Capability	Value
Activation —Specify service access or activation behavior.	None (default value)—Indicates that the router is not capable of activating services during initial authorization or activation.
	Access-Accept —Indicates that the router supports activating services only in RADIUS Access-Accept messages.
	COA —Indicates that the router supports activating services in COA only.
	Both —Enables both Access-Accept and COA requests.
Modification —Specify service modification behavior.	False (default value)—This attribute must be set to false.
Bundle —Indicates whether and how the router handles multiple service activations or deactivations in one COA.	None (default value)—Indicates no bundling.
	Single —Indicates that the router accepts multiple requests.

To configure the device capabilities:

1. From configuration mode, access the statement that configures the device capabilities and specify the capability you want to configure. The following sample procedure uses `g1` as the SIC group name and `dt1` as the device template name.

```
[edit]
user@host# edit shared sic group identifier device-template dt1 capabilities capability
(activation | modification | bundle)
```

For example, to specify the **bundle** capability with a value of **single**, enter:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 capabilities capability bundle
```

2. Specify a value for the capability. For example, to set the **bundle** capability to the value of **single**:

```
[edit shared sic group g1 device-template dt1 capabilities capability bundle]
user@host# set single
```

Related Documentation

- [Configuring Device Templates \(SRC CLI\) on page 545](#)
- [Device and Service Template Configuration Overview \(SRC CLI\) on page 539](#)
- [Sample Service Templates on page 573](#)

Configuration Statements for SIC Service Templates (SRC CLI)

Use the following statements to configure service templates:

```

shared sic group identifier device-template id service-template name {
  description description;
}
shared sic group identifier device-template id service-template name mode (activation |
  deactivation | initial-authorization | service-correlation-id | service-profile-download)
shared sic group identifier device-template id service-template name mode (activation |
  deactivation | initial-authorization | service-correlation-id | service-profile-download)
  variable name {
    value value;
    type (integer | string);
  }
shared sic group identifier device-template id service-template name mode (activation |
  deactivation | initial-authorization | service-correlation-id | service-profile-download)
  attributes {
  }
shared sic group identifier device-template id service-template name mode (activation |
  deactivation | initial-authorization | service-correlation-id | service-profile-download)
  attributes attribute id
shared sic group identifier device-template id service-template name mode (activation |
  deactivation | initial-authorization | service-correlation-id | service-profile-download)
  attributes attribute id required {
    name name;
    copy-from copy-from;
  }
shared sic group identifier device-template id service-template name mode (activation |
  deactivation | initial-authorization | service-correlation-id | service-profile-download)
  attributes attribute id normal {
    name name;
    copy-from copy-from;
  }
shared sic group identifier device-template id service-template name mode (activation |
  deactivation | initial-authorization | service-correlation-id | service-profile-download)
  attributes attribute id default {
    name name;
    value value;
    copy-from copy-from;
  }
shared sic group identifier device-template id service-template name mode (activation |
  deactivation | initial-authorization | service-correlation-id | service-profile-download)
  attributes attribute id parameterized {
    format format;
    name name;
  }
shared sic group identifier device-template id service-template name mode (activation |
  deactivation | initial-authorization | service-correlation-id | service-profile-download)
  attributes attribute id override {
    name name;
    value value;
  }

```

- Related Documentation**
- [Device and Service Template Configuration Overview \(SRC CLI\) on page 539](#)
 - [Configuring the Device Capabilities Supported in the Device Template \(SRC CLI\) on page 546](#)
 - [Configuring Tagged Attributes in SIC Service Templates \(SRC CLI\) on page 558](#)
 - [Configuring Global Service Templates \(SRC CLI\) on page 565](#)

Configuring SIC Service Templates (SRC CLI)

Service templates are used to specify any services that you want to enable for your router. What services are available vary from router to router, so it is important that you understand the properties of your router to successfully implement custom services.

When you configure a service template, you need to specify the mode, and any variables or attributes you want included in the template. :

Refer to “[Device and Service Template Configuration Overview \(SRC CLI\)](#)” on page 539 for details on configuring the options in the following procedure.

- [Creating an SIC Service Template \(SRC CLI\) on page 549](#)
- [Configuring the Mode of the SIC Service Template \(SRC CLI\) on page 550](#)
- [Configuring Variables for the SIC Service Template \(SRC CLI\) on page 550](#)
- [Configuring Normal Attributes for the SIC Service Template \(SRC CLI\) on page 551](#)
- [Configuring Required Attributes for the SIC Service Template \(SRC CLI\) on page 552](#)
- [Configuring Default Attributes for the SIC Service Template \(SRC CLI\) on page 553](#)
- [Configuring Parameterized Attributes for the SIC Service Template \(SRC CLI\) on page 554](#)
- [Configuring Override Attributes for the SIC Service Template \(SRC CLI\) on page 555](#)

Creating an SIC Service Template (SRC CLI)

Use the following statements to create an SIC service template:

```
shared sic group identifier device-template id service-template name {
    description description;
}
```

To create an SIC service template:

1. From configuration mode, access the statement that configures the service template and specify the name of the template.

```
[edit]
user@host# edit shared sic group identifier device-template id service-template name
```

For example, to specify a service template called st1:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 service-template st1
```

2. (Optional) Specify a description for the template.

```
[edit shared sic group g1 device-template dt1 service-template st1]
user@host# set description description
```

Configuring the Mode of the SIC Service Template (SRC CLI)

Use the following statements to configure the mode of service template:

```
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
```

To configure the mode of the SIC service template:

- From configuration mode, access the statement that configures the service template mode. For example, to specify the **activation** mode:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode
activation
```

Configuring Variables for the SIC Service Template (SRC CLI)

Variables control the behavior of the rendering process.

Use the following statements to configure service template variables:

```
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
variable name {
  value value;
  type (integer | string);
}
```

To configure variables in the service template:

1. From configuration mode, access the statement that configures variables for the service template and specify a name for the variable. For example, to create a variable named `var1`:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode
activation variable var1
```

Specify the type of variable you want to add to the template. For example, to specify an integer for the variable:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
variable var1]
user@host# set type integer
```

Where the type is either:

- integer

- string
2. Specify the value of the variable. For example, to specify a value of 5 for the variable:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
variable var1]
user@host# set value 5
```

Configuring Normal Attributes for the SIC Service Template (SRC CLI)

```
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes {
}
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes attribute id
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes attribute id normal {
name name;
copy-from copy-from;
}
```

To configure normal attributes to be included in the service template:

1. (Optional) From configuration mode, access the statement that configures normal attributes and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes attribute attr1
```

2. (Optional) Specify the attribute as a normal attribute.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1]
user@host# edit normal
```

3. Specify the name of the attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 normal]
user@host# set name Unisphere-Service-Timeout
```

4. (Optional) Specify the attribute to copy the value from. For example, to copy the value from the Session-Timeout attribute contained in the downstream AAA server response, and place it in the Unisphere-Service-Timeout attribute:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 normal]
user@host# set copy-from Session-Timeout
```

5. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 normal]
user@host# show
```

```
copy-from Session-Timeout;
name Unisphere-Service-Timeout;
```

```
[edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes attribute attr1 normal]
user@host#
```

Configuring Required Attributes for the SIC Service Template (SRC CLI)

With required attributes, if the renderer finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message for the router, otherwise, rendering fails.

```
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes {
}
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes attribute id
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes attribute id required {
name name;
copy-from copy-from;
}
```

To configure required attributes to be included in the service template:

1. (Optional) From configuration mode, access the statement that configures required attributes and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes attribute attr1
```

2. (Optional) Specify the attribute as a required attribute.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1]
user@host# edit required
```

3. Specify the name of the attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 required]
user@host# set name Unisphere-Service-Timeout
```


4. (Optional) Specify the attribute to copy the value from. For example, to copy the value from the Session-Timeout attribute contained in the downstream AAA server response, and place it in the Unisphere-Service-Timeout attribute:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 required]
user@host# set copy-from Session-Timeout
```

5. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 required]
user@host# show
```

```
copy-from Session-Timeout;
name Unisphere-Service-Timeout;
```

```
[edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes attribute attr1 required]
user@host#
```

Configuring Default Attributes for the SIC Service Template (SRC CLI)

With default attributes, if the renderer finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message. Otherwise, it creates the attribute name with the specified value.

```
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes {
}
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes attribute id
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes attribute id default {
name name;
value value;
copy-from copy-from;
}
```

To configure default attributes to be included in the service template:

1. (Optional) From configuration mode, access the statement that configures default attributes and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes attribute attr1
```

2. (Optional) Specify the attribute as a default attribute.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1]
```

```
user@host# edit default
```

3. Specify the name of the attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
 attributes attribute attr1 default]
user@host# set name Unisphere-Service-Timeout
```

4. Specify the value of the attribute. For example, to specify the value of 5:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
 attributes attribute attr1 default]
user@host# set value 5
```

5. (Optional) Specify the attribute to copy the value from. For example, to copy the value from the Session-Timeout attribute contained in the downstream AAA server response, and place it in the Unisphere-Service-Timeout attribute:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
 attributes attribute attr1 default]
user@host# set copy-from Session-Timeout
```

If the rendering process finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message. Otherwise, it creates the attribute name with the specified value.

6. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
 attributes attribute attr1 default]
user@host# show
```

```
copy-from Session-Timeout;
name Unisphere-Service-Timeout;
value 5;
```

```
[edit shared sic group g1 device-template dt1 service-template st1 mode
 activation attributes attribute attr1 default]
user@host#
```

Configuring Parameterized Attributes for the SIC Service Template (SRC CLI)

```
shared sic group identifier device-template id service-template name mode (activation |
 deactivation | initial-authorization | service-correlation-id | service-profile-download)
 attributes {
 }
shared sic group identifier device-template id service-template name mode (activation |
 deactivation | initial-authorization | service-correlation-id | service-profile-download)
 attributes attribute id
shared sic group identifier device-template id service-template name mode (activation |
 deactivation | initial-authorization | service-correlation-id | service-profile-download)
 attributes attribute id parameterized {
 format format;
```

```

    name name;
}

```

To configure parameterized attributes to be included in the service template:

1. (Optional) From configuration mode, access the statement that configures parameterized attributes and specify an identifier for the attribute. For example, to create an identifier named `attr1`:

```

[edit]
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes attribute attr1

```

2. (Optional) Specify the attribute as a parameterized attribute.

```

[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1]
user@host# edit parameterized

```

3. Specify the format of the parameterized attribute.

```

[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 parameterized]
user@host# set format format

```

4. Specify the name of the attribute.

```

[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 parameterized]
user@host# set name name

```

5. Verify the configuration.

```

[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 parameterized]
user@host# show

```

```

copy-from Session-Timeout;
name Unisphere-Service-Timeout;

```

```

[edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes attribute attr1 parameterized]
user@host#

```

Configuring Override Attributes for the SIC Service Template (SRC CLI)

With override attributes, whether or not the renderer finds the attribute in the downstream AAA server response, it creates the attribute name with the specified value.

```

shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes {
}
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes attribute id

```

```
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes attribute id override {
  name name;
  value value;
}
```

To configure override attributes to be included in the service template:

1. (Optional) From configuration mode, access the statement that configures override attributes and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes attribute attr1
```

2. (Optional) Specify the attribute as a override attribute.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1]
user@host# edit override
```

3. Specify the name of the override attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 override]
user@host# set name Unisphere-Service-Timeout
```

4. Specify the value of the attribute. For example, to specify a value of 5:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 override]
user@host# set value 5
```

5. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes attribute attr1 override]
user@host# show
```

```
name Unisphere-Service-Timeout;
value 5;
```

```
[edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes attribute attr1 override]
user@host#
```

Related Documentation

- [Configuring Device Templates \(SRC CLI\) on page 545](#)
- [Device and Service Template Configuration Overview \(SRC CLI\) on page 539](#)
- [Configuring the Device Capabilities Supported in the Device Template \(SRC CLI\) on page 546](#)

- [Configuring Tagged Attributes in SIC Service Templates \(SRC CLI\) on page 558](#)
- [Configuring Global Service Templates \(SRC CLI\) on page 565](#)

Configuration Statements for Tagged Attributes in SIC Service Templates (SRC CLI)

Use the following statements to configure tagged attributes in a service template:

```
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group {
}
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group attribute id
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group attribute id required {
name name;
copy-from copy-from;
}
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group attribute id normal {
name name;
copy-from copy-from;
}
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group attribute id default {
name name;
value value;
copy-from copy-from;
}
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group attribute id parameterized {
format format;
name name;
}
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group item id override {
name name;
value value;
}
```

Related Documentation

- [Device and Service Template Configuration Overview \(SRC CLI\) on page 539](#)
- [Configuring the Device Capabilities Supported in the Device Template \(SRC CLI\) on page 546](#)
- [Configuring Tagged Attributes in SIC Service Templates \(SRC CLI\) on page 558](#)
- [Configuring Global Service Templates \(SRC CLI\) on page 565](#)

Configuring Tagged Attributes in SIC Service Templates (SRC CLI)

The examples in the following procedures use the following:

- sic group=g1
- device template=dt1
- service template=st1
- mode=activation
- attribute identifier=attr1
- attribute=Unisphere-Service-Timeout
- [Creating a Tagged Attribute Group in the SIC Service Template \(SRC CLI\) on page 558](#)
- [Configuring Normal Attributes in a Tagged Attribute Group \(SRC CLI\) on page 559](#)
- [Configuring Default Attributes in a Tagged Attribute Group \(SRC CLI\) on page 560](#)
- [Configuring Required Attributes in a Tagged Attribute Group \(SRC CLI\) on page 561](#)
- [Configuring Override Attributes in a Tagged Attribute Group \(SRC CLI\) on page 562](#)
- [Configuring Parameterized Attributes in a Tagged Attribute Group \(SRC CLI\) on page 563](#)

Creating a Tagged Attribute Group in the SIC Service Template (SRC CLI)

Use the following statements to create a tagged attribute group in the SIC service template:

```
shared sic group identifier device-template id service-template name mode (activation |  
    deactivation | initial-authorization | service-correlation-id | service-profile-download)  
    attributes tagged-group {  
    }
```

To create a tagged attribute group in the SIC service template:

- From configuration mode, access the statement that configures the tagged attribute group in the service template.

```
[edit]  
user@host# edit shared sic group identifier device-template id service-template name  
    mode (activation | deactivation | initial-authorization | service-correlation-id |  
    service-profile-download) attributes tagged-group
```

Attributes defined within tagged attributes will be tagged when included in the render result if this attribute supports tagging.

Configuring Normal Attributes in a Tagged Attribute Group (SRC CLI)

With normal attributes, if the renderer finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message for the router. Otherwise, no action occurs. Unlike required attributes, the rendering does not fail in this case.

```
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group attribute id
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group attribute id normal {
  name name;
  copy-from copy-from;
}
```

To configure normal attributes in the tagged attribute group:

1. (Optional) From configuration mode, access the statement that configures normal attributes in the tagged attribute group and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes tagged-group attribute attr1
```

2. (Optional) Specify the attribute as a normal attribute.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1]
user@host# edit normal
```

3. Specify the name of the attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 normal]
user@host# set name Unisphere-Service-Timeout
```

4. (Optional) Specify the attribute to copy the value from. For example, to copy the value from the Session-Timeout attribute contained in the downstream AAA server response, and place it in the Unisphere-Service-Timeout attribute:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 normal]
user@host# set copy-from Session-Timeout
```

5. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 normal]
user@host# show
```

```
copy-from Session-Timeout;
name Unisphere-Service-Timeout;

[edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes tagged-group attribute attr1 normal]
user@host#
```

Configuring Default Attributes in a Tagged Attribute Group (SRC CLI)

With default attributes, if the renderer finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message. Otherwise, it creates the attribute name with the specified value.

```
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group attribute id
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group attribute id default {
  name name;
  value value;
  copy-from copy-from;
}
```

To configure default attributes in the tagged attribute group:

1. (Optional) From configuration mode, access the statement that configures default attributes in the tagged attribute group and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes tagged-group attribute attr1
```

2. (Optional) Specify the attribute as a default attribute.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1]
user@host# edit default
```

3. Specify the name of the attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 default]
user@host# set name Unisphere-Service-Timeout
```

4. Specify the value of the attribute. For example, to specify a value of 5:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 default]
user@host# set value 5
```


5. (Optional) Specify the attribute to copy the value from. For example, to copy the value from the Session-Timeout attribute contained in the downstream AAA server response, and place it in the Unisphere-Service-Timeout attribute:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
  attributes tagged-group attribute attr1 default]
user@host# set copy-from Session-Timeout
```

6. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
  attributes tagged-group attribute attr1 default]
user@host# show
```

```
copy-from Session-Timeout;
name Unisphere-Service-Timeout;
value 5;
```

```
[edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes tagged-group attribute attr1 default]
user@host#
```

Configuring Required Attributes in a Tagged Attribute Group (SRC CLI)

With required attributes; if the renderer finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message for the router. otherwise, the renderer fails.

```
shared sic group identifier device-template id service-template name mode (activation |
  deactivation | initial-authorization | service-correlation-id | service-profile-download)
  attributes tagged-group attribute id
shared sic group identifier device-template id service-template name mode (activation |
  deactivation | initial-authorization | service-correlation-id | service-profile-download)
  attributes tagged-group attribute id required {
    name name;
    copy-from copy-from;
  }
```

To configure required attributes in the tagged attribute group:

1. (Optional) From configuration mode, access the statement that configures required attributes in the tagged attribute group and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode
  activation attributes tagged-group attribute attr1
```

2. (Optional) Specify the attribute as a required attribute.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
  attributes tagged-group attribute attr1]
user@host# edit required
```

3. Specify the name of the attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 required]
user@host# set name Unisphere-Service-Timeout
```

4. Specify the attribute to copy the value from. For example, to copy the value from the Session-Timeout attribute contained in the downstream AAA server response, and place it in the Unisphere-Service-Timeout attribute:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 required]
user@host# set copy-from Session-Timeout
```

5. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 required]
user@host# show
```

```
copy-from Session-Timeout;
name Unisphere-Service-Timeout;
```

```
[edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes tagged-group attribute attr1 required]
user@host#
```

Configuring Override Attributes in a Tagged Attribute Group (SRC CLI)

With override attributes, whether or not the renderer finds the attribute in the downstream AAA server response, it creates the attribute name with the specified value.

```
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group attribute id
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group attribute id override {
  name name;
  value value;
}
```

To configure override attributes in the tagged attribute group:

1. (Optional) From configuration mode, access the statement that configures override attributes in the tagged attribute group and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes tagged-group attribute attr1
```

2. (Optional) Specify the attribute as a override attribute.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1]
user@host# edit override
```

3. Specify the name of the attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 override]
user@host# set name Unisphere-Service-Timeout
```

4. Specify the value of the attribute. For example, to specify a value of 5:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 override]
user@host# set value 5
```

5. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 override]
user@host# show
```

```
name Unisphere-Service-Timeout;
value 5;
```

```
[edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes tagged-group attribute attr1 override]
user@host#
```

Configuring Parameterized Attributes in a Tagged Attribute Group (SRC CLI)

Parameterized attributes is the most powerful and flexible part of the template. It generates attribute values using a format specification, which makes it the most flexible of the attributes.

```
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group name attribute id
shared sic group identifier device-template id service-template name mode (activation |
deactivation | initial-authorization | service-correlation-id | service-profile-download)
attributes tagged-group name attribute id parameterized {
format format;
name name;
}
```

To configure parameterized attributes in the tagged attribute group:

1. (Optional) From configuration mode, access the statement that configures parameterized attributes in the tagged attribute group and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes tagged-group attribute attr1
```

2. (Optional) Specify the attribute as a normal attribute.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1]
user@host# edit parameterized
```

3. Specify the name of the attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 parameterized]
user@host# set name Unisphere-Service-Timeout
```

4. Specify the format of the parameterized attribute.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 parameterized]
user@host# set format format
```

5. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 service-template st1 mode activation
attributes tagged-group attribute attr1 parameterized]
user@host# show
```

```
name Unisphere-Service-Timeout;
```

```
[edit shared sic group g1 device-template dt1 service-template st1 mode
activation attributes tagged-group attribute attr1 parameterized]
user@host#
```

Related Documentation

- [Device and Service Template Configuration Overview \(SRC CLI\) on page 539](#)
- [Configuring the Device Capabilities Supported in the Device Template \(SRC CLI\) on page 546](#)
- [Configuring Global Service Templates \(SRC CLI\) on page 565](#)

Configuration Statements for SIC Global Service Templates (SRC CLI)

Use the following statements to configure a global service template:

```
shared sic group identifier device-template id global-template {
  description description;
}
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session)
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) variable name {
  value value;
  type (integer | string);
}
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes {
```

```

}
shared sic group identifier device-template id global-template mode (authentication |
  accounting | abort-session) attributes attribute id
shared sic group identifier device-template id global-template mode (authentication |
  accounting | abort-session) attributes attribute id required {
  name name;
  copy-from copy-from;
}
shared sic group identifier device-template id global-template mode (authentication |
  accounting | abort-session) attributes attribute id normal {
  name name;
  copy-from copy-from;
}
shared sic group identifier device-template id global-template mode (authentication |
  accounting | abort-session) attributes attribute id default {
  name name;
  value value;
  copy-from copy-from;
}
shared sic group identifier device-template id global-template mode (authentication |
  accounting | abort-session) attributes attribute id parameterized {
  format format;
  name name;
}
shared sic group identifier device-template id global-template mode (authentication |
  accounting | abort-session) attributes attribute id override {
  name name;
  value value;
}

```

Related Documentation

- [Device and Service Template Configuration Overview \(SRC CLI\) on page 539](#)
- [Configuring the Device Capabilities Supported in the Device Template \(SRC CLI\) on page 546](#)
- [Configuring Tagged Attributes in SIC Service Templates \(SRC CLI\) on page 558](#)
- [Configuring Global Service Templates \(SRC CLI\) on page 565](#)

Configuring Global Service Templates (SRC CLI)

A global service template is a unique service template that specifies rendering used as part of any mode of any other service template. It is used to control rendering of service-independent requests, such as AbortSession. This template is unique in that its modes, attributes, and variables are available to all services that you define. It is therefore a mandatory part of any router configuration. The global service template is called in every possible scenario.

The examples in this procedure use the following configuration:

- sic group=g1
- device template=dt1
- mode=authentication

- attribute identifier-attr1
- attribute=Unisphere-Service-Timeout
- [Creating an SIC Global Service Template \(SRC CLI\) on page 566](#)
- [Configuring the Mode of the SIC Global Service Template \(SRC CLI\) on page 566](#)
- [Configuring Variables for the SIC Global Service Template \(SRC CLI\) on page 567](#)
- [Configuring Normal Attributes for the SIC Global Service Template \(SRC CLI\) on page 567](#)
- [Configuring Required Attributes for the SIC Global Service Template \(SRC CLI\) on page 568](#)
- [Configuring Default Attributes for the SIC Global Service Template \(SRC CLI\) on page 569](#)
- [Configuring Parameterized Attributes for the SIC Global Service Template \(SRC CLI\) on page 571](#)
- [Configuring Override Attributes for the SIC Global Service Template \(SRC CLI\) on page 572](#)

Creating an SIC Global Service Template (SRC CLI)

Use the following statements to create an SIC global service template:

```
shared sic group identifier device-template id global-template {  
    description description;  
}
```

To create an SIC global service template:

1. From configuration mode, access the statement that configures the global.

```
[edit]  
user@host# edit shared sic group identifier device-template id global-template
```

2. (Optional) Specify a description for the template.

```
[edit shared sic group g1 device-template dt1 global-template]  
user@host# set description description
```

Configuring the Mode of the SIC Global Service Template (SRC CLI)

Use the following statements to configure the mode of global service template:

```
shared sic group identifier device-template id global-template mode (authentication |  
    accounting | abort-session)
```

To configure the mode of the SIC global service template:

- From configuration mode, access the statement that configures the global service template mode. For example, to specify the **authentication** mode:

```
[edit]  
user@host# edit shared sic group g1 device-template dt1 global-template mode  
    authentication
```

Configuring Variables for the SIC Global Service Template (SRC CLI)

Variables control the behavior of the rendering process.

Use the following statements to configure global service template variables:

```
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) variable name {
  value value;
  type (integer | string);
}
```

To configure variables in the global service template:

- From configuration mode, access the statement that configures variables for the global service template and specify a name for the variable. For example, to create a variable named `var1`:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 global-template mode
authentication variable var1
```

Specify the type of variable you want to add to the template. For example, to specify an integer for the variable:

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
variable var1]
user@host# set type integer
```

Where the type is either:

- integer
- string
- Specify the value of the variable. For example, to specify a value of 5 for the variable:

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
variable var1]
user@host# set value 5
```

Configuring Normal Attributes for the SIC Global Service Template (SRC CLI)

```
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes {
}
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id normal {
  name name;
  copy-from copy-from;
}
```

To configure normal attributes to be included in the global service template:

1. (Optional) From configuration mode, access the statement that configures normal attributes and specify an identifier for the attribute. For example, to create an identifier named `attr1`:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 global-template mode
authentication attributes attribute attr1
```

2. (Optional) Specify the attribute as a normal attribute.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1]
user@host# edit normal
```

3. Specify the name of the attribute. For example, to specify the attribute `Unisphere-Service-Timeout`:

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 normal]
user@host# set name Unisphere-Service-Timeout
```

4. (Optional) Specify the attribute to copy the value from. For example, to copy the value from the `Session-Timeout` attribute contained in the downstream AAA server response, and place it in the `Unisphere-Service-Timeout` attribute:

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 normal]
user@host# set copy-from Session-Timeout
```

5. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 normal]
user@host# show
```

```
copy-from Session-Timeout;
name Unisphere-Service-Timeout;
```

```
[edit shared sic group g1 device-template dt1 global-template mode
authentication attributes attribute attr1 normal]
user@host#
```

Configuring Required Attributes for the SIC Global Service Template (SRC CLI)

With required attributes, if the renderer finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message for the router, otherwise, rendering fails.

```
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes {
}
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id
```



```
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id required {
  name name;
  copy-from copy-from;
}
```

To configure required attributes to be included in the global service template:

1. (Optional) From configuration mode, access the statement that configures required attributes and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 global-template mode
authentication attributes attribute attr1
```

2. (Optional) Specify the attribute as a required attribute.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1]
user@host# edit required
```

3. Specify the name of the attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 required]
user@host# set name Unisphere-Service-Timeout
```

4. (Optional) Specify the attribute to copy the value from. For example, to copy the value from the Session-Timeout attribute contained in the downstream AAA server response, and place it in the Unisphere-Service-Timeout attribute:

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 required]
user@host# set copy-from Session-Timeout
```

5. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 required]
user@host# show
```

```
copy-from Session-Timeout;
name Unisphere-Service-Timeout;
```

```
[edit shared sic group g1 device-template dt1 global-template mode
authentication attributes attribute attr1 required]
user@host#
```

Configuring Default Attributes for the SIC Global Service Template (SRC CLI)

With default attributes, if the renderer finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message. Otherwise, it creates the attribute name with the specified value.

```
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes {
}
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id default {
name name;
value value;
copy-from copy-from;
}
```

To configure default attributes to be included in a global service template:

1. (Optional) From configuration mode, access the statement that configures default attributes and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 global-template mode
authentication attributes attribute attr1
```

2. (Optional) Specify the attribute as a default attribute.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1]
user@host# edit default
```

3. Specify the name of the attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 default]
user@host# set name Unisphere-Service-Timeout
```

4. Specify the value of the attribute. For example, to specify the value of 5:

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 default]
user@host# set value 5
```

5. (Optional) Specify the attribute to copy the value from. For example, to copy the value from the Session-Timeout attribute contained in the downstream AAA server response, and place it in the Unisphere-Service-Timeout attribute:

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 default]
user@host# set copy-from Session-Timeout
```

If the rendering process finds the attribute in the downstream AAA server response, it copies the value into the RADIUS message. Otherwise, it creates the attribute name with the specified value.

6. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 default]
user@host# show
```

```
copy-from Session-Timeout;
name Unisphere-Service-Timeout;
value 5;
```

```
[edit shared sic group g1 device-template dt1 global-template mode
authentication attributes attribute attr1 default]
user@host#
```

Configuring Parameterized Attributes for the SIC Global Service Template (SRC CLI)

```
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes {
}
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id parameterized {
format format;
name name;
}
```

To configure parameterized attributes to be included in a global service template:

1. (Optional) From configuration mode, access the statement that configures parameterized attributes and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 global-template mode
authentication attributes attribute attr1
```

2. (Optional) Specify the attribute as a parameterized attribute.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1]
user@host# edit parameterized
```

3. (Optional) Specify the format of the parameterized attribute.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 parameterized]
user@host# set format format
```

4. Specify the name of the attribute.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 parameterized]
user@host# set name name
```

5. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 parameterized]
user@host# show
```

```
name Unisphere-Service-Timeout;
```

```
[edit shared sic group g1 device-template dt1 global-template mode
authentication attributes attribute attr1 parameterized]
user@host#
```

Configuring Override Attributes for the SIC Global Service Template (SRC CLI)

With override attributes, whether or not the renderer finds the attribute in the downstream AAA server response, it creates the attribute name with the specified value.

```
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes {
}
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id
shared sic group identifier device-template id global-template mode (authentication |
accounting | abort-session) attributes attribute id override {
name name;
value value;
}
```

To configure override attributes to be included in a global service template:

1. (Optional) From configuration mode, access the statement that configures override attributes and specify an identifier for the attribute. For example, to create an identifier named attr1:

```
[edit]
user@host# edit shared sic group g1 device-template dt1 global-template mode
authentication attributes attribute attr1
```

2. (Optional) Specify the attribute as a override attribute.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1]
user@host# edit override
```

3. Specify the name of the override attribute. For example, to specify the attribute Unisphere-Service-Timeout:

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 override]
user@host# set name Unisphere-Service-Timeout
```

4. Specify the value of the attribute. For example, to specify a value of 5:

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 override]
user@host# set value 5
```

5. Verify the configuration.

```
[edit shared sic group g1 device-template dt1 global-template mode authentication
attributes attribute attr1 override]
user@host# show
```

```
name Unisphere-Service-Timeout;
value 5;
```

```
[edit shared sic group g1 device-template dt1 global-template mode
authentication attributes attribute attr1 override]
user@host#
```

Related Documentation

- [Device and Service Template Configuration Overview \(SRC CLI\) on page 539](#)
- [Configuring the Device Capabilities Supported in the Device Template \(SRC CLI\) on page 546](#)
- [Configuring Tagged Attributes in SIC Service Templates \(SRC CLI\) on page 558](#)

Sample Service Templates

We provide templates for Juniper Networks E Series Broadband Services Routers running JunosE Software release 7.2 or later and for Cisco routers running Cisco IOS Release 12.2SB. These templates include sample global and service templates that you can modify for your specific environment. Each sample includes a global service template and at least one service template.

Juniper Networks Routers Service Template

This example shows the complete Juniper Networks service template configuration. This template supports Juniper Networks E Series Broadband Services Routers running JunosE Software release 7.2 or later.

```
user@host# show device-template juniper-router-junose-7.2-plus
capabilities {
  capability activation {
    Both;
  }
  capability bundle {
    Single;
  }
  capability modification {
    false;
  }
}
model juniper-router-junose-7.2-plus;
global-template {
  description 'global section';
  mode abort-session {
    attributes {
      attribute Acct-Session-Id {
        required {
          name Acct-Session-Id;
        }
      }
    }
  }
}
```

```

    }
    variable RadiusCode {
        type integer;
        value 40;
    }
}
mode accounting {
    attributes;
}
mode authentication {
    attributes;
}
}
service-template content_provider_tiered {
    description 'content_provider_tiered service';
    mode activation {
        attributes {
            attribute Acct-Session-Id {
                required {
                    name Acct-Session-Id;
                }
            }
        }
        tagged-group {
            attribute Unisphere-Activate-Service {
                parameterized {
                    format 'content_provider_tiered\\($(contentProviderAddress),$(contentP
tProviderMask),$(subscriberAddress),$(subscriberMask),$(upstreamBandwidth),$(dow
nstreamBandwidth)\\)';
                    name Unisphere-Activate-Service;
                }
            }
            attribute Unisphere-Service-Stats {
                default {
                    name Unisphere-Service-Stats;
                    value 2;
                }
            }
        }
    }
    variable RadiusCode {
        type integer;
        value 43;
    }
}
mode deactivation {
    attributes {
        attribute Acct-Session-Id {
            required {
                name Acct-Session-Id;
            }
        }
        attribute Unisphere-Deactivate-Service {
            parameterized {
                format 'content_provider_tiered\\($(contentProviderAddress),$(contentP
roviderMask),$(subscriberAddress),$(subscriberMask),$(upstreamBandwidth),$(downs
treamBandwidth)\\)';
                name Unisphere-Deactivate-Service;
            }
        }
    }
    variable RadiusCode {

```

```

        type integer;
        value 43;
    }
}
mode initial-authorization {
    attributes {
        tagged-group {
            attribute Unisphere-Activate-Service {
                parameterized {
                    format 'content_provider_tiered\\($(contentProviderAddress),$(contentP
tProviderMask),$(subscriberAddress),$(subscriberMask),$(upstreamBandwidth),$(dow
nstreamBandwidth)\\)';
                    name Unisphere-Activate-Service;
                }
            }
            attribute Unisphere-Service-Stats {
                default {
                    name Unisphere-Service-Stats;
                    value 2;
                }
            }
        }
    }
}
mode service-correlation-id {
    attributes {
        attribute Juniper-Service-Correlation-Id {
            parameterized {
                format 'content_provider_tiered\\($(contentProviderAddress),$(contentP
roviderMask),$(subscriberAddress),$(subscriberMask),$(upstreamBandwidth),$(downs
treamBandwidth)\\)';
                name Juniper-Service-Correlation-Id;
            }
        }
    }
}
service-template internet_tiered {
    description 'internet_tiered service';
    mode activation {
        attributes {
            attribute Acct-Session-Id {
                required {
                    name Acct-Session-Id;
                }
            }
        }
        tagged-group {
            attribute Unisphere-Activate-Service {
                parameterized {
                    format 'internet_tiered\\($(upstreamBandwidth),$(downstreamBandwidth
)\\)';
                    name Unisphere-Activate-Service;
                }
            }
            attribute Unisphere-Service-Stats {
                default {
                    name Unisphere-Service-Stats;
                    value 2;
                }
            }
        }
    }
}

```

```

    }
    variable RadiusCode {
        type integer;
        value 43;
    }
}
mode deactivation {
    attributes {
        attribute Acct-Session-Id {
            required {
                name Acct-Session-Id;
            }
        }
        attribute Unisphere-Deactivate-Service {
            parameterized {
                format 'internet_tiered\\($(upstreamBandwidth),$(downstreamBandwidth)\\
\\)';
                name Unisphere-Deactivate-Service;
            }
        }
    }
    variable RadiusCode {
        type integer;
        value 43;
    }
}
mode initial-authorization {
    attributes {
        tagged-group {
            attribute Unisphere-Activate-Service {
                parameterized {
                    format 'internet_tiered\\($(upstreamBandwidth),$(downstreamBandwidth
)\\)';
                    name Unisphere-Activate-Service;
                }
            }
            attribute Unisphere-Service-Stats {
                default {
                    name Unisphere-Service-Stats;
                    value 2;
                }
            }
        }
    }
}
mode service-correlation-id {
    attributes {
        attribute Juniper-Service-Correlation-Id {
            parameterized {
                format 'internet_tiered\\($(upstreamBandwidth),$(downstreamBandwidth)\\
\\)';
                name Juniper-Service-Correlation-Id;
            }
        }
    }
}
}
service-template guided_entrance {
    description 'guided_entrance service';
    mode activation {
        attributes {

```



```

    attribute Acct-Session-Id {
        required {
            name Acct-Session-Id;
        }
    }
    tagged-group {
        attribute Unisphere-Activate-Service {
            parameterized {
                format 'guided_entrance\\($(redirectAddress),$(redirectPort),$(redirectRemainingUrl),$(originalAddress),$(originalMask),$(originalPort)\\)';
                name Unisphere-Activate-Service;
            }
        }
        attribute Unisphere-Service-Stats {
            default {
                name Unisphere-Service-Stats;
                value 1;
            }
        }
    }
}
variable RadiusCode {
    type integer;
    value 43;
}
}
mode deactivation {
    attributes {
        attribute Acct-Session-Id {
            required {
                name Acct-Session-Id;
            }
        }
        attribute Unisphere-Deactivate-Service {
            parameterized {
                format 'guided_entrance\\($(redirectAddress),$(redirectPort),$(redirectRemainingUrl),$(originalAddress),$(originalMask),$(originalPort)\\)';
                name Unisphere-Deactivate-Service;
            }
        }
    }
    variable RadiusCode {
        type integer;
        value 43;
    }
}
mode initial-authorization {
    attributes {
        tagged-group {
            attribute Unisphere-Activate-Service {
                parameterized {
                    format 'guided_entrance\\($(redirectAddress),$(redirectPort),$(redirectRemainingUrl),$(originalAddress),$(originalMask),$(originalPort)\\)';
                    name Unisphere-Activate-Service;
                }
            }
        }
        attribute Unisphere-Service-Stats {
            default {
                name Unisphere-Service-Stats;
                value 1;
            }
        }
    }
}

```

```
}  
}  
}  
}  
mode service-correlation-id {  
    attributes {  
        attribute Juniper-Service-Correlation-Id {  
            parameterized {  
                format 'guided_entrance\\($(redirectAddress),$(redirectPort),$(redirectRemainingUrl),$(originalAddress),$(originalMask),$(originalPort)\\)';  
                name Juniper-Service-Correlation-Id;  
            }  
        }  
    }  
}  
}  
}  
}  
}  
}  
}  
  
vendor juniper;  
  
[edit shared sic group test-group]  
user@host#
```

Cisco Router Service Template

This example shows the complete Cisco router service template configuration. This template supports Cisco routers running Cisco IOS Release 12.2SB.

```

user@host# show device-template cisco-router-ios-12.2-sb
capabilities {
  capability activation {
    Both;
  }
  capability bundle {
    None;
  }
  capability modification {
    false;
  }
}
model cisco-router-ios-12.2-sb;
global-template {
  description 'global section';
  mode abort-session {
    attributes {
      attribute Acct-Session-Id {
        required {
          name Acct-Session-Id;
        }
      }
    }
  }
  variable RadiusCode {
    type integer;
    value 40;
  }
}
mode accounting {
  attributes;
}
mode authentication {
  attributes;
}
}

```

```

service-template content_provider_tiered {
  description 'content_provider_tiered service';
  mode activation {
    attributes {
      attribute Acct-Session-Id {
        required {
          name Acct-Session-Id;
        }
      }
      attribute Cisco-SSG-Command-Code {
        override {
          name Cisco-SSG-Command-Code;
          value '{hex}0B 63 6F 6E 74 65 6E 74 5F 70 72 6F 76 69 64 65 72 5F 74
69 65 72 65 64';
        }
      }
      attribute Cisco-AVPair-1 {
        parameterized {
          format 'ip:inac1#10=permit ip any $(subscriberAddress)
$(subscriberMask)';
          name Cisco-AVPair;
        }
      }
      attribute Cisco-AVPair-2 {
        parameterized {
          format 'ip:outac1#20=permit ip $(contentProviderAddress)
$(contentProviderMask) any';
          name Cisco-AVPair;
        }
      }
      attribute Cisco-AVPair-3 {
        override {
          name Cisco-AVPair;
          value 'ip:traffic-class=in access-group name 10 priority 10';
        }
      }
      attribute Cisco-AVPair-4 {
        override {
          name Cisco-AVPair;
          value 'ip:traffic-class=in default drop';
        }
      }
      attribute Cisco-AVPair-5 {
        override {
          name Cisco-AVPair;
          value 'ip:traffic-class=out access-group name 20 priority 10';
        }
      }
      attribute Cisco-AVPair-6 {
        override {
          name Cisco-AVPair;
          value 'ip:traffic-class=out default drop';
        }
      }
      attribute Cisco-SSG-Service-Info {
        parameterized {
          format 'QU;$(upstreamBandwidth);;;D;$(downstreamBandwidth);';
          name Cisco-SSG-Service-Info;
        }
      }
      attribute Cisco-AVPair-7 {

```

```

        override {
            name Cisco-AVPair;
            value accounting-list=default;
        }
    }
    variable RadiusCode {
        type integer;
        value 43;
    }
}
mode deactivation {
    attributes {
        attribute Acct-Session-Id {
            required {
                name Acct-Session-Id;
            }
        }
        attribute Cisco-SSG-Command-Code {
            override {
                name Cisco-SSG-Command-Code;
                value '{hex}0C 63 6F 6E 74 65 6E 74 5F 70 72 6F 76 69 64 65 72 5F 74
69 65 72 65 64';
            }
        }
    }
    variable RadiusCode {
        type integer;
        value 43;
    }
}
mode service-correlation-id {
    attributes {
        attribute Juniper-Service-Correlation-Id {
            override {
                name Juniper-Service-Correlation-Id;
                value Ncontent_provider_tiered;
            }
        }
    }
}
mode service-profile-download {
    attributes {
        attribute User-Name {
            override {
                name User-Name;
                value content_provider_tiered;
            }
        }
    }
}
}
service-template internet_tiered {
    description 'internet_tiered service';
    mode activation {
        attributes {
            attribute Acct-Session-Id {
                required {
                    name Acct-Session-Id;
                }
            }
        }
    }
}

```

```

attribute Cisco-SSG-Command-Code {
  override {
    name Cisco-SSG-Command-Code;
    value '{hex}0B 69 6E 74 65 72 6E 65 74 5F 74 69 65 72 65 64';
  }
}
attribute Cisco-SSG-Service-Info {
  parameterized {
    format 'QU;$(upstreamBandwidth);;;D;$(downstreamBandwidth);';
    name Cisco-SSG-Service-Info;
  }
}
attribute Cisco-AVPair {
  override {
    name Cisco-AVPair;
    value accounting-list=default;
  }
}
}
variable RadiusCode {
  type integer;
  value 43;
}
}
mode deactivation {
  attributes {
    attribute Acct-Session-Id {
      required {
        name Acct-Session-Id;
      }
    }
  }
  attribute Cisco-SSG-Command-Code {
    override {
      name Cisco-SSG-Command-Code;
      value '{hex}0C 69 6E 74 65 72 6E 65 74 5F 74 69 65 72 65 64';
    }
  }
}
variable RadiusCode {
  type integer;
  value 43;
}
}
mode service-correlation-id {
  attributes {
    attribute Juniper-Service-Correlation-Id {
      override {
        name Juniper-Service-Correlation-Id;
        value Ninternet_tiered;
      }
    }
  }
}
}
mode service-profile-download {
  attributes {
    attribute User-Name {
      override {
        name User-Name;
        value internet_tiered;
      }
    }
  }
}

```

```
    }
  }
}
service-template guided_entrance {
  description 'guided_entrance service';
  mode activation {
    attributes {
      attribute Acct-Session-Id {
        required {
          name Acct-Session-Id;
        }
      }
      attribute Cisco-SSG-Command-Code {
        override {
          name Cisco-SSG-Command-Code;
          value '{hex}0B 67 75 69 64 65 64 5F 65 6E 74 72 61 6E 63 65';
        }
      }
      attribute Cisco-AVPair-1 {
        parameterized {
          format 'ip:inac1#10=permit ip any $(originalAddress) $(originalMask)';

          name Cisco-AVPair;
        }
      }
      attribute Cisco-AVPair-2 {
        parameterized {
          format 'ip:inac1#20=permit tcp any eq $(originalPort)';
          name Cisco-AVPair;
        }
      }
      attribute Cisco-AVPair-3 {
        override {
          name Cisco-AVPair;
          value 'ip:traffic-class=in access-group name 10 priority 10';
        }
      }
      attribute Cisco-AVPair-4 {
        override {
          name Cisco-AVPair;
          value 'ip:traffic-class=in access-group name 20 priority 10';
        }
      }
      attribute Cisco-AVPair-5 {
        parameterized {
          format 'ip:l4redirect=redirect to ip $(redirectAddress) port
$(redirectPort)';
          name Cisco-AVPair;
        }
      }
    }
  }
  variable RadiusCode {
    type integer;
    value 43;
  }
}
mode deactivation {
  attributes {
    attribute Acct-Session-Id {
      required {
        name Acct-Session-Id;
      }
    }
  }
}
```

```

    }
  }
  attribute Cisco-SSG-Command-Code {
    override {
      name Cisco-SSG-Command-Code;
      value '{hex}0C 67 75 69 64 65 64 5F 65 6E 74 72 61 6E 63 65';
    }
  }
}
variable RadiusCode {
  type integer;
  value 43;
}
}
mode service-correlation-id {
  attributes {
    attribute Juniper-Service-Correlation-Id {
      override {
        name Juniper-Service-Correlation-Id;
        value Nguided_entrance;
      }
    }
  }
}
mode service-profile-download {
  attributes {
    attribute User-Name {
      override {
        name User-Name;
        value guided_entrance;
      }
    }
  }
}
}
}
vendor cisco;

[edit shared sic group test-group]
user@host#

```

Related Documentation

- [Device and Service Template Configuration Overview \(SRC CLI\) on page 539](#)
- [Configuring the Device Capabilities Supported in the Device Template \(SRC CLI\) on page 546](#)
- [Configuring SIC Service Templates \(SRC CLI\) on page 549](#)
- [Configuring Global Service Templates \(SRC CLI\) on page 565](#)

Monitoring the Subscriber Information Collector with the SRC CLI

- [Viewing Statistics for RADIUS Client Accounting Requests \(SRC CLI\) on page 585](#)
- [Viewing Statistics for RADIUS Client Authentication Requests \(SRC CLI\) on page 585](#)
- [Viewing RADIUS Host Statistics for Accounting Transactions \(SRC CLI\) on page 586](#)
- [Viewing RADIUS Host Statistics for Authentication Transactions \(SRC CLI\) on page 586](#)
- [Viewing RADIUS Target Statistics for Accounting Requests \(SRC CLI\) on page 587](#)
- [Viewing RADIUS Target Statistics for Authentication Requests \(SRC CLI\) on page 587](#)
- [Viewing Diameter Host Statistics \(SRC CLI\) on page 588](#)
- [Viewing Diameter Peer Statistics \(SRC CLI\) on page 588](#)

Viewing Statistics for RADIUS Client Accounting Requests (SRC CLI)

Purpose View RADIUS client statistics for accounting requests. Statistics are presented for any client from which the server has received packets.

Action `user@host> show sic statistics radius client accounting`

Related Documentation

- [Viewing Statistics for RADIUS Client Authentication Requests \(SRC CLI\) on page 585](#)
- [Viewing RADIUS Host Statistics for Accounting Transactions \(SRC CLI\) on page 586](#)
- [Viewing RADIUS Target Statistics for Accounting Requests \(SRC CLI\) on page 587](#)

Viewing Statistics for RADIUS Client Authentication Requests (SRC CLI)

Purpose View RADIUS client statistics for authentication requests. Statistics are presented for any client from which the server has received packets.

Action `user@host> show sic statistics radius client authentication`

Related Documentation

- [Viewing RADIUS Host Statistics for Authentication Transactions \(SRC CLI\) on page 586](#)
- [Viewing RADIUS Target Statistics for Authentication Requests \(SRC CLI\) on page 587](#)

- [Viewing Statistics for RADIUS Client Accounting Requests \(SRC CLI\) on page 585](#)

Viewing RADIUS Host Statistics for Accounting Transactions (SRC CLI)

Purpose View RADIUS host statistics for accounting transactions, as well as server runtime and packet error statistics.

Action user@host> **show sic statistics radius host accounting**

```
RADIUS Host Accounting Statistics
Name as accounting server      SIC
Up time:                      6791110
Reset time:                    0
Server status:                 4
Requests:                     1660
Invalid requests:              0
Duplicate requests:            0
Responses:                     1660
Malformed requests:            0
Bad authenticators:            0
Packets dropped:                0
No records:                    0
Packets of unknown types:      0
Response from invalid addresses: 0
Name as accounting client      SIC
```

- Related Documentation**
- [Viewing Statistics for RADIUS Client Accounting Requests \(SRC CLI\) on page 585](#)
 - [Viewing RADIUS Target Statistics for Accounting Requests \(SRC CLI\) on page 587](#)
 - [Viewing RADIUS Host Statistics for Authentication Transactions \(SRC CLI\) on page 586](#)

Viewing RADIUS Host Statistics for Authentication Transactions (SRC CLI)

Purpose View RADIUS host statistics for authentication transactions, as well as server runtime and packet error statistics.

Action user@host> **show sic statistics radius host authentication**

```
RADIUS Host Authentication Statistics
Name as authentication server      SIC
Up time:                          6791110
Reset time:                        0
Server status:                     4
Requests:                          1660
Invalid requests:                   0
Duplicate requests:                 0
Access accepts:                     1620
Access rejects:                     20
Access challenges:                  20
Malformed requests:                 0
Bad authenticators:                 0
Packets dropped:                     0
No records:                         0
Packets of unknown types:           0
Response from invalid addresses: 0
Name as authentication client      SIC
```

- Related Documentation**
- [Viewing RADIUS Host Statistics for Accounting Transactions \(SRC CLI\) on page 586](#)
 - [Viewing Statistics for RADIUS Client Authentication Requests \(SRC CLI\) on page 585](#)
 - [Viewing RADIUS Target Statistics for Authentication Requests \(SRC CLI\) on page 587](#)

Viewing RADIUS Target Statistics for Accounting Requests (SRC CLI)

Purpose View RADIUS target statistics for accounting requests.

Action user@host> **show sic statistics radius target accounting**

- Related Documentation**
- [Viewing RADIUS Target Statistics for Authentication Requests \(SRC CLI\) on page 587](#)
 - [Viewing Statistics for RADIUS Client Accounting Requests \(SRC CLI\) on page 585](#)
 - [Viewing RADIUS Target Statistics for Authentication Requests \(SRC CLI\) on page 587](#)

Viewing RADIUS Target Statistics for Authentication Requests (SRC CLI)

Purpose View RADIUS target statistics for authentication requests. Statistics are available for RADIUS dynamic authorization and authentication targets that are defined in the server.

```

Action    user@host> show sic statistics radius target authentication
user@host> show sic statistics radius target authentication host 10.1.2.3
RADIUS Target Authentication Statistics
Index                0
Address Type          Unknown
Address               /10.1.2.3
Port                  0
Round trip time:      0
Requests:             0
Retransmitted requests: 0
Access accepts:       0
Access rejects:       0
Access challenges:    0
Malformed responses:  0
Bad authenticators:   0
Pending requests:     0
Timeouts:             0
Packets of unknown types: 0
Packets dropped:      0
Counter Discontinuity: 0

user@host>

```

- Related Documentation**

 - [Viewing RADIUS Target Statistics for Accounting Requests \(SRC CLI\) on page 587](#)
 - [Viewing RADIUS Host Statistics for Authentication Transactions \(SRC CLI\) on page 586](#)
 - [Viewing Statistics for RADIUS Client Authentication Requests \(SRC CLI\) on page 585](#)

Viewing Diameter Host Statistics (SRC CLI)

Purpose	View Diameter host statistics, including server runtime statistics and global summary statistics.
----------------	---

```
Action    user@host> show sic statistics diameter host
```

- Related Documentation**
- [Viewing Diameter Peer Statistics \(SRC CLI\) on page 588](#)
 - [Viewing RADIUS Host Statistics for Accounting Transactions \(SRC CLI\) on page 586](#)

Viewing Diameter Peer Statistics (SRC CLI)

Purpose Display Diameter peer statistics. These statistics include:

- Connection-related statistics—Statistics related to the connection between the server and the peer.
- Request/Answer statistics—Statistics related to Diameter Request and Diameter Answer messages between the server and the peer.
- Packet error statistics—Statistics related to Diameter errors and message receipt failures.

Action user@host> **show sic statistics diameter peer name *name***

Specify the name of the Diameter peer to display statistics; if omitted, statistics related to all Diameter peers are displayed.

Related Documentation • [Viewing Diameter Host Statistics \(SRC CLI\) on page 588](#)

PART 9

Controlling Volume Usage with the SRC VTA

- [Overview of Controlling Volume Usage with the SRC VTA on page 593](#)
- [Prerequisites for Running the SRC VTA on page 619](#)
- [Configuring the SRC VTA \(SRC CLI\) on page 639](#)
- [Managing the SRC VTA \(SRC CLI\) on page 701](#)
- [Monitoring and Testing the SRC VTA on page 705](#)

CHAPTER 36

Overview of Controlling Volume Usage with the SRC VTA

- [SRC VTA Overview on page 593](#)
- [SRC VTA Architecture and Connections to Other SRC Components on page 595](#)
- [How the SRC VTA Works on page 596](#)
- [SRC VTA Operation on page 601](#)
- [Configuring Event Handlers Overview on page 602](#)
- [Configuring Actions Overview on page 606](#)
- [Managing SRC VTA Accounts and Sessions Overview on page 613](#)
- [Adjusting the Interim Accounting Interval Overview on page 614](#)
- [Locating the SAE That Manages a Subscriber for the SRC VTA on page 615](#)
- [Using JavaScript Programs in SRC VTA Configurations on page 615](#)
- [SRC VTA SOAP Interface on page 616](#)
- [Example: Limiting Subscriber Access Based on Account Balances on page 616](#)

SRC VTA Overview

The SRC Volume-Tracking Application (SRC VTA) allows service providers to track and control the network usage of subscribers and services. You can control volume and time usage on a per-subscriber or per-service basis. This level of control means that service providers can offer tiered services that use volume as a metric, while also controlling abusive subscribers and applications.

When a subscriber or service exceeds bandwidth limits (or quotas), the SRC VTA can take actions, including imposing rate limits on traffic, sending an e-mail notification, or charging extra for additional bandwidth consumed. You can configure multiple SRC VTAs.

If you use the SRC VTA with the deep packet inspection (DPI) feature, you can control the volume of traffic for specific applications, such as peer-to-peer file sharing.

Terminology

Table 44 on page 594 defines terms that are used in the SRC VTA documentation and sample data.

For information about loading the sample data for the SRC VTA, see *Loading Sample Data into a Juniper Networks Database (SRC CLI)*.

Table 44: SRC VTA Terms

Term	Definition
Bought quota	Allowance of data volume that subscribers purchase and can transfer (upload or download) at any time. (This term is used in sample and typical SRC VTA configurations and is not inherent in the SRC VTA itself.)
Bought account	Record that details a subscriber's use of bought quota. (This term is used in sample and typical SRC VTA configurations and is not inherent in the SRC VTA itself.)
Periodic quota	Allowance of data volume that a service provider allocates to subscribers on a recurrent basis. Subscribers use this allowance to upload or download data. (This term is used in sample and typical SRC VTA configurations and is not inherent in the SRC VTA itself.)
Periodic account	Record that tracks a subscriber's use of periodic quota. (This term is used in sample and typical SRC VTA configurations and is not inherent in the SRC VTA itself.)
Quota service	Service for which the SRC VTA monitors usage. The SRC VTA activates the service for subscribers when they have a positive balance in their SRC VTA accounts, and deactivates the service when the SRC VTA account has a negative balance. (This term is used in sample and typical SRC VTA configurations and is not inherent in the SRC VTA itself.)
VTA account	Record of credit and debit entries that track a subscriber's use of a particular network resource.
VTA session	Period of activity between a subscriber and an SRC VTA.

SRC VTA Service and Subscriber Accounts

SRC VTA accounts represent the resources available to a service or a subscriber. You can configure SRC VTA accounts and then charge a particular service or subscriber's usage against the account. Each subscriber or service can have a different quota, or allowance of data volume.

You can set up the way the SRC VTA charges accounts and how account balances are updated.

You can also configure actions in response to changes in account balances. Available actions include stopping a service, starting a service, updating an account balance, sending an e-mail, and running a script. For example, if account A is emptied, the action might be to stop services X and Y, and start service Z.

The SRC VTA requires a relational database to store information about accounts. The SRC VTA installation includes sample schemas for the MySQL and Oracle databases.

SRC VTA Sessions

The SRC VTA tracks subscriber activity through VTA sessions. SRC VTA sessions do not necessarily correspond to an individual subscriber session or service session. For example, a single service session can correspond to multiple SRC VTA sessions if the service session covers multiple billing periods.

The SRC VTA can track more than just the volume and time of a service session, it can track any state of a subscriber derived from SAE plug-in events and respond to the state change.

The SRC VTA requires a relational database to store information about sessions. The SRC VTA installation includes sample schemas for the MySQL and Oracle databases.

Volume-Based Services

The SRC VTA lets you set triggers at multiple levels to provide flexible and extensive volume-based services. For example:

- When the volume remaining for the account is 300 MB, turn on the internet-256 service, turn off the internet-512 service, and send an e-mail to the subscriber.
- When the volume level reaches 100 MB, send an e-mail warning to the subscriber.
- When the volume level is 0 MB, turn on the continue-TCP-only service, turn off the internet-256 service, send an e-mail to the subscriber, and notify the accounting server.
- When the volume level is -100 MB, turn off the continue-TCP-only service, send an e-mail to subscriber, and notify the accounting server.

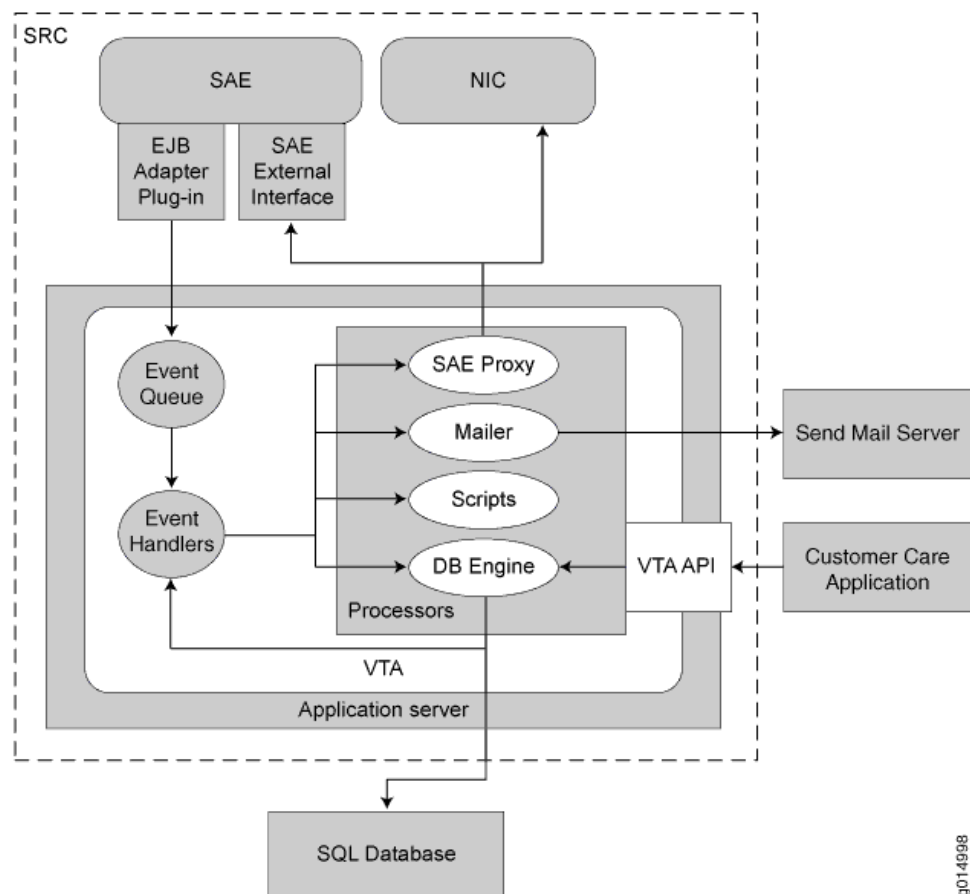
Related Documentation

- [How the SRC VTA Works on page 596](#)
- [SRC VTA Operation on page 601](#)
- [Before You Configure the SRC VTA on page 619](#)
- [SRC VTA Architecture and Connections to Other SRC Components on page 595](#)

SRC VTA Architecture and Connections to Other SRC Components

[Figure 78 on page 596](#) shows the SRC VTA architecture and the position of the SRC VTA in the SRC network.

Figure 78: SRC VTA Architecture and Position in the SRC Network



- Related Documentation**
- [SRC VTA Overview on page 593](#)
 - [How the SRC VTA Works on page 596](#)
 - [SRC VTA Operation on page 601](#)

How the SRC VTA Works

The SRC VTA manages subscriber accounts using a rule-driven event-processing system that can prioritize the actions taken for certain conditions. The SRC VTA is triggered by events, such as the logging in of subscribers, the use of network services, or the changing of account balances. These events can cause actions, such as updating account balances, starting or stopping network services, or running scripts to perform external actions.

The SRC VTA processes external events based on its configuration. The SRC VTA configuration is made up of:

- Event handlers
- Actions
- Processors

Events

Each SRC VTA event corresponds to one subscriber and contains some attributes. The SRC VTA supports the following types of events:

- Service and subscriber-tracking events from the SAE; for example, start or stop tracking events.
- Account update events triggered by updating database accounts.
- Callback events triggered by an SRC VTA API call.

Event Handlers

An event handler defines how the SRC VTA processes an event. SRC VTA event handlers consist of:

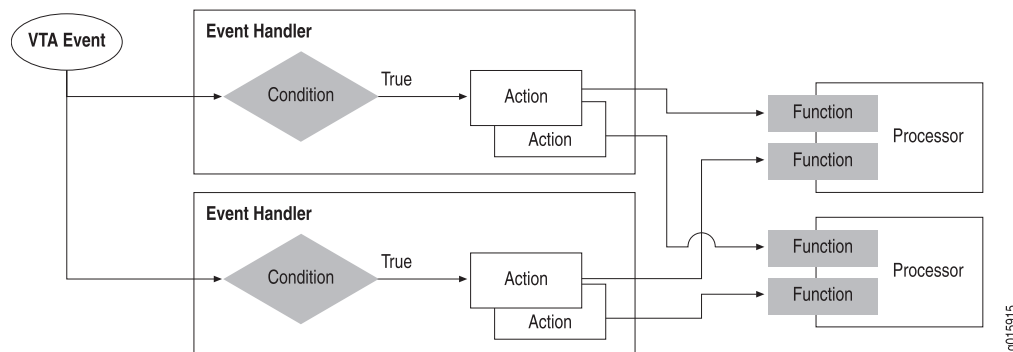
- Type of event—Tracking, update, or callback event; for example, a subscriber start-tracking event or a service start-tracking event.
- Priority—Priority at which event handlers are evaluated and executed. Event handlers are evaluated and executed from high to low priority.
- Condition—Condition that the event handler evaluates to determine whether the event handler should handle the event.
- Actions—List of actions to be performed by the event handler.

You can set up multiple event handlers to process events. For example, the first event handler could retrieve the balance for a quota account, and the next event handler could refill the quota account, depending on whether the condition of the second event handler is met.

[Figure 79 on page 598](#) shows the event handler model. The SRC VTA processes an event as follows:

1. The event handler with the highest priority receives the event, determines whether the event's type is the same as the event type of the event handler, and determines whether the event satisfies the condition of the event handler.
2. If the condition is met, the SRC VTA performs the corresponding actions based on the event attributes. An action invokes a function and provides the parameters required by that function to the processor.
3. When an event handler finishes processing an event, the next applicable event handler according to the priority of the event handler processes the event.

Figure 79: SRC VTA Event Handler Model



Actions

You specify actions that the SRC VTA takes in response to events; for example, updating an account balance, starting a service, or stopping a service. An action is modeled as a call to a function. If an event matches the type and condition requirements of the event handler, then all actions defined for the event handler process the event, one after another, in the order the actions are configured.

An action can update event attributes or add new attributes to an event for subsequent processing of the same event by another action in the same event handler, or by actions in subsequent event handlers. The updated attributes can also be used to change whether the event satisfies a subsequent event handler's condition.

An action configuration includes the following:

- Function—Function that the action invokes
- Parameter—Parameters and corresponding values to be passed to the function

Processors

Processors implement the functions that receive and process events. The SRC VTA has four processors:

- Db-engine processor—Acts as a proxy to a database
- Mailer processor—Sends e-mail notifications when certain events occur
- SAE proxy processor—Acts as a proxy to the SAE. (Requires no configuration)
- Scripts processor—Runs external scripts or JavaScript programs

Db-Engine Processor

The db-engine processor acts as a proxy to the external database. You can use the functions provided by the db-engine processor to:

- Calculate the use of network resources for a service.
- Calculate the interim accounting interval for each service based on a subscriber's remaining resources and use of the service.
- Update SRC VTA accounts with a JavaScript program.
- Terminate an SRC VTA session. This feature is usually used at the end of a billing period so that you can finish collecting data for the current billing period and start a new SRC VTA session for the new billing period. Depending on what practices you use for terminating SRC VTA sessions, you may choose to optimize the way the SRC VTA terminates sessions.

If an SRC VTA session is terminated while the SAE is in the middle of a service session, the SAE service session can split into two or more SRC VTA sessions. This process works fine if you are not frequently terminating a large number of SRC VTA sessions. However, in some cases, you may choose to terminate SRC VTA sessions on a regular basis—for example, every 24 hours. In this case, the SAE session can last a long time and can result in hundreds of SRC VTA sessions. When this occurs, processing an interim accounting event can be slow because the SRC VTA needs to find and read many records. To improve the processing time in these environments, you can optimize the SRC VTA database.

To optimize SRC VTA database operations when you are terminating large numbers of SRC VTA sessions on a regular basis, you can set the **sessions-terminated-frequently** option under the **[edit shared vta group name processor db-engine]** hierarchy level. When this option is set, the SRC VTA creates a special VTA session. Instead of finding and reading all session records, the SRC VTA needs to read from and write to only the special VTA session record.



NOTE: Setting the **sessions-terminated-frequently** option is an irreversible act. You can set this option to true, but you cannot change it back to false.

To use the **sessions-terminated-frequently** optimization feature with a set of SRC VTAs, you must follow this specific process:

1. All C Series Controllers running SRC VTAs must be running SRC software release 4.3.x or later; otherwise, you must upgrade the system.
2. Send all SAE events to only one SRC VTA.
3. Wait until the removed SRC VTAs (from Step 2) have finished processing events in their event queues.
4. Set the **sessions-terminated-frequently** option to true.
5. Wait until all SRC VTAs have reacted to the option being set. The SRC VTA information log displays the following message: "DBEngine will assume VTA sessions are terminated frequently."
6. Resume distributing SAE events to all SRC VTAs.

SAE Proxy Processor

The SAE proxy processor is a proxy to the SAE external interface that resolves the subscriber interface based on the event types to which functions are applied.

- If a function is applied to SAE subscriber-tracking or service-tracking events, the processor finds the SAE reference in the event message.

There is an exception if the **current-subscriber-only** parameter is set to false. In this case, the function finds subscribers in all SAEs with the NIC.

- If a function is applied to other events, the processor uses the subscriber's ID in the event as the key for the NIC to find the SAE reference. The SRC VTA uses the SAE reference and the **subscriber-id-solution** option that you specify under the VTA group to look up the SAE and the subscriber.

You can use the functions provided by the SAE proxy processor to:

- Set an interim interval for a service.
- Set a service session timeout for a subscription.
- Set a session timeout for a subscriber.
- Start a subscription to a service. You can specify the parameter substitutions to use when the service is started.
- Stop a subscription to a service. You can include a reason for stopping the subscription. When the service is stopped, the reason is sent to the billing system so it can differentiate between service stops.

The SAE proxy processor does not require configuration. To use the functions provided by the SAE proxy processor, you configure an action that calls the respective function for the event handler.

Mailer Processor

You can use the functions provided by the mailer processor to set up the SRC VTA to send e-mail notifications when certain events occur. You can specify that e-mail notifications be sent to subscribers, system administrators, or an automated billing system.

Scripts Processor

The scripts processor can invoke external executable scripts or JavaScript scripts. We recommend using JavaScript, where possible, for better performance.

- External scripts are executable programs, such as shell scripts, that are available on the SRC VTA's host. Each external script can perform a task and return a value. If the script returns a value, the value can be added to the current event as an event attribute.
- JavaScript programs are used to process SRC VTA event attributes and can also be used for any arbitrary purpose, just like external scripts. For example, a JavaScript program can convert an SRC VTA event attribute in a timestamp to a date string and add it to the event as a new attribute. The attribute can then be used for subsequent

actions, such as sending an e-mail notification to the subscriber. The JavaScript program can refer to any attributes of the event being processed, and it must return a value.

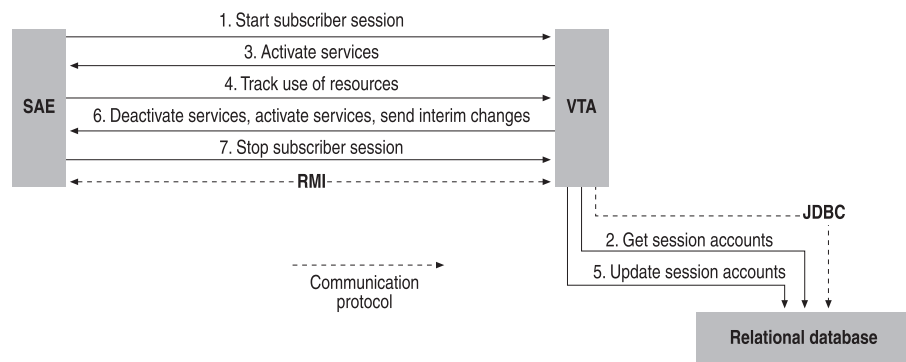
Related Documentation

- [SRC VTA Overview on page 593](#)
- [SRC VTA Operation on page 601](#)
- [SRC VTA Architecture and Connections to Other SRC Components on page 595](#)
- [Configuring Event Handlers Overview on page 602](#)
- [Configuring Actions Overview on page 606](#)
- [Configuring the DB-Engine Processor for the SRC VTA Group \(SRC-CLI\) on page 651](#)
- [Configuring the Mailer Processor for the SRC VTA Group \(SRC CLI\) on page 661](#)

SRC VTA Operation

Figure 80 on page 601 illustrates the SRC VTA operation process.

Figure 80: Operation of the SRC VTA



The SRC VTA operates as follows:

1. When an event that activates a service occurs (for example, a subscriber logs in), the SAE sends a session start event to the SRC VTA through the ejb- adaptor plug-in you configure on the SAE.
2. Optionally, the SRC VTA queries a database for the subscriber's current set of sessions and accounts.
3. Depending on the configuration of the SRC VTA, it may activate or deactivate services based on the subscriber's use of resources.
4. The SRC VTA tracks the subscriber's use of resources for a period of time.
5. The SRC VTA updates sessions and account balances in the database.

6. The SRC VTA sends to the SAE changes in the interim accounting interval and takes action to limit excessive bandwidth use or to allow increased bandwidth use.
7. When an event that deactivates a service occurs (for example, the subscriber logs out), the SRC VTA updates the VTA session, closes the VTA session, and may update the account balances.

Events received through the SRC VTA API can also cause the SRC VTA to activate services, deactivate services, or change the accounting interval.

**Related
Documentation**

- [SRC VTA Overview on page 593](#)
- [How the SRC VTA Works on page 596](#)
- [SRC VTA Architecture and Connections to Other SRC Components on page 595](#)

Configuring Event Handlers Overview

Event handlers define how the SRC VTA processes events. An event handler configuration includes the following options:

- **Events**—Type of event including tracking, update, or callback event; for example, a service-start tracking event.
- **Priority**—Priority at which event handlers are evaluated and executed. Event handlers are evaluated and executed from low to high.
- **Condition**—Condition that the event handler evaluates to determine whether the event handler should process the event.
- **Actions**—List of actions to be performed by the event handler.

You can set up multiple event handlers to process events. For example, the first event handler could retrieve the balance for a quota account, and the next event handler could refill the quota account depending on whether the condition of the second event handler is met.

When an event is received, the corresponding event type and condition configured for the event handler are evaluated based on the priority. When a condition is met, the corresponding actions are performed according to the event attributes. The action can update or add event attributes to the events for subsequent processing of the same event. An action can invoke a function provided by any processor. After an event handler has completed its processing, event processing continues with the next applicable event handler.

The SRC VTA communicates with the SAE through the Enterprise JavaBean (EJB) adapter plug-in. When an event is sent by the ejb-adapter plug-in on the SAE, it is received by the SAEEventListener, which places the event in the event queue. Because you can have multiple VTA instances, each SRC VTA has a separate SAEEventListener. The SRC VTA takes events one by one and presents them to an ordered sequence of event handlers. Each event handler includes a list of actions. If the event handler is configured to handle the event, it acts on the event based on the configured actions—for example, updating a balance or starting a service. Each action can specify a function.

The ejb-adapter plug-in can use either the round-robin algorithm or the primary/backup algorithm to send events from the SAE to the SRC VTAs. When you configure the ejb-adapter plug-in, you specify a comma-separated list of SRC VTA IP addresses or hostnames. When you use the round-robin algorithm, an outgoing event from the SAE is sent to each of the SRC VTAs specified in the comma-separated list until it is handled by one of the SRC VTAs. When you use the primary/backup algorithm, the first SRC VTA in the comma-separated list is the primary SRC VTA; all others are secondary SRC VTAs. All outgoing events from the SAE are forwarded to the primary SRC VTA until it becomes full or inaccessible. When the handling of an event fails in the primary SRC VTA, the event is sent to the secondary VTAs in a round-robin fashion until one of the secondary SRC VTAs handles the event. On successful handling of the event, the secondary SRC VTA is promoted and becomes the new primary SRC VTA.

Priority

When you configure an event handler, you need to specify the priority for evaluating and running the event handler. The priority is an integer where the smaller number has the higher priority. The event handler with the highest priority receives the event, determines whether the event's type is the same as the event type of the event handler, and determines whether the event satisfies the condition of the event handler. When an event handler finishes processing an event, the next applicable event handler according to the priority of the event handler processes the event.

Events

Each SRC VTA event corresponds to one subscriber and contains some attributes. The SRC VTA supports the following event types:

- Service and subscriber tracking events from the SAE; for example, start or stop tracking events.
- Account update events triggered by updating database accounts.
- Callback events triggered by an SRC VTA API call.

Various events can be received by the SRC VTA from the SAE. [Table 45 on page 603](#) describes the events supported by the SRC VTA.

Table 45: SRC VTA Event Types

Event	Description
account-update	Database update event
callback: <i>callid</i>	External callback event for the specified call
service-interim: <i>service name</i>	Service interim-tracking event for the specified service
service-start: <i>service name</i>	Service start-tracking event for the specified service
service-stop: <i>service name</i>	Service stop-tracking event for the specified service
user-interim	User interim-tracking event

Table 45: SRC VTA Event Types (*continued*)

user-start	User start—tracking event
user-stop	User stop—tracking event

The *service-name* is the name of any configured SRC service.

To process an event, the event handler must be configured to handle the particular event type. You configure which events are handled by the event handler with the **shared vta group name event-handler event-handler-name events events** statement. Separate events with a comma. You can configure an event handler to handle multiple event types. If an event handler is configured to handle an event, it can pass that event to a sequence of actions.

An example of an event is:

service-start: *QuotaInternet*, callback: *TerminateSession*

Event Attributes

Each event carries attributes. [Table 46 on page 604](#) describes the types of attributes that are available for each type of event.

Table 46: Event Attributes

Event Type	Available Attributes
Service and subscriber tracking events from the SAE	<ul style="list-style-type: none"> Plug-in attributes, such as PA_SERVICE_NAME or PA_LOGIN_NAME, associated with an SAE plug-in event. For a list of SAE plug-in events, see "Testing the SRC VTA Configuration Overview" on page 710. currentTime attribute—Time since January 1, 1970 UTC when the SRC VTA begins passing the event to the event handlers (events are queued in the event queue and then passed to the event handlers). The value is the number of milliseconds in the range 0–9223372036854775807. subscriberId—Subscriber ID based on attributes of a service or a subscriber-tracking event. The subscriberId event attribute is a result of the calculation. It identifies the subscriber of the corresponding SRC VTA event.
Account update events	<ul style="list-style-type: none"> old_status_accountname—The old status of the account. new_status_accountname—Returns the new status of the specified account. old_lastUpdateTime_accountname—Returns the old last update time of the account. new_lastUpdateTime_accountname—Returns the new last update time of the account. old_balance_accountname—Returns the old balance of the account. new_balance_accountname—Returns the new balance of the specified account. currentTime—Time since January 1, 1970 UTC, when the SRC VTA begins passing the event to the event handlers (events are queued in the event queue and then passed to the event handlers). The value is the number of milliseconds in the range 0–9223372036854775807. subscriberId—Subscriber ID based on attributes of a service or a subscriber-tracking event. The subscriberId event attribute is a result of the calculation. It identifies the subscriber of the corresponding SRC VTA event.

Table 46: Event Attributes (*continued*)

Event Type	Available Attributes
Callback events	<ul style="list-style-type: none"> • <code>callId</code>—When the callback event type is invoked the <code>callId</code> value is specified. The <code>callId</code> value is placed in to the event using this event attribute and then processed. If an event handler is configured to handle this event type, the actions specified for the event handler can invoke functions, which have access to the <code>callId</code> in this event attribute. • <code>currentTime</code>—Time since January 1, 1970 UTC, when the SRC VTA begins passing the event to the event handlers (events are queued in the event queue and then passed to the event handlers). The value is the number of milliseconds in the range 0–9223372036854775807. • <code>subscriberId</code>—Subscriber ID based on attributes of a service or a subscriber-tracking event. The <code>subscriberId</code> event attribute is a result of the calculation. It identifies the subscriber of the corresponding SRC VTA event.

Condition

Each event handler evaluates conditions to determine whether it should handle the event. You specify the condition as a script written in the JavaScript programming language that must return one of the following values:

- True—Event handler should handle the event.
- False—Event handler should not handle the event.

If the condition is met, the SRC VTA performs the corresponding actions based on the event attributes. An action invokes a function and provides the parameters required by that function to the db-engine processor. If no condition is specified, true is returned as the value. If a referenced attribute does not exist in the event, the referenced attribute's value is null. Following is an example condition:

```
var newBalance=<balance_BoughtQuota>+<balance_PeriodicQuota>;
if(<old_balance_PeriodicQuota>==null)
  <old_balance_PeriodicQuota>=<balance_PeriodicQuota>;
if(<old_balance_BoughtQuota>==null)
  <old_balance_BoughtQuota>=<balance_BoughtQuota>;
return <old_balance_PeriodicQuota>+<old_balance_BoughtQuota><=0&&newBalance>0;
```



NOTE: In any condition or other JavaScript script, event attributes are referred to by enclosing them in angle < > brackets.

Actions

When you configure an event handler, you specify actions that the event handler executes in response to an event; for example, updating an account balance, starting a service, or stopping a service. An action is performed only if the event matches the specified event type and condition you configure for the event handler. An action can invoke functions provided by any processor.



NOTE: We recommend that when you configure event handlers and their actions, you ensure that for any given event, all database operations are performed before any other operations that have permanent effects. This is because if a database error occurs—for example, due to normal contention for database records between different event threads—the SRC VTA rolls back the current database transaction (no changes are made to the database) and then restarts processing the event. If the event performs some other operation other than database operations before such an error, such as start a service, then that other operation is performed again when the event is reprocessed following the error.

**Related
Documentation**

- [Testing the SRC VTA Configuration Overview on page 710](#)
- [How the SRC VTA Works on page 596](#)
- [Configuring Event Handlers \(SRC CLI\) on page 648](#)
- [Configuring Actions Overview on page 606](#)
- [Configuring Actions for SRC VTA Event Handlers \(SRC CLI\) on page 647](#)

Configuring Actions Overview

Actions are executed by event handlers in response to an event. For example, the action may update an account balance, start a service, or stop a service. You configure what action the event handler takes in response to an event when you configure an event handler. An action is performed only if an event matches the event type and condition specified in the event handler configuration. An action can invoke functions provided by any processor.

An action can update event attributes or add new attributes to an event for subsequent processing of the same event by subsequent actions in the same event handler, or in subsequent event handlers.

You define an action with the `edit shared vta group name action action-name` configuration statement. You refer to a previously defined action when you configure an event handler. Event handlers can refer to multiple actions. To configure an event handler to use a particular action, specify it as follows:

```
[edit shared vta group name event-handler name]  
user@host# set actions actions
```



NOTE: We recommend that when you configure event handlers and their actions, you ensure that for any given event, all database operations are performed before any other operations that have permanent effects. This is because if a database error occurs—for example, due to normal contention for database records between different event threads—the SRC VTA rolls back the current database transaction (no changes are made to the database) and then restarts processing the event. If the event performs some other operation other than database operations before such an error, such as start a service, then that other operation is performed again when the event is reprocessed following the error.

When you configure an action, you specify the following options:

- **Function**—The function that the action invokes
- **Parameter**—The parameters and corresponding values to be passed to the function
- **On-error**—What the event handler does in response to an error

Functions are variables that an action invokes, for example, to update database accounts. The function takes as input the configured parameters, the event, and the event context (which may include values created by previously executed actions). The SRC VTA provides a set of available functions. Each function takes a different set of input parameters.

[Table 47 on page 607](#) describes the available SRC VTA functions and their associated input parameters.

Table 47: SRC VTA Functions and Input Parameters

Function Name	Description	Input Parameters
db-engine-calculate-interim	<p>Calculates the interim interval in the service-tracking event by using the interim interval function for the service, as defined with the shared vta group name processor db-engine service name interim-interval-function configuration statement. This function adds the following attribute to the event after the function is executed:</p> <ul style="list-style-type: none"> • <i>interimInterval</i>—Interim interval of the service 	None

Table 47: SRC VTA Functions and Input Parameters (*continued*)

db-engine-calculate-usage	<p>Calculates usage in the service-tracking event by using the usage metric function for the service, as defined with the shared vta group name processor db-engine service name usage-metric-function configuration statement. This function adds the following attributes to the event after the function is executed:</p> <ul style="list-style-type: none"> • <i>currentUsage</i>—Usage since the previous usage report • <i>interimTime</i>—Session length since the previous usage report 	None
db-engine-get-accounts	<p>Retrieves account data for the corresponding subscriber for the event. Data for multiple accounts is retrieved; accounts are identified by the <i>accountName</i> suffix. Accounts are defined with the shared vta group name processor db-engine account name configuration statement, where the account name corresponds to the <i>_accountName</i> suffix. Subsequent event handlers of the event can use the retrieved data. This function adds the following attributes to the event after the function is executed:</p> <ul style="list-style-type: none"> • <i>balance_accountName</i>—Balance for the account. • <i>lastUpdateTime_accountName</i>—Last update time in milliseconds since January 1, 1970 UTC for the account. • <i>status_accountName</i>—Status of the account. 	None
db-engine-terminate-session	<p>Closes active SRC VTA sessions (for the subscriber for which the event occurred) that have a status of Start or Interim. It does not stop the corresponding services in the SAE. You can use this function to stop a session at the end of a billing period. Usage data collected after the SRC VTA session is stopped is stored in new SRC VTA session records.</p>	None

Table 47: SRC VTA Functions and Input Parameters (*continued*)

db-engine-update-accounts	<p>Runs an account update script that changes the account balances of the corresponding subscriber for the event. Scripts are defined with the shared vta group name processor db-engine account-update-script name configuration statement. The predefined script name is used as input to this function. This function adds the following attributes to the event after the function is executed:</p> <ul style="list-style-type: none"> • <i>balance_accountName</i>—Balance for the account after it is updated. • <i>lastUpdateTime_accountName</i>—Last update time in milliseconds since January 1, 1970 UTC for the account after it is updated. • <i>status_accountName</i>—Status of the account after it is updated. 	<p>script-name—Name of the account update script defined in the db-engine processor.</p>
mailer-send	<p>Sends e-mail notifications when certain events occur. You can specify that e-mail notifications be sent to anyone, for example, subscribers, system administrators, or an automated billing system.</p>	<ul style="list-style-type: none"> • recipient—Destination e-mail address • from—Source e-mail address • subject—Subject of the e-mail • text—Content of the e-mail
sae-set-interim-interval	<p>Sets the interim accounting interval for the service session identified in the event to be the value of the <i>interimInterval</i> attribute currently found in the event. Before this function is called, the db-engine-calculate-interim function of the db-engine processor must be called in order to place the <i>interimInterval</i> attribute into the event.</p>	<p>current-subscriber-only—Specifies whether the function is applied only to the subscriber identified in the event or to all subscribers who have the same subscriber ID.</p> <ul style="list-style-type: none"> • Set to true to apply the function to the current subscriber only. If you do not set this parameter, true is the default behavior. • Set to false to apply the function to all subscribers who have the same subscriber ID.

Table 47: SRC VTA Functions and Input Parameters (*continued*)

sae-set-service-timeout	Sets the service session timeout for the service session identified in the event.	<ul style="list-style-type: none"> • subscription-name—Name of the subscription in the format <i>serviceName%subscriptionId</i>. Default subscriptions have the same name as the service. This parameter is optional when a service-tracking event is being processed. If <i>%subscriptionId</i> is omitted, the default subscription is assumed. • session-name—Name of the service session. This parameter is ignored if a service-tracking event is being processed and the subscription name is omitted. In this case, the session name from the service-tracking event is used. If this parameter is omitted, the default service session is used. • session-timeout—Length of the service session timeout in seconds. If the session timeout is set to 0, the service session is stopped immediately. When the session timeout expires, the service session is stopped. • current-subscriber-only—Specifies whether the function is applied only to the subscriber identified in the event or to all subscribers who have the same subscriber ID. <ul style="list-style-type: none"> • Set to true to apply the function to the current subscriber only. If you do not set this parameter, true is the default behavior. • Set to false to apply the function to all subscribers who have the same subscriber ID.
sae-set-user-timeout	Sets the subscriber session timeout of the subscriber identified in the event.	<ul style="list-style-type: none"> • session-timeout—Length of the subscriber session timeout in seconds. If the session timeout is set to 0, the subscriber session is stopped immediately. When the session timeout expires, the subscriber is logged out. • current-subscriber-only—Specifies whether the function is applied only to the subscriber identified in the event or to all subscribers who have the same subscriber ID. <ul style="list-style-type: none"> • Set to true to apply the function to the current subscriber only. If you do not set this parameter, true is the default behavior. • Set to false to apply the function to all subscribers who have the same subscriber ID.

Table 47: SRC VTA Functions and Input Parameters (*continued*)

sae-start-service	Starts the specified service subscription.	<ul style="list-style-type: none"> • subscription-name—Name of the subscription in the format <i>serviceName%subscriptionId</i>. Default subscriptions have the same name as the service. This parameter is optional when a service-tracking event is being processed. If <i>%subscriptionId</i> is omitted, the default subscription is assumed. • session-timeout—Length of the service session timeout in seconds. When the session timeout expires, the service session is stopped. • session-name—Name of the service session. If this parameter is omitted, the default service session is used. • current-subscriber-only—Specifies whether the function is applied only to the subscriber identified in the event or to all subscribers who have the same subscriber ID. <ul style="list-style-type: none"> • Set to true to apply the function to the current subscriber only. If you do not set this parameter, true is the default behavior. • Set to false to apply the function to all subscribers who have the same subscriber ID. • substitution—Policy parameter substitution to use when starting the service. This is specified as substitution name value value. If this parameter is omitted, the service is started without substitutions. • persistent—Specifies whether a service session is persistent. If you use the SRC VTA to activate a service with the persistent option, this service is subsequently activated by the SAE every time the subscriber connects to the network. This option provides efficiency because the SRC VTA does not need to make a decision to activate the service on subsequent logins and because some applications can more efficiently activate a group of services at login. <ul style="list-style-type: none"> • Set to true to cause the session to be persistent. • Set to false to specify that the session is not persistent and the service is activated or deactivated only for the current subscriber session.
-------------------	--	--

Table 47: SRC VTA Functions and Input Parameters (*continued*)

sae-stop-service	Stops the specified subscription to the specified service.	<ul style="list-style-type: none"> • subscription-name—Name of the subscription in the format <i>serviceName%subscriptionId</i>. Default subscriptions have the same name as the service. This parameter is optional when a service-tracking event is being processed. If <i>%subscriptionId</i> is omitted, the default subscription is assumed. • session-name—Name of the service session. If the <i>subscriptionName</i> parameter is omitted, this parameter is ignored. If this parameter is omitted, the default service session is used. • reason—Reason for the termination. When the service is stopped, the termination cause can be sent to the billing system so it can differentiate between service stops. If this parameter is omitted, no termination cause is provided to the billing system. Specify an integer that identifies the termination cause. Possible values are defined in RFC 2866—RADIUS Accounting (June 2000). • current-subscriber-only—Specifies whether the function is applied only to the subscriber identified in the event or to all subscribers who have the same subscriber ID. <ul style="list-style-type: none"> • Set to true to apply the function to the current subscriber only. If you do not set this parameter, true is the default behavior. • Set to false to apply the function to all subscribers who have the same subscriber ID. • persistent—Specifies whether a service session is persistent. <ul style="list-style-type: none"> • Set to true to cause the session to be persistent. • Set to false to specify that the session is not persistent and the service is deactivated only for the subscriber identified in the event.
scripts-run-external-script	Executes an external script that is present on the local SRC file system, as previously defined with the shared vta group name processor scripts external-script name configuration statement.	<ul style="list-style-type: none"> • script-name—Name of an external script previously defined with the shared vta group name processor scripts external-script name configuration statement. • name—Any other parameters the script expects to receive as previously defined under the shared vta group name processor scripts external-script name parameters configuration statement. These parameters are entered as name/value pairs.

Table 47: SRC VTA Functions and Input Parameters (*continued*)

scripts-run-javascript	<p>Executes a script written in the JavaScript programming language that you previously defined by using the shared vta group name processor scripts javascript name configuration statement.</p> <ul style="list-style-type: none"> • script-name—Name of a script written in the JavaScript programming language that you previously defined by using the shared vta group name processor scripts javascript name configuration statement. <p>Any parameters referenced in the script itself are taken from the event and passed to the function automatically.</p>
------------------------	---

On Error

For each action, you must specify what the event handler does if an error occurs; you specify this with the **on-error** option. If an error occurs, the event handler can do one of the following:

- **Abort-event-processing**—Stop processing the current event.
- **Go-to-next-action**—Continue with the next action, if any, in the same event handler.
- **Go-to-next-event-handler**—Skip any remaining actions in the current event handler and proceed to the next event handler (if any).

Related Documentation

- [How the SRC VTA Works on page 596](#)
- [Configuring Event Handlers Overview on page 602](#)
- [Configuring Actions for SRC VTA Event Handlers \(SRC CLI\) on page 647](#)
- [Configuring Event Handlers \(SRC CLI\) on page 648](#)

Managing SRC VTA Accounts and Sessions Overview

The SRC VTA allows service providers to manage accounts and sessions by:

- [Identifying Subscribers, SAEs, and Sessions on page 613](#)
- [Managing SRC VTA Accounts and Sessions on page 614](#)

Identifying Subscribers, SAEs, and Sessions

The SRC VTA must be able to identify each subscriber by a unique identifier. The SRC VTA uses the identifier to manage:

- SRC VTA accounts and sessions
- Subscriber and service sessions

You can configure the SRC VTA to use data keys to identify corresponding data values for these management tasks. The data keys depend on the subscriber's identifier and comprise one or more plug-in attributes. Some identifiers are suitable for residential subscribers and some for enterprise subscribers.

Managing SRC VTA Accounts and Sessions

Depending on the information that identifies subscribers in your SRC configuration, you can configure the SRC VTA to use several types of plug-in attributes as data keys to identify accounts and sessions in the SRC VTA database. If you use a NIC with the SRC VTA, the SRC VTA can also use some of these plug-in attributes to construct a data key that the NIC can use to determine which SAE manages a subscriber. When the NIC identifies an SAE, the SRC VTA can also obtain a key to identify the subscriber session that the SAE is managing for the subscriber.

You configure the data keys that identify accounts and sessions with the **shared vta group name** statement and specifying the data keys by setting the **subscriber-id-solution** option.

Related Documentation

- [SRC VTA Overview on page 593](#)
- [Configuring a Database to Store Account and Session Data \(SRC CLI\) on page 633](#)
- [Configuring the Initial Balance and Status of a Subscriber Account in the External Database \(SRC CLI\) on page 652](#)
- [Example: Limiting Subscriber Access Based on Account Balances on page 616](#)

Adjusting the Interim Accounting Interval Overview

When you configure services in the SRC VTA, you can optionally define a formula to dynamically adjust the interim accounting interval for each service based on the subscriber's remaining resources and use of the network for that service. Each service in the SRC VTA can use a different formula. You can configure the SRC VTA software to evaluate the formula to obtain the accounting intervals. Depending on the result, the SRC VTA performs the following functions:

- If the result is zero, the SRC VTA disables interim accounting.
- If the result is a negative number, the SRC VTA does not change the interim accounting interval.
- If the result is a positive number, the SRC VTA changes the interim accounting interval to this value.

The variables used to define the interim accounting interval are categorized as:

- Current service—Provides session data of the service for the current service-tracking event.
- Other service—Provides service session usage information for another subscriber service for the current service-tracking event. For example, if a subscriber has two quota services, QuotaLocal and QuotaInternet, the interim formula for QuotaLocal can provide usage information to QuotaInternet.
- Account balance—Provides the balance in the account.

For details on the variables used to define the interim accounting interval formula, see [“Variables Used to Define the Interim Accounting Interval for Services” on page 656](#).

- Related Documentation**
- [Configuring SRC VTA Services and Policies on page 635](#)
 - [Variables Used to Define the Interim Accounting Interval for Services on page 656](#)
 - [Configuring the Interim Account Interval and Usage Metric of a Service in the External Database \(SRC CLI\) on page 654](#)

Locating the SAE That Manages a Subscriber for the SRC VTA

You can use NIC proxies if the SRC VTA software needs to locate the SAE that manages a particular subscriber. For example, if the SRC VTA receives an account update event and determines that it needs to reconfigure the corresponding SAE session, the SRC VTA must find the SAE that is managing the session. The SRC VTA can do this through the NIC.

You can also use the NIC with the SRC VTA to allow the following:

- Immediately activate subscriptions to quota services—The SRC VTA immediately activates a subscriber's quota service when a deposit is made to the subscriber's account. In this case, the NIC maps the subscriber's identifier to the SAE reference. This scenario is for subscribers who connect to the network through routers running JunosE or Junos OS.

If you do not set up a NIC for this purpose or you use an identifier that the NIC cannot map to an SAE reference, subscribers must log out and log in again before the SRC VTA can activate their quota services when deposits are made to their accounts.

- Related Documentation**
- [Identifying Subscribers, SAEs, and Sessions on page 613](#)
 - [Configuring Subscribers and Subscriptions to SRC VTA Services on page 635](#)
 - [Configuring a NIC for the SRC VTA \(SRC CLI\) on page 636](#)
 - [Configuring NIC Proxies for the SRC VTA on page 636](#)

Using JavaScript Programs in SRC VTA Configurations

You can use JavaScript programs in your SRC VTA configuration for such tasks as calculating a usage metric or an interim accounting interval, specifying an event condition, updating event attributes in processors, and writing scripts to update accounts.

You can reference a specific set of predefined variables in a JavaScript program, such as plug-in attributes, event attributes, or account balances.

When a variable is referenced or updated in the JavaScript program, enclose it in angle brackets (<>) so that the JavaScript program retrieves only the necessary information. Within the JavaScript program, only one instance of the referenced variable must be enclosed in angle brackets.

You define formulas in the JavaScript scripting language (see <http://wp.netscape.com/eng/mozilla/3.0/handbook/javascript/index.html>).

- Related Documentation**
- [How the SRC VTA Works on page 596](#)
 - [Configuring JavaScript Programs on page 663](#)

SRC VTA SOAP Interface

The SRC Volume Tracking Application (SRC VTA) API is a Simple Object Access Protocol (SOAP) interface that allows developers to create gateway clients and that administrators use to manage SRC VTA subscribers and sessions. The SRC Web Services Gateway allows a gateway client—an application that is not part of the SRC network—to interact with SRC components, such as the SRC VTA, through a SOAP interface.

SRC VTA SOAP Interface URL

After you have installed the SRC VTA, you can access a Web Services Definition Language (WSDL) file for the application. The WSDL file defines the SOAP properties that you or your customers can use to develop a client. The URL for the SRC VTA WSDL file is:

`http://<host-name>:8080:/<vta-group>/api/soap/services/vtaAPI?wsdl`

- **<host-name>**—Name of the C Series Controller that is running the SRC VTA
- **<vta-group>**—Name of the SRC VTA group

- Related Documentation**
- [Enabling the SOAP Interface for an SRC VTA Group \(SRC CLI\) on page 676](#)
 - [Methods for the SRC Volume Tracking Application SOAP Interface on page 677](#)
 - [Viewing SRC VTA SOAP API Statistics \(SRC CLI\) on page 709](#)

Example: Limiting Subscriber Access Based on Account Balances

The sample data provides an example called Quota that limits a subscriber's access rate based on the balances of accounts that record the subscriber's use of network resources. Subscribers receive a quota of transfer (upload and download) volume in two ways:

- **Periodic quota**—Volume that is periodically added to a subscriber's account. For example, a subscriber may receive a 25-MB periodic quota each month. The periodic quota is tracked in the periodic account.
- **Bought quota**—Additional volume that a subscriber can purchase and use at any time. For example, a subscriber may purchase 25 MB of bought quota in January, and use the bought quota between January and March. Bought quota is tracked in a bought account.

As a subscriber consumes volume, the SRC VTA debits the accounts, using first the periodic quota and, if no periodic quota is available, the bought quota.

Subscribers managed by the SRC VTA require a subscription to quota services—services for which the SRC VTA monitors and manages usage. You must configure these subscriptions to be activated when the subscriber logs in. When a subscriber logs in to

the SAE, the SAE tries to activate the quota services. However, if neither the periodic account nor the bought account has a positive balance, the SRC VTA deactivates the quota services, and the SAE applies to the subscriber either the default policy or another policy that implements a service with a lower bandwidth.

The units of the accounts depend on the formula that you define to determine the use of network resources. With this formula, the SRC VTA calculates the change to the accounts and the interim accounting interval.

The Quota configuration example provides an SRC VTA that operates as follows:

1. When a service session for the quota service starts, the SAE sends a start event to the SRC VTA.
2. The SRC VTA starts a session that has the same identifier as the service session and a qualifier of zero.
3. When the SRC VTA receives the first interim update from the SAE in the SRC VTA session, it records a balance change that details the use of resources and the event time for the SRC VTA session. When the SRC VTA receives interim updates in the SRC VTA session, it updates the use of resources and the event time in the balance change recorded previously.
 - If the periodic account contains sufficient resources to cover the balance change, the SRC VTA changes only the balance of that account.
 - If the periodic account does not contain sufficient resources for the change, the SRC VTA records one balance change for the resources available in that account and records another balance change for the difference in the bought account. In this case, the SRC VTA records subsequent balance changes to the bought account.
 - If neither account has sufficient resources, the SRC VTA deactivates the quota service.
4. If the SAE session ends, the SRC VTA session ends. When a new service session starts, Steps 1 to 3 recur. However, a service session may last for several SRC VTA sessions. In this case, the SAE and SRC VTA continue the process described in Step 3.
5. When an administrator replenishes the periodic quota, the SRC VTA ends the session, finalizes all balance changes for the session, and records a credit to the periodic account.
6. When the subscriber buys additional volume, the SRC VTA ends the session, finalizes all balance changes for the session, and records a credit to the bought account.
7. When the SRC VTA next receives an interim update event from the SAE, it starts a new session. The SRC VTA obtains the start time for the session from the SAE event, and records debits to the accounts as described in Step 3.

The SRC VTA always ends the session when an administrator replenishes periodic quota or the subscriber buys volume. However, a service session may last for several billing periods. In this case, when the SRC VTA starts a new session, it continues to assign the SAE session identifier to the session and increments the qualifier by one. Keeping the session within a billing period allows the SRC VTA to finalize balance changes.

**Related
Documentation**

- *Loading Sample Data into a Juniper Networks Database (SRC CLI)*
- [SRC VTA Overview on page 593](#)
- [Identifying Subscribers, SAEs, and Sessions on page 613](#)
- [Managing SRC VTA Accounts and Sessions on page 614](#)

Prerequisites for Running the SRC VTA

- [Before You Configure the SRC VTA on page 619](#)
- [Configuring the Web Application Server \(SRC CLI\) on page 621](#)
- [Configuration Statements for the Web Application Server on page 621](#)
- [Configuring Local Properties for the Web Application Server \(SRC CLI\) on page 622](#)
- [Configuring the Web Application Server Shared Cluster Configuration \(SRC CLI\) on page 623](#)
- [Configuring the Nodes in the Web Application Server Cluster \(SRC CLI\) on page 624](#)
- [Configuring Remote Access to the Application Server \(SRC CLI\) on page 625](#)
- [Configuring Virtual Hosts for the Web Applications \(SRC CLI\) on page 626](#)
- [Configuring User Accounts for Web Applications \(SRC CLI\) on page 627](#)
- [Installing Web Applications in the SRC Web Application Server on page 629](#)
- [Starting the Web Application Server on a C Series Controller on page 629](#)
- [Configuring the SAE to Send Tracking Events to the SRC VTA \(SRC CLI\) on page 630](#)
- [Specifying Tracking Plug-Ins for Enterprise Subscribers on Junos OS Routing Platforms on page 632](#)
- [Configuring the External Database on page 633](#)
- [Troubleshooting Database Deadlocks on page 634](#)
- [Configuring SRC VTA Services and Policies on page 635](#)
- [Configuring Subscribers and Subscriptions to SRC VTA Services on page 635](#)
- [Configuring a NIC for the SRC VTA \(SRC CLI\) on page 636](#)
- [Configuring NIC Proxies for the SRC VTA on page 636](#)

Before You Configure the SRC VTA

Because the SRC VTA relies on other components in the SRC network, you must complete several tasks before you configure the SRC VTA.

Before you configure the SRC, you must complete the following tasks:

1. Deploy a working SRC network.

To support the SRC VTA, you must install SAEs to manage the routers or other devices through which subscribers connect to the network.

See *Configuring the SAE (SRC CLI)*.

2. Configure the SRC Web application server.

See *“Configuring the Web Application Server (SRC CLI)” on page 621*.

3. (Optional) Configure a NIC that identifies the SAE reference for each subscriber type. You need to complete this task only if your configuration requires a NIC—for example, if an event handler action affects more than just the current subscriber described by an event.

See *“Configuring a NIC for the SRC VTA (SRC CLI)” on page 636*.

4. (Optional) Create the Java scripts that the SRC VTA invokes.

5. On a separate host, install a relational database to store the data that the SRC VTA tracks. You must have the database server running, and then you must create the SRC VTA database within your database server.

See, *“Configuring a Database to Store Account and Session Data (SRC CLI)” on page 633*.

6. To allow the SRC Web application server to connect to your external database, copy the JDBC driver .jar file for your database server brand and version to the `/opt/UMC/appsvr/common/lib` directory on every system running the SRC VTA.

See *“Installing the JDBC Driver .jar File” on page 634*.

7. Configure the associated services and policies.

See *“Configuring SRC VTA Services and Policies” on page 635*.

8. Configure your subscribers and subscriptions.

See *“Configuring Subscribers and Subscriptions to SRC VTA Services” on page 635*.

9. Configure the SAE to send tracking events to the SRC VTA.

See *“Configuring the SAE to Send Tracking Events to the SRC VTA (SRC CLI)” on page 630*.

**Related
Documentation**

- *SRC VTA Overview on page 593*
- *How the SRC VTA Works on page 596*
- *SRC VTA Operation on page 601*
- *Web Application Server on C Series Controllers Overview*
- *Configuration Statements for the Web Application Server on page 621*

Configuring the Web Application Server (SRC CLI)

Tasks to configure the Web application server are:

1. Configure the Web application server shared cluster configuration.
See [“Configuring the Web Application Server Shared Cluster Configuration \(SRC CLI\)” on page 623.](#)
2. Configure the operating properties.
See [“Configuring Local Properties for the Web Application Server \(SRC CLI\)” on page 622.](#)
3. Configure the nodes of the Web application server cluster.
See [“Configuring the Nodes in the Web Application Server Cluster \(SRC CLI\)” on page 624.](#)
4. Configure remote access to the application server.
See [“Configuring Remote Access to the Application Server \(SRC CLI\)” on page 625.](#)
5. Configure the virtual host for the Web application, including whether to allow or deny access by specific remote clients.
See [“Configuring Virtual Hosts for the Web Applications \(SRC CLI\)” on page 626.](#)
6. Configure the user accounts for the Web application.
See [“Configuring User Accounts for Web Applications \(SRC CLI\)” on page 627.](#)

Related Documentation

- [Web Application Server on C Series Controllers Overview](#)
- [Configuring the Web Application Server Shared Cluster Configuration \(SRC CLI\) on page 623](#)
- [Configuring the Nodes in the Web Application Server Cluster \(SRC CLI\) on page 624](#)

Configuration Statements for the Web Application Server

Use the following configuration statements to configure the operating properties for the Web application server at the **[edit]** hierarchy level.

```
slot number application-server {
    java-garbage-collection-options java-garbage-collection-options;
    java-heap-size java-heap-size;
    shared-cluster shared-cluster
}

shared application-server cluster name {
    channel-stack (udp|tcp);
    multicast-address multicast-address;
}

shared application-server cluster name nodes node address {
    node-id node-id;
```

```
}  
  
slot number application-server web http {  
    port port;  
    interface interface;  
}  
  
slot number application-server web https {  
    local-certificate local-certificate;  
    port port;  
    interface interface;  
}  
  
slot number application-server web virtual-host host-name {  
    alias alias;  
    allow-address allow-address;  
    allow-host allow-host;  
    deny-address deny-address;  
    deny-host deny-host;  
}  
  
shared application-server user name  
  
shared application-server user name authentication {  
    encrypted-password encrypted-password;  
    plain-text-password;  
}
```

**Related
Documentation**

- [Configuring the Web Application Server \(SRC CLI\) on page 621](#)
- [Configuring Remote Access to the Application Server \(SRC CLI\) on page 625](#)
- [Configuring Virtual Hosts for the Web Applications \(SRC CLI\) on page 626](#)
- [Configuring User Accounts for Web Applications \(SRC CLI\) on page 627](#)
- [Web Application Server on C Series Controllers Overview](#)

Configuring Local Properties for the Web Application Server (SRC CLI)

To configure basic local properties:

1. From configuration mode, access the configuration statement that configures the local properties.

 user@host# **edit slot 0 application-server**
2. Configure the garbage collection functionality of the Java Virtual Machine.

 [edit slot 0 application-server]
 user@host# **set java-garbage-collection-options *java-garbage-collection-options***
3. (Optional) If you encounter problems caused by lack of memory, change the maximum memory size available to the JRE.

 [edit slot 0 application-server]
 user@host# **set java-heap-size *java-heap-size***
4. (Optional) Configure the cluster name. Specify the shared-cluster as **/application-server/shared-cluster**.

```
[edit slot 0 application-server]
user@host# set shared-cluster /application-server/shared-cluster
```

For example, to configure a shared cluster called cluster-1:

```
[edit slot 0 application-server]
user@host# set shared-cluster /application-server/cluster-1
```



NOTE: If you change the shared cluster name, you must restart the local application server for the change to take effect.

5. (Optional) Verify your configuration.

```
[edit slot 0 application-server]
user@host# show

shared-cluster /application-server/cluster-1;
web {
  http {
    interface eth0;
    port 8080;
  }
  virtual-host eth0;
}
```

**Related
Documentation**

- [Configuring the Web Application Server Shared Cluster Configuration \(SRC CLI\) on page 623](#)
- [Configuring the Nodes in the Web Application Server Cluster \(SRC CLI\) on page 624](#)
- [Web Application Server on C Series Controllers Overview](#)

Configuring the Web Application Server Shared Cluster Configuration (SRC CLI)

Use the following statements to configure a Web application server shared cluster configuration:

```
shared application-server cluster name {
  channel-stack (udp|tcp);
  multicast-address multicast-address;
}
```

To configure the Web application server shared cluster configuration:

1. From configuration mode, access the statement that configures the shared cluster configuration. The name you specify must match the name you configured for the local configuration at the **[edit slot 0 application-server]** hierarchy level.

```
user@host# edit shared application-server cluster name
```

For example, if you have the following local configuration:

```
[edit slot 0 application-server]
shared-cluster /application-server/cluster-1
```

You need to specify cluster-1 as the cluster name for the shared configuration:

```
user@host# edit shared application-server cluster cluster-1
```

2. Configure the channel stack.

```
[edit shared application-server cluster cluster-1]
```

```
user@host# set channel-stack (udp|tcp)
```

3. (Optional) Specify the multicast address. The multicast address is required only if UDP is selected as the channel stack.

```
[edit shared application-server cluster cluster-1]
```

```
user@host# set multicast-address multicast-address
```

4. (Optional) Verify your configuration.

```
[edit shared application-server cluster cluster-1]
```

```
user@host# show
```

```
channel-stack tcp;
```

```
[edit shared application-server cluster cluster-1]
```

```
user@host#
```

Related Documentation

- [Web Application Server on C Series Controllers Overview](#)
- [Configuring the Nodes in the Web Application Server Cluster \(SRC CLI\) on page 624](#)
- [Configuring the Web Application Server \(SRC CLI\) on page 621](#)
- [Configuring Local Properties for the Web Application Server \(SRC CLI\) on page 622](#)
- [Viewing the Web Application Server Cluster Status \(SRC CLI\)](#)

Configuring the Nodes in the Web Application Server Cluster (SRC CLI)

Use the following statements to configure the nodes in the Web application server cluster:

```
shared application-server cluster name nodes node address {  
  node-id node-id;  
}
```

To configure the Web application server cluster nodes:

1. From configuration mode, access the statement that configures the cluster nodes and specify the IP address of the node.

```
user@host# shared application-server cluster name nodes node address {
```

2. Configure the node ID for the node. The node ID is a random number you assign to the node. Each node must have a unique node ID specified as the integer type.

```
[edit shared application-server cluster name nodes node address]
```

```
user@host# set node-id node-id
```

3. (Optional) Verify your configuration.

```
[edit shared application-server cluster name nodes node address]
```

```
user@host# show
```

Following is a sample output of the cluster node configuration:


```

channel-stack udp;
multicast-address 255.255.100.100;
nodes {
  node 10.1.2.3 {
    node-id 2;
  }
  node 10.1.2.4 {
    node-id 1;
  }
  node 10.1.2.5 {
    node-id 4;
  }
}

```

Related Documentation

- [Web Application Server on C Series Controllers Overview](#)
- [Configuring the Web Application Server Shared Cluster Configuration \(SRC CLI\) on page 623](#)
- [Configuring the Web Application Server \(SRC CLI\) on page 621](#)
- [Configuring Local Properties for the Web Application Server \(SRC CLI\) on page 622](#)

Configuring Remote Access to the Application Server (SRC CLI)

Before you can start using the application server, you need to configure and enable access to the application server. You can make the application server accessible through secure HTTP (HTTPS) or HTTP.

- [Configuring Access to the Application Server Through Secure HTTP on page 625](#)
- [Configuring Access to the Application Server Through HTTP on page 626](#)

Configuring Access to the Application Server Through Secure HTTP

Before you configure access to the application server through HTTPS, obtain a digital security certificate on the system.

To make the application server accessible through HTTPS:

1. From configuration mode, access the statement that configures access through HTTPS.

```
user@host# edit slot 0 application-server web https
```

2. Specify which TCP port is to receive incoming connection requests for the application server.

```
[edit slot 0 application-server web https]
user@host# set port port
```

3. Specify the interface to be used for connections to the application server.

```
[edit slot 0 application-server web https]
user@host# set interface interface
```

On a C Series Controller, use **eth1** for built-in Web applications; you can use **eth0** for demonstration applications.

4. Specify the name of the certificate on the local system.

```
[edit slot 0 application-server web https]
user@host# set local-certificate local-certificate
```

5. (Optional) Configure user accounts to allow specified clients to authenticate with the application server.

Configuring Access to the Application Server Through HTTP

To make the application server accessible through HTTP:

1. From configuration mode, access the statement that configures access through HTTP.

```
user@host# edit slot 0 application-server web http
```

2. Specify which TCP port is to receive incoming connection requests for the application server.

```
[edit slot 0 application-server web http]
user@host# set port port
```

3. Specify the interface to be used for connections to the application server.

```
[edit slot 0 application-server web http]
user@host# set interface interface
```

On a C Series Controller, use **eth1** for built-in Web applications; you can use **eth0** for demonstration applications.

4. (Optional) Configure user accounts to allow specified clients to authenticate with the application server.

Related Documentation

- [Configuring the Web Application Server \(SRC CLI\) on page 621](#)
- [Configuring User Accounts for Web Applications \(SRC CLI\) on page 627](#)
- [Web Application Server on C Series Controllers Overview](#)
- [Digital Certificates Overview](#)

Configuring Virtual Hosts for the Web Applications (SRC CLI)

Use the following configuration statements to configure virtual hosts at the **[edit]** hierarchy level:

```
slot number application-server web virtual-host host-name {
  alias [alias...];
  allow-address [allow-address...];
  allow-host [allow-host...];
  deny-address [deny-address...];
  deny-host [deny-host...];
}
```

To configure virtual hosts for the Web applications:

1. From configuration mode, access the statement that configures the virtual host.

By default, SRC applications run on the virtual host **eth0**. You must configure **eth0** as a virtual host. The hostname must be unique.

```
user@host# edit slot 0 application-server virtual-host eth0
```

2. Specify the alternate DNS names or IP addresses for the virtual host.

```
[edit slot 0 application-server virtual-host eth0]
user@host# set alias [alias ...]
```

The alias must be unique. Specify the following alias for the **eth0** virtual host:

- The IP address assigned to **eth0**.
- The name for the SRC host configured at the **[edit system host-name]** and **[edit system domain-name]** hierarchy levels.

3. Configure access to the virtual host. Specify the IP addresses for remote clients that are allowed access to the virtual host.

```
[edit slot 0 application-server virtual-host eth0]
user@host# set allow-address [allow-address...]
```

4. Configure access to the virtual host. Specify the hostnames for remote clients that are allowed access to the virtual host.

```
[edit slot 0 application-server virtual-host eth0]
user@host# set allow-host [allow-host...]
```

5. Deny access to the virtual host. Specify the IP addresses for remote clients that are denied access to the virtual host.

```
[edit slot 0 application-server virtual-host eth0]
user@host# set deny-address [deny-address...]
```

6. Deny access to the virtual host. Specify the hostnames for remote clients that are denied access to the virtual host.

```
[edit slot 0 application-server virtual-host eth0]
user@host# set deny-host [deny-host...]
```

Related Documentation

- [Configuring the Web Application Server \(SRC CLI\) on page 621](#)
- [Web Application Server on C Series Controllers Overview](#)

Configuring User Accounts for Web Applications (SRC CLI)

User accounts provide one way for clients to authenticate with the application server. For each account, you define the login name for the user, authentication information, and role. You can configure plain-text password or encrypted password as the type of authentication for user accounts. When you delete user accounts, the software verifies that the user account is not referenced by another configuration.



NOTE: Client profiles can be cached by applications for 30 minutes. If you change the password or role of a client that has been used within the last 30 minutes, it can take up to 30 minutes before these changes take effect.

If you do not want to wait 30 minutes for the changes to take effect, restart the Web application server.

Use the following configuration statements to configure user accounts at the **[edit]** hierarchy level:

```
shared application-server user name
shared application-server user name authentication {
  encrypted-password encrypted-password;
  plain-text-password;
  role [DSA | PCMM | VTA-group name;
}
```

To configure a user account:

1. From configuration mode, access the configuration statement that configures a user account and specify a username that identifies the client.

```
user@host# edit shared application-server user name
```

The username must be unique within the system. Do not include spaces, colons, or commas in the username.

2. Configure authentication for the user account.

```
[edit shared application-server user name]
user@host# set authentication (plain-text-password | encrypted-password)
```

where:

- **plain-text-password**—Prompt for a plain-text (unencrypted) password.
- **encrypted-password**—Password encoded with crypt. The format of encrypted passwords is "{crypt}<13-characters in a-zA-Z0-9./>".

We recommend that you not enter the password in encrypted format.

For example:

```
user@host# set authentication plain-text-password
New password: type password here
Retype new password: retype password here
```

3. Configure the role for the user account.

```
[edit shared application-server user name]
user@host# set role VTA-Quota
```

Set the role to one of the following values:

- **DSA**—Role for clients accessing the DSA services: dsa-service and dsa2-service
- **PCMM**—Role for clients accessing the DSA service: pcmm-service

- *VTA-group name*—Role for clients accessing the SOAP API for the SRC VTA. The CLI returns all SRC VTA groups configured under the **[edit shared vta group]** hierarchy with the prefix “VTA”. For example, set the role to VTA-Quota for clients accessing the SOAP API for the SRC VTA group called Quota.

Related Documentation

- [Configuring Remote Access to the Application Server \(SRC CLI\) on page 625](#)
- [SRC VTA SOAP Interface on page 616](#)
- [Enabling the SOAP Interface for an SRC VTA Group \(SRC CLI\) on page 676](#)
- [Methods for the SRC Volume Tracking Application SOAP Interface on page 677](#)

Installing Web Applications in the SRC Web Application Server

The SRC software includes a Web application server component for deploying Web applications for lab tests and demonstrations.

Use the following procedure to deploy Web applications in the SRC Web application server.



NOTE: You can deploy a Web application in the Web application server for lab tests and demonstrations. However, running non-SRC Web applications in production environments is not supported.

To deploy a Web application in the SRC Web application server:

1. Start the Web application server.
2. Prepare the Web application archive (WAR) file on a machine other than the C Series Controller.
3. Deploy the WAR file on the C Series Controller. The SRC Web application server automatically starts the Web application when a new WAR file is deployed.

```
user@host> request appsvr deploy file name
```

For example:

```
user@host> request appsvr deploy file ftp://host/path/ssportal.war
```

Related Documentation

- [Removing Web Applications from the Application Server](#)
- [Starting the Web Application Server on a C Series Controller on page 629](#)
- [Restarting the Web Application Server on a C Series Controller](#)

Starting the Web Application Server on a C Series Controller

To start the Web application server on a C Series Controller:

```
user@host> enable component appsvr
```

- Related Documentation**
- *Restarting the Web Application Server on a C Series Controller*
 - *Stopping the Web Application Server on a C Series Controller*

Configuring the SAE to Send Tracking Events to the SRC VTA (SRC CLI)

The SRC VTA communicates with the SAE through the Enterprise JavaBean (EJB) adapter plug-in. This plug-in is an SAE plug-in and performs the following functions:

- Filters SAE plug-in events for the SRC VTA.
- Adapts internal SAE events to EJB-compatible methods.
- Sends SAE tracking plug-in events to the SRC VTA.

To configure the EJB adapter plug-in:

1. From configuration mode, access the EJB adapter plug-in configuration. In this sample procedure, the EJB adapter plug-in called QuotaVTA is configured in the nw-area SAE group.

```
user@host# edit shared sae group nw-area configuration plug-ins name QuotaVTA
ejb-adapter
```

2. Configure a list of all SRC VTAs to which SAE tracking events are sent.

```
[edit shared sae group nw-area configuration plug-ins name QuotaVTA ejb-adapter]
user@host# set application-server-url application-server-url
```

Specify the *application-server-url* as a comma-separated list of SRC VTA IP addresses or hostnames. The prefix *jnp://* and the suffix *:1099* are allowed, but unnecessary. The default value is 127.0.0.1.



NOTE: The setting of this option changes as of SRC software release 4.3.x. When upgrading from earlier releases of the SRC software to Release 4.3.x, it is mandatory that you configure this option. See the *SRC Release Notes* for Release 4.3 for the complete step-by-step upgrade procedure. It is very important that the upgrade procedure is followed exactly as described in the *SRC Release Notes*.

3. Configure the JNDI name of the SAEEventListener EJB of the peer SRC VTA. Because multiple SRC VTA groups are supported, you must specify the SAEEventListenerBean for each SRC VTA group. Specify the SAEEventListenerBean in the following format:

vta-VTA group name/SAEEventListenerBean

For example, if you want a particular SAE to send events to a particular SRC VTA instance (group) called “apple”, set the **jndi-sae-event-listener** as follows:

```
[edit shared sae group nw-area configuration plug-ins name QuotaVTA ejb-adapter]
user@host# set jndi-sae-event-listener vta-apple/SAEEventListenerBean
```

If you want a particular SAE to send some events to one SRC VTA instance (group) called “apple”, and some events (can be exactly the same events) to another SRC VTA instance called “orange”, configure two separate ejb-adapter plug-ins—one with the `jndi-sae-event-listener` set to “vta-apple/SAEEventListenerBean”, and the other with the `jndi-sae-event-listener` set to “vta-orange/SAEEventListenerBean.”

4. (Optional) Configure the LDAP filter that determines the subscriber and service events that the EJB adapter plug-in sends to the SRC VTA. If you specify plug-in attributes in this field, you must include the same attributes in the attributes option.

[edit shared sae group nw-area configuration plug-ins name QuotaVTA ejb-adapter]
user@host# set event-admitter *event-admitter*

Table 48 on page 631 lists the values that you can use for LDAP filter strings.

Table 48: Settings for Filter Strings

Filter String	Action
()	Matches no objects
(*)	Matches all objects
<p>List of <attribute>= <value> pairs</p> <p><attribute>—Name of a property or attribute <ldapAttributeName></p> <p><value>—One of the following:</p> <ul style="list-style-type: none"> * (asterisk) Explicit string String that contains an * <p>Note: To define a special character (*, &, !, \) in a string, precede it with the backslash symbol (\).</p>	<ul style="list-style-type: none"> If <value> is *, checks for any value. If <value> is an explicit string, checks whether any value of the property matches the string, regardless of case. If <value> is a string that contains a *, checks whether any value of the property contains the string, regardless of case.
(&<filter><filter>...)	True if all filters match
(<filter><filter>...)	True if at least one filter matches
(!<filter>)	True if the filter does not match

The variables in the filter include the names of plug-in attributes and a `PluginEventType` variable. The value of this variable is the name of the type of event, such as `PE_START_SERVICE`. For names of plug-in attributes and plug-in event types, see the SAE CORBA plug-in documentation on the Juniper Networks Web site at <http://www.juniper.net/techpubs/software/management/src/api-index.html> or in the **SDK+AppSupport+Demos+Samples.tar.gz** file on the Juniper Web site at <https://www.juniper.net/support/products/src/index.html>.

5. (Optional) Configure the plug-in attributes that the EJB adapter plug-in sends to the SRC VTA listener. If you do not define a list of attributes, the EJB adapter plug-in sends

all plug-in attributes to the SRC VTA. Sending unnecessary plug-in attributes can adversely affect the performance of SRC components.

```
[edit shared sae group nw-area configuration plug-ins name QuotaVTA ejb-adapter]
user@host# set attributes [(host | router-name | interface-name | ...)...]
```

Specify at least the following plug-in attributes: router-name, session-id, login-name, user-ip-address, ssp-host, domain, service-name, event-time, session-time, in-octets, out-octets, in-packets, out-packets, session-timeout, downstream-bandwidth, upstream-bandwidth, service-session-name, subscription-name. You may need to add attributes if you use them for the event admitter.

6. Configure how you want the SAE to send events to the SRC VTAs. Events can be sent using either the round-robin algorithm or the primary/backup algorithm.

```
[edit shared sae group nw-area configuration plug-ins name QuotaVTA ejb-adapter]
user@host# set use-primary-vta-if-available
```

By default, the SAE communicates with the SRC VTAs by using the round-robin algorithm. To use the primary/backup algorithm for SAE-VTA communication, enable the **use-primary-vta-if-available** option.

- Related Documentation**
- [How the SRC VTA Works on page 596](#)
 - [Installing the JDBC Driver .jar File on page 634](#)

Specifying Tracking Plug-Ins for Enterprise Subscribers on Junos OS Routing Platforms

When user-tracking plug-ins are attached to the retailer on routers running the Junos OS, login names are needed to trigger the user-tracking plug-in and generate user-tracking events. Because enterprise subscribers do not have a login name, the SRC VTA cannot get the required user-tracking events.

To allow enterprise subscribers on routers running the Junos OS to use retailer-attached user-tracking plug-ins, configure the EJB adapter plug-in to filter SAE plug-in events for the SRC VTA and send SAE tracking events to the SRC VTA.

To use the EJB adapter plug-in to send events for a specific retailer:

1. Configure the event admitter of the EJB adapter plug-in (see [“Configuring the SAE to Send Tracking Events to the SRC VTA \(SRC CLI\)” on page 630](#)).

Specify the PA_USER_DN event attribute with the retailer’s relative distinguished name (RDN). For example, the following event admitter matches events from subscribers in the SP-Quota retailer:

```
PA_USER_DN=*SP-Quota*
```

2. Configure the EJB adapter plug-in as the global subscriber-tracking plug-in for the SAE. See [Configuring Tracking Plug-Ins \(SRC CLI\)](#).

In the sample procedure, the EJB adapter plug-in called QuotaVTA is configured as the subscriber-tracking plug-in in the nw-area SAE group.

```
[edit shared sae group nw-area configuration plug-ins event-publishers]
user@host# set subscriber-tracking QuotaVTA
```

- Related Documentation**
- [Configuring the SAE to Send Tracking Events to the SRC VTA \(SRC CLI\) on page 630](#)
 - [Configuring Administrative Information for Enterprise Subscribers \(SRC CLI\)](#)

Configuring the External Database

This section describes how to configure an external database to store SRC VTA account and session data.

1. [Configuring a Database to Store Account and Session Data \(SRC CLI\) on page 633](#)
2. [Installing the JDBC Driver .jar File on page 634](#)

Configuring a Database to Store Account and Session Data (SRC CLI)

The SRC VTA requires a relational database to store accounts and session data. You must create this database on a separate, non-SRC machine before you can run the SRC VTA. For information about databases that we have tested for use with the SRC VTA, see the *SRC Release Notes*.

For each SRC VTA instance, you need to create a database that uses the schema for the SRC VTA. To configure a database:

1. From any C Series Controller running SRC software release 4.2.x or later, navigate to the /opt/UMC/vta/database/ directory and copy the appropriate SRC VTA schema file and save it on your external database machine. We provide the following schema files:
 - /opt/UMC/vta/database/vta-database-oracle.sql
 - /opt/UMC/vta/database/vta-database-mysql.sql

These files contain the SQL statements that you must execute to create a database in either a MySQL or Oracle database server. These files are examples that show the required database schema (in the file for MySQL, you must modify the database username and password before you can execute the SQL statements). If you have a different type of database, you can use these files as a reference on how to create a database with the same schema as is described in these two files.

2. Configure access to the database for an administrator by using the SRC VTA to monitor and manage subscribers. Edit the following options in the file as required for your external database:
 - SRC VTA database name
 - Username
 - Password

3. Use your standard mechanism to execute the SQL statements contained in the file. This creates a database inside your database server, with the schema needed by the SRC VTA.



NOTE: If the provided file does not contain SQL sufficient to create a database inside your database server, you must create the database manually. Use the provided file as a reference to understand the exact schema required by the SRC VTA

Installing the JDBC Driver .jar File

To allow the SRC Web application server to connect to your external database, you need to copy the JDBC driver .jar file for your database server brand and version to every system running the SRC VTA.

1. Obtain the relevant .jar file that contains the JDBC driver for your particular database server brand and version.
2. Copy the file to the `/opt/UMC/appsvr/common/lib` directory on every system running the SRC VTA. You can use the SRC CLI "file" commands, FTP, or SCP to copy the file.
3. Restart the Web application server on each SRC system.

```
user@host> restart component appsvr
```

Related Documentation

- [Configuring the Connection Between the SRC VTA and the External Account and Session Database \(SRC CLI\) on page 642](#)
- [Configuring the Web Application Server \(SRC CLI\) on page 621](#)
- [Web Application Server on C Series Controllers Overview](#)
- [Configuring the Web Application Server Shared Cluster Configuration \(SRC CLI\) on page 623](#)

Troubleshooting Database Deadlocks

Problem The JBoss application server logs the following error when the database reports a deadlock—a condition in which the database operation cannot continue because two processes are both waiting for the other process to be completed before they proceed.

```
java.sql.SQLException: General error, message from server: "Deadlock found when trying to get lock; Try restarting transaction"
```

Solution Deadlocks can occur for a variety of reasons in normal database operation. The SRC VTA resolves deadlocks in the database, and you should ignore this message.

Related Documentation

- [Installing Web Applications in the SRC Web Application Server on page 629](#)
- [Configuring a Database to Store Account and Session Data \(SRC CLI\) on page 633](#)

- [Configuring the Initial Balance and Status of a Subscriber Account in the External Database \(SRC CLI\) on page 652](#)

Configuring SRC VTA Services and Policies

Only the SRC VTA should activate and deactivate services that the SRC VTA controls, and you must ensure that these services are not visible on a portal for subscribers to control manually. You can use other services with the SRC VTA if you design the policies and priorities for those services to work together.

For example, if you manage subscribers with the SRC VTA, you can allow subscribers to manually activate a service that overrides the quota service, and consequently prevents charges in the periodic and bought accounts. You would account for use of this service through RADIUS rather than the SRC VTA, and subscribers would incur an extra cost for using the service. In this case, you configure the overriding service with a higher precedence than the quota service.

To configure services for the SRC VTA:

1. Create services for which the SRC VTA monitors and manages usage.
2. Configure policies that specify ingress and egress accounting rules consistent with the usage formula.

For information about configuring accounting rules for a policy, see *Policy Management Overview*.

Related Documentation

- [How the SRC VTA Works on page 596](#)
- [Configuring Subscribers and Subscriptions to SRC VTA Services on page 635](#)

Configuring Subscribers and Subscriptions to SRC VTA Services

To configure subscribers to SRC VTA services, see the *SRC PE Subscribers and Subscriptions Guide*.

1. Create at least one shared subscriber.
2. For all subscribers managed by the SRC VTA, create an individual or a group subscription to services for which the SRC VTA monitors and manages usage.

Related Documentation

- [SRC VTA Overview on page 593](#)
- [Locating the SAE That Manages a Subscriber for the SRC VTA on page 615](#)
- [Configuring SRC VTA Services and Policies on page 635](#)
- [Configuring a Database to Store Account and Session Data \(SRC CLI\) on page 633](#)

Configuring a NIC for the SRC VTA (SRC CLI)

For demonstrations and installations with few subscribers, you can configure the SRC VTA to use a NIC proxy stub, which explicitly defines a set of data mappings. However, for most standard installation with a significant number of subscribers and multiple SAEs, you need to set up a full NIC configuration. The requirement to set up a NIC depends on your SRC VTA configuration. For example, configure a NIC if you specify an action that affects more than just the current subscriber described by an event.

To configure a NIC for the SRC VTA:

1. Use the OnePopLogin configuration scenario (see [“NIC Configuration Scenarios” on page 140](#)).
2. Plan and configure the NIC hosts. See [“Configuring the NIC \(SRC CLI\)” on page 152](#).
3. Add the NIC SAE agents to each SAE configuration as external plug-ins. Specify these plug-in attributes: router-name, session-id, user-type, login-name, user-ip-address.
For information about configuring SAE plug-ins, see *Configuring the SAE for External Plug-Ins (SRC CLI)*.
4. (Optional) Configure a NIC proxy stub. See [“Configuring NIC Test Data \(SRC CLI\)” on page 193](#) for information about configuring the NIC proxy stub.
5. Configure a NIC proxy for the SRC VTA. See [“Configuring NIC Proxies for the SRC VTA” on page 636](#).

Related Documentation

- [Before You Configure the NIC on page 151](#)
- [Locating the SAE That Manages a Subscriber for the SRC VTA on page 615](#)
- [Starting the NIC \(SRC CLI\) on page 166](#)
- [Configuration Statements for the NIC on page 149](#)

Configuring NIC Proxies for the SRC VTA

For information about NIC proxies, see [“NIC Proxy Configuration Overview” on page 185](#).

You can configure a NIC proxy that passes the subscriber's identifier to a NIC resolver and receives the corresponding SAE reference. This NIC allows the SRC VTA to immediately activate a subscriber's quota service when a deposit is made to the subscriber's account. This feature is available for subscribers who connect to the network through routers running JunosE or Junos OS.

Use the following statement to configure the NIC proxy for the SRC VTA:

```
shared vta nic-proxy-configuration name
```

To configure the NIC proxy for the SRC VTA:

1. From configuration mode, access the statement that specifies the NIC proxy for the SRC VTA.

```
[edit]  
user@host# edit shared vta nic-proxy-configuration name
```

2. To complete the configuration of the NIC proxy, follow the procedure described in chapter *Configuring SRC Applications to Communicate with an SAE (SRC CLI)* in the *SRC PE Network Guide*.
3. Create a reference between the shared SRC VTA group configuration and the NIC proxy configuration.

From the **[edit shared vta group *name*]** hierarchy level execute:

```
[edit shared vta group name]  
user@host# set nic-proxy nic-proxy
```

Set the **nic-proxy** option to name you specified in Step 1. For more information, see [“Creating and Configuring an SRC VTA Shared Group Configuration \(SRC CLI\)”](#) on page 639.

**Related
Documentation**

- [Before You Configure a NIC Proxy on page 186](#)
- [Locating the SAE That Manages a Subscriber for the SRC VTA on page 615](#)
- [Configuring a NIC for the SRC VTA \(SRC CLI\) on page 636](#)

Configuring the SRC VTA (SRC CLI)

- [Configuring an SRC VTA Shared Group Configuration \(SRC CLI\) on page 639](#)
- [Configuring the Connection Between the SRC VTA and the External Account and Session Database \(SRC CLI\) on page 642](#)
- [Cleaning Up the Sessions Records and SessionBalanceChanges Records from the SRC VTA Database \(SRC CLI\) on page 644](#)
- [Configuring Actions for SRC VTA Event Handlers \(SRC CLI\) on page 647](#)
- [Configuring Event Handlers \(SRC CLI\) on page 648](#)
- [Configuring Event Queues for SRC VTA Groups \(SRC CLI\) on page 650](#)
- [Configuring the DB-Engine Processor for the SRC VTA Group \(SRC-CLI\) on page 651](#)
- [Configuring the Mailer Processor for the SRC VTA Group \(SRC CLI\) on page 661](#)
- [Configuring the SRC VTA Scripts Processor \(SRC CLI\) on page 662](#)
- [Configuring SRC VTA Logging \(SRC CLI\) on page 666](#)
- [Enabling, Disabling, and Restarting the SRC VTA \(SRC CLI\) on page 674](#)
- [Enabling the SOAP Interface for an SRC VTA Group \(SRC CLI\) on page 676](#)
- [Methods for the SRC Volume Tracking Application SOAP Interface on page 677](#)
- [SOAP Fault Codes for the SRC Volume Tracking Application on page 687](#)
- [Using One SRC VTA Account for Multiple Subscriber Sessions on page 698](#)

Configuring an SRC VTA Shared Group Configuration (SRC CLI)

- [Creating and Configuring an SRC VTA Shared Group Configuration \(SRC CLI\) on page 639](#)
- [Keys Used to Specify the Subscriber ID Solution \(SRC CLI\) on page 640](#)

Creating and Configuring an SRC VTA Shared Group Configuration (SRC CLI)

You can set up multiple SRC VTAs; each SRC VTA is configured as a separate group with its own shared configuration.

Use the following statements to configure an SRC VTA shared configuration:

```
shared vta group name {  
    subscriber-id-solution subscriber-id-solution ;  
    nic-proxy nic-proxy;  
}
```

To configure an SRC VTA shared configuration:

1. From configuration mode, access the statement that configures an SRC VTA shared group configuration. For example, to configure an SRC VTA group called `vta1`:

```
[edit]
user@host# edit shared vta group vta1
```

2. Specify the **subscriber-id-solution** option. This option specifies a data key that uniquely identifies subscriber accounts and sessions in the external database. Some settings also provide information that the NIC and the SAE use to identify subscribers.

```
[edit shared vta group vta1]
user@host# set subscriber-id-solution subscriber-id-solution
```

For more information about configuring the **subscriber-id-solution** option, see [“Keys Used to Specify the Subscriber ID Solution \(SRC CLI\)” on page 640](#).

3. Specify the location of the NIC proxy configuration relative to the configuration properties for the SRC VTA.

```
[edit shared vta group vta1]
user@host# set nic-proxy nic-proxy
```

Set this option to the name you configured under the **[edit shared vta nic-proxy-configuration *name*]** configuration statement. See [“Configuring NIC Proxies for the SRC VTA” on page 636](#).

If you are using a NIC to map subscriber identifiers to an SAE, and you select an SRC VTA subscriber ID value that provides a data key for the NIC, specify the NIC proxy that uses that data key.

4. Verify your configuration.

```
[edit shared vta group vta1]
user@host# show
```

5. Commit your configuration.

```
[edit shared vta group vta1]
user@host# commit
```



NOTE: You must enable the new SRC VTA instance with the `enable component vta-name` command.

Keys Used to Specify the Subscriber ID Solution (SRC CLI)

You use the **subscriber-id-solution** option under the **shared vta group *name*** configuration statement to specify keys for managing accounts and sessions. [Table 49 on page 641](#) shows the keys that you can specify for the SRC VTA to query the database, NIC, and SAE. For the SRC VTA to use a subscriber identifier, the plug-in event must include the

corresponding NIC or SAE attributes that are listed in [Table 49 on page 641](#) (attributes that start with PA_). For more information about plug-in attributes, see the documentation for the SAE CORBA plug-in on the Juniper Networks Web site at

<http://www.juniper.net/techpubs/software/management/src/api-index.html>

Table 49: Keys That the SRC VTA Constructs to Manage Accounts and Sessions

Subscriber's Identifier	Database Key	Corresponding NIC Key	Corresponding SAE Key
accounting-id	PA_ACCOUNTING_ID	PA_ACCOUNTING_ID	IP address that is returned from the NIC lookup
user-dn	PA_USER_DN	PA_USER_DN	PA_USER_DN
interface-alias	PA_INTERFACE_ALIAS	None	None
interface-alias-and-router	PA_INTERFACE_ALIAS@PA_ROUTER_NAME	PA_ROUTER_NAME	None
interface-and-router	PA_INTERFACE_NAME@PA_ROUTER_NAME	PA_ROUTER_NAME	PA_INTERFACE_NAME
login-name	PA_LOGIN_NAME	PA_LOGIN_NAME	PA_LOGIN_NAME (default)
mac-address	PA_USER_MAC_ADDRESS	None	None
primary-user-name (PPP login name or public DHCP name)	PA_PRIMARY_USER_NAME	None	PA_PRIMARY_USER_NAME
nas-port-id-and-router	PA_PORT_ID@PA_ROUTER_NAME	None	None

login-name, **user-dn**, and **interface-and-router** also provide data keys for the NIC and the SAE; the other settings do not.

Related Documentation

- [Managing SRC VTA Accounts and Sessions Overview on page 613](#)
- [SRC VTA Overview on page 593](#)
- [Using One SRC VTA Account for Multiple Subscriber Sessions on page 698](#)
- [Managing SRC VTA Accounts and Sessions on page 614](#)

Configuring the Connection Between the SRC VTA and the External Account and Session Database (SRC CLI)

The SRC VTA requires a relational database to store accounts and session data. The database connection information specifies how the SRC VTA connects to this database. You must configure the connection to the external database for each SRC VTA group.



NOTE: The values shown in the following sample procedure should work for a MySQL database. The values are different for an Oracle database, or other database servers. The values may also differ for different versions of the same brand of database server. You must understand how to connect to your particular brand and version of database server using JDBC before configuring the database connection.

Following is an example Oracle configuration:

```
user@host> show configuration shared vta group test database
check-valid-connection-sql 'select 1 from dual';
connection-url jdbc:oracle:thin:@//10.10.1.2:1521/vta;
datasource-mapping Oracle9i;
driver-class oracle.jdbc.OracleDriver;
max-pool-size 50;
min-pool-size 5;
password *****;
user-name vta;

user@host>
```

Use the following statements to configure the connection to the external database:

```
shared vta group name database {
  connection-url connection-url ;
  driver-class driver-class;
  user-name user-name;
  password password;
  data-source-mapping data-source-mapping;
  min-pool-size min-pool-size;
  max-pool-size max-pool-size;
  check-valid-connection-sql check-valid-connection-sql;
}
```

To configure the connection to the external database:

1. From configuration mode, access the statement that configures the connection to the external database. For example, to configure the connection for the SRC VTA group called vta1:

```
[edit]
user@host# edit shared vta group vta1 database
```

2. Configure the URL connection information. For example, to specify the connection url as jdbc:mysql://10.10.10.1:3306/vta1:

```
[edit shared vta group vta1 database]
user@host# set connection-url jdbc:mysql://10.10.10.1:3306/vta1
```

3. Configure the driver class information. For example, to configure a JDBC driver for MySQL:

```
[edit shared vta group vta1 database]
user@host# set driver-class com.mysql.jdbc.Driver
```

4. Configure the username used to access the database. For example to configure the username called admin1:

```
[edit shared vta group vta1 database]
user@host# set user-name admin1
```

5. Configure the password used to access the database. For example, to configure the password as pwd1:

```
[edit shared vta group vta1 database]
user@host# set password pwd1
```

6. Configure the data source mapping information. For example, for MySQL:

```
[edit shared vta group vta1 database]
user@host# set data-source-mapping mySQL
```

7. Specify the minimum number of simultaneous connections to the database.

```
[edit shared vta group vta1 database]
user@host# set min-pool-size min-pool-size ;
```

8. Specify the maximum number of simultaneous connections to the database.

```
[edit shared vta group vta1 database]
user@host# set max-pool-size max-pool-size ;
```



NOTE: The max-concurrency option should be set greater than or equal to the max-pool-size option; otherwise, database connections are likely to be unused, see [“Configuring Event Handlers Overview” on page 602](#).

9. Specify the SQL used to check whether a database connection is still valid.

```
[edit shared vta group vta1 database]
user@host# check-valid-connection-sql check-valid-connection-sql ;
```

The specified SQL depends on the type of database server. For example, with Oracle, you can use “select 1 from dual,” and with MySQL or MS SQL Server, you can use “select 1”.

Related Documentation

- [Configuring the External Database on page 633](#)

- [Configuring the Web Application Server \(SRC CLI\) on page 621](#)
- [Managing SRC VTA Accounts and Sessions on page 614](#)
- [Troubleshooting Database Deadlocks on page 634](#)

Cleaning Up the Sessions Records and SessionBalanceChanges Records from the SRC VTA Database (SRC CLI)

To remove the VTA sessions records and SessionBalanceChanges records automatically from the SRC VTA database, use the **edit shared vta group name database cleanup** configuration statement with the **interval**, **host-ip**, and **vta-sessions-max-age** or **session-balance-changes-max-age** options.

The automatic cleanup mechanism periodically deletes the numerous stale database records from the SRC VTA database and keeps only the user specified time–line of data in the SRC VTA database. Hence, reduces the database table size, requires less storage and eliminates the performance degradation with the SRC VTA database access.



NOTE: The automatic cleanup mechanism supports on Oracle and MySQL database servers.

The automatic cleanup mechanism removes the following:

- Stopped session records that are older than the value configured using the **vta-sessions-max-age** option.
- Unstopped orphan session records that did not receive an interim update for the entire period of the value configured using the **vta-sessions-max-age** option.
- SessionBalanceChange records that are older than the value configured using the **session-balance-changes-max-age** option.

The automatic cleanup mechanism does not remove the records from the balance change table.

Use the following statements to clean up the VTA sessions records and SessionBalanceChanges records automatically:

```
shared vta group name database cleanup{
    interval (daily | weekly | monthly)
    vta-sessions-max-age vta-sessions-max-age;
    session-balance-changes-max-age session-balance-changes-max-age;
    host-ip host-ip;
    time-duration time-duration;
}
```

**NOTE:**

1. When you execute the `edit shared vta group name database cleanup` configuration statement, ensure that the VTA database contains only the VTA sessions records and SessionBalanceChanges records not older than the value planned to be set for `vta-sessions-max-age` *vta-sessions-max-age* and `session-balance-changes-max-age` *session-balance-changes-max-age* options. Else it is recommended to remove the old records manually using third party vendor support or cleanup scripts or using the following commands:
 - `request vta delete balance-changes`
 - `request vta delete sessions`
2. You must configure an appropriate value for `vta-sessions-max-age` *vta-sessions-max-age* and `session-balance-changes-max-age` *session-balance-changes-max-age*, because you cannot recover any deleted records and that may have an impact on billing or revenue.

To automatically clean up the VTA sessions records and SessionBalanceChanges records from the SRC VTA database:

1. From configuration mode, access the statement that runs the cleanup mechanism.

```
[edit]
```

```
user@host# edit shared vta group vta1 database cleanup
```

2. Configure the time interval (daily or weekly or monthly) for running the cleanup mechanism.

Use the options described in [Table 50 on page 645](#) to configure the cleanup interval.

For example,

```
[edit shared vta group vta1 database cleanup]
```

```
user@host# set interval daily hour hour minute minute
```

You can clean up the SRC VTA database tables in the following time interval:

Table 50: Options Used to Configure the Cleanup Interval

Option	Description
daily	(Optional) Cleans up the data every day for the specified hour and minute of the day.
weekly	(Optional) Cleans up the data weekly for the specified day of week, hour, and minute of the day.
monthly	(Optional) Cleans up the data monthly for the specified day of month, hour, and minute of the day.

To specify the day of week, month, day of month, hour, and minute, use the **day-of-week**, **month**, **day-of-month**, **hour**, and **minute** options.

The values allowed for these options are described in [Table 51 on page 646](#):

Table 51: Values Allowed for the Options

Options	Values
day-of-week	1-7 or SUN-SAT
month	1-12
day-of-month	1-31
hour	0-23
minute	0-59

3. (Optional) Configures the maximum age for the records (in number of months) in the VTA_SESSIONS table. The value range is 1 through 24.

```
[edit shared vta group vta1 database cleanup]
user@host# set vta-sessions-max-age vta-sessions-max-age
```

4. (Optional) Configures the maximum age for the records (in number of months) in the SESSION_BALANCE_CHANGE table. The value range is 1 through 24.

```
[edit shared vta group vta1 database cleanup]
user@host# set session-balance-changes-max-age session-balance-changes-max-age
```

5. Specifies the VTA host name or the IP address of the SRC VTA node. In a cluster of SRC VTA nodes, only one SRC VTA performs the database cleanup. If the **host-ip** value is changed to an IP address other than the current VTA node's IP address, then the scheduler is cancelled in the current VTA node.

```
[edit shared vta group vta1 database cleanup]
user@host# set host-ip host-ip
```

6. (Optional) Defines the maximum time for which the cleanup task can take. The cleanup task suspends when it reaches the time limit value. The value range is 30 through 900 minutes. The default value is 60 minutes.

```
[edit shared vta group vta1 database cleanup]
user@host# set time-duration time-duration
```

7. (Optional) Verify your configuration.

```
user@host# show
host-ip 10.212.10.20;
interval {
  daily {
    hour 12;
    minute 30;
  }
}
session-balance-changes-max-age 2;
time-duration 60;
vta-sessions-max-age 3;
```

- Related Documentation**
- [Configuring the Connection Between the SRC VTA and the External Account and Session Database \(SRC CLI\) on page 642](#)
 - [SRC VTA Overview on page 593](#)
 - [Configuring the External Database on page 633](#)

Configuring Actions for SRC VTA Event Handlers (SRC CLI)

Use the following statement to configure an action that can be invoked by one or more event handlers:

```
shared vta group name action action-name {
    function (db-engine-calculate-interim | db-engine-calculate-usage |
        db-engine-get-accounts | db-engine-terminate-session | db-engine-update-accounts
        | mailer-send | sae-set-interim-interval | sae-set-service-timeout | sae-set-user-timeout
        | sae-start-service | sae-stop-service | scripts-run-external-script |
        scripts-run-javascript);
    on-error (abort-event-processing | go-to-next-action | go-to-next-event-handler);
    parameter;
}
```



NOTE: We recommend that when you configure event handlers and their actions, you ensure that for any given event, all database operations are performed before any other operations that have permanent effects. This is because if a database error occurs—for example, due to normal contention for database records between different event threads—the SRC VTA rolls back the current database transaction (no changes are made to the database) and then restarts processing the event. If the event performs some other operation other than database operations before such an error, such as start a service, then that other operation is performed again when the event is reprocessed following the error.

1. From configuration mode, access the statement that configures an action and specify a name for the action. For example, to configure an action called `act1` in the SRC VTA group called `vta1`:

```
[edit]
user@host# edit shared vta group vta1 action atc1
```

2. Specify a function you want the action to call. For example, to have the action call the `db-engine-calculate-usage` function:

```
[edit shared vta group vta1 action atc1]
user@host# set function db-engine-calculate-usage
```

3. (Optional) Specify any input parameters required for the function.

```
[edit shared vta group vta1 action atc1]
user@host# set parameter
```

The parameters that you need to set depend on what function you specified. Some functions require no input parameters. You can use the “?” completer to see what parameters need to be set, or see [“Configuring Actions Overview” on page 606](#) for a complete list of functions and their associated input parameters.

4. Specify what you want the action to do in response to an error. For example, to configure the action to abort processing if an error occurs:

```
[edit shared vta group vta1 action atc1]
user@host# set on-error abort-event-processing
```

5. Verify your configuration.

```
[edit shared vta group vta1 action atc1]
user@host# show
```

```
user@host# show
function db-engine-calculate-usage;
on-error abort-event-processing;
```

```
[edit shared vta group vta1 action atc1]
user@host#
```

6. Commit your configuration.

```
[edit shared vta group vta1 action atc1]
user@host# commit
```

**Related
Documentation**

- [Configuring Actions Overview on page 606](#)
- [Configuring Event Handlers Overview on page 602](#)
- [Configuring Event Handlers \(SRC CLI\) on page 648](#)

Configuring Event Handlers (SRC CLI)

Use the following statements to configure event handlers:

```
shared vta group name event-handler event-handler-name {
  events events;
  actions actions;
  condition condition;
  priority priority;
}
```

To configure an event handler:



NOTE: We recommend that when you configure event handlers and their actions, you ensure that for any given event, all database operations are performed before any other operations that have permanent effects. This is because if a database error occurs—for example, due to normal contention for database records between different event threads—the SRC VTA rolls back the current database transaction (no changes are made to the database) and then restarts processing the event. If the event performs some other operation other than database operations before such an error, such as start a service, then that other operation is performed again when the event is reprocessed following the error.

1. From configuration mode, access the statement that configures SRC VTA event handlers. For example, to configure an event handler called evh1 in the SRC VTA group called vta1:

```
[edit]
user@host# edit shared vta group vta1 event-handler evh1
```

2. Specify an event. For example, to have the event handler handle start events for the service named QuotaInternet:

```
[edit shared vta group vta1 event-handler evh1]
user@host# set events service-start:QuotaInternet
```

3. Specify previously configured actions that the event handler invokes in response to the event. For example, to specify the previously configured actions called act1, act2, and act3:

```
[edit shared vta group vta1 event-handler evh1]
user@host# set actions [act1 act2 act3]
```

4. Specify the condition the event handler uses to determine whether it should handle the event.

```
[edit shared vta group vta1 event-handler evh1]
user@host# set condition condition
```

Specify the condition as script written in the JavaScript programming language that must return one of the following Boolean values:

- True—Event handler should handle the event.
- False—Event handler should not handle the event.

5. Specify the priority of the event handler.

```
[edit shared vta group vta1 event-handler evh1]
user@host# set priority priority
```

Event handlers with the lower number priority have the highest priority and evaluate the event first.

6. Verify your configuration.

```
[edit shared vta group vta1 event-handler evh1]
user@host# show
```

**Related
Documentation**

- [How the SRC VTA Works on page 596](#)
- [Configuring Actions Overview on page 606](#)
- [Configuring Event Handlers Overview on page 602](#)
- [Configuring Actions for SRC VTA Event Handlers \(SRC CLI\) on page 647](#)

Configuring Event Queues for SRC VTA Groups (SRC CLI)

Use the following statement to configure the event queue:

```
shared vta group name queue {
  max-concurrency max-concurrency;
  persistent;
}
```

To configure the event queue:

1. From configuration mode, access the statement that configures the event queue. For example, to configure the event queue for the SRC VTA group called vta1:

```
[edit]
user@host# edit shared vta group vta1 queue
```

2. Specify the maximum number of threads consuming events from the queue and passing them to the configured event handlers.

```
[edit shared vta group vta1 queue]
user@host# set max-concurrency max-concurrency
```



NOTE: max-concurrency should be set greater than or equal to the max-pool-size option; otherwise, database connections are likely to be unused, see [“Configuring the Connection Between the SRC VTA and the External Account and Session Database \(SRC CLI\)” on page 642](#).

3. (Optional) Specify whether the event queue is persistent (events are saved to disk) or non-persistent (events are saved in memory). If set, events are saved to disk. By default, the event queue is set to non-persistent.

```
[edit shared vta group vta1 queue]
user@host# set persistent
```

**Related
Documentation**

- [Configuring Event Handlers Overview on page 602](#)

Configuring the DB-Engine Processor for the SRC VTA Group (SRC-CLI)

The SRC VTA uses the db-engine processor to update database and subscriber accounts. The db-engine processor works as a proxy to the external database. It calculates usage, updates account balances, retrieves account and active session data, and sets initial balances of subscriber accounts. A subscriber account is a record of credit and debit entries in the database that track a subscriber's use of a particular network resource. You can also use the db-engine processor to dynamically adjust interim accounting intervals based on a service or based on a subscriber's remaining resources and use of the network for that service. Configuring the db-engine processor involves the following tasks:

- [Recording Balance Changes and Calculating the Average Usage Rate \(SRC CLI\) on page 651](#)
- [Configuring the Initial Balance and Status of a Subscriber Account in the External Database \(SRC CLI\) on page 652](#)
- [Configuring Scripts That Update Accounts \(SRC CLI\) on page 653](#)
- [Configuring the Interim Account Interval and Usage Metric of a Service in the External Database \(SRC CLI\) on page 654](#)
- [Configuring SRC VTA Database Optimization \(SRC CLI\) on page 655](#)
- [Variables Used to Define the Interim Accounting Interval for Services on page 656](#)
- [Variables Used to Define the Usage Metric for Services on page 659](#)
- [Sample Formulas for Usage Metrics for the SRC VTA on page 660](#)

Recording Balance Changes and Calculating the Average Usage Rate (SRC CLI)

You can configure the db-engine processor to record account balance changes and calculate the average rate at which the subscriber is consuming volume in units per second, called the `averageUsageRate`.

The **session-history-depth** option determines the number of historical service session records that are used to calculate the `averageUsageRate` for a service. When you configure the interim accounting interval, you can specify the variable `averageUsageRate_serviceName`. If this variable is specified, the SRC VTA calculates the `averageUsageRate` for the specified service by querying the history sessions of the service over the most recent *X* hours, where *X* is defined by the **session-history-depth** option.

For more information, see “[Configuring the Interim Account Interval and Usage Metric of a Service in the External Database \(SRC CLI\)](#)” on page 654 and “[Variables Used to Define the Interim Accounting Interval for Services](#)” on page 656.

Use the following statements to configure the SRC VTA to record balance changes and specify the session history depth:

```
shared vta group name processor db-engine {
    record-balance-change;
    session-history-depth;
}
```



NOTE: The CLI editing level must be set to expert to set the **session-history-depth** option.

To configure the SRC VTA to record balance changes and specify the session history depth:

1. From configuration mode, access the statement used to configure the db-engine processor. For example, to configure the db-engine processor for the SRC VTA group called **vta1**:

```
[edit]
user@host# edit shared vta group vta1 processor db-engine
```

2. (Optional) Configure the SRC VTA to record balance changes.

```
[edit shared vta group vta1 processor db-engine]
user@host# set record-balance-change
```

This option is set or not set. If it is set, every account balance change action is recorded in the database. Not setting this option requires fewer database updates and less database space.

3. (Optional) Specify the session-history-depth (in hours) used to calculate the **averageUsageRate** for a service.

```
[edit shared vta group vta1 processor db-engine]
user@host# set session-history-depth
```

Configuring the Initial Balance and Status of a Subscriber Account in the External Database (SRC CLI)

Use the following statements to set the initial balance and status of a subscriber account:

```
shared vta group name processor db-engine account account{
  initial-balance initial-balance;
  initial-status initial-status;
}
```

To set the initial balance and status of a subscriber account:

1. From configuration mode, access the statement used to configure database accounts. For example, to configure an account called **BoughtQuota** in the SRC VTA group called **vta1**:

```
[edit]
user@host# edit shared vta group vta1 processor db-engine account BoughtQuota
```

2. Specify the initial balance for the account.

```
[edit shared vta group vta1 processor db-engine account BoughtQuota]
user@host# set initial-balance initial-balance
```

Enter the value as an integer in the range -9223372036854775807 through 9223372036854775807.

3. Specify the initial status for the account.

```
[edit shared vta group vta1 processor db-engine account BoughtQuota]
user@host# set initial-status initial-status
```

Enter the initial status as a text string—for example “active.”

Configuring Scripts That Update Accounts (SRC CLI)

You can set up scripts to update balances in the accounts from which the usage of a service is charged and update accounts by assigning values to variables for the account balances.

Use the following statements to configure scripts that update accounts:

```
shared vta group name processor db-engine account-update-script name{
  script script;
}
```

To configure scripts that update accounts:

1. From configuration mode, access the statement used to configure scripts that update accounts. For example, to configure a script called DebtQuotaUsage in the SRC VTA group called vta1:

```
[edit]
user@host# edit shared vta group vta1 processor db-engine account-update-script
DebtQuotaUsage
```

2. Specify the script parameters.

```
[edit shared vta group vta1 processor db-engine account-update-script
DebtQuotaUsage]
user@host# set script script
```

Enter a JavaScript program that updates a subscriber's account. The script can refer to the name of any attributes in the event being processed. An account can be updated by assigning values to the following parameters:

- `balance_<accountName>`—Values written to this parameter are put in the balance field of the account.
- `status_<accountName>`—Values written to this parameter are put in the status field of the account.
- `lastUpdateTime_<accountName>`—Values written to this parameter are put in the `last_update_time` field of the account.

For example:

```
[edit shared vta group vta1 processor db-engine account-update-script
DebtQuotaUsage]
```

```
user@host# set script <balance_PeriodicQuota>=<balance_PeriodicQuota>-\  
<currentUsage>;<lastUpdateTime_PeriodicQuota>=<currentTime>;
```

Configuring the Interim Account Interval and Usage Metric of a Service in the External Database (SRC CLI)

To configure how the db-engine processor manages services, you need to specify the usage metric and the interim accounting interval options. Both of these options are configured with JavaScript variables.

Use the following statement to configure how the SRC VTA manages services:

```
shared vta group name processor db-engine service name {  
    interim-interval-function interim-interval-function;  
    usage-metric usage-metric;  
}
```



NOTE: The CLI editing level must be set to expert for this task.

To configure how the db-engine processor manages services:

1. From configuration mode, access the statement that configures the db-engine processor to manage services and specify the name of the service. For example, to configure a service called QuotaLocal in the SRC VTA group vta1:

```
[edit]  
user@host# edit shared vta group vta1 processor db-engine service QuotaLocal
```

2. Specify the JavaScript function that defines the usage metric for the service.

```
[edit shared vta group vta1 processor db-engine service QuotaLocal]  
user@host# set usage-metric usage-metric
```

3. Specify the JavaScript variables that define the interim accounting interval for the service.

```
[edit shared vta group vta1 processor db-engine service QuotaLocal]  
user@host# set interim-interval-function interim-interval-function
```

Configuring SRC VTA Database Optimization (SRC CLI)

To optimize SRC VTA database operations, you can set the **sessions-terminated-frequently** option.



NOTE: Setting the **sessions-terminated-frequently** option is an irreversible act. You can set this option to true, but you cannot change it back to false.

To use the **sessions-terminated-frequently** optimization feature with a set of SRC VTAs, you must follow this specific process:

1. All C Series Controllers running SRC VTAs must be running SRC software release 4.3 or later; otherwise, you must upgrade the system.
2. Send all SAE events to only one SRC VTA.
3. Wait until the removed SRC VTAs (from Step 2) have finished processing events in their event queues.
4. Set the **sessions-terminated-frequently** option to true.
5. Wait until all SRC VTAs have reacted to the option being set. The SRC VTA information log displays the following message "DBEngine will assume VTA sessions are terminated frequently."
6. Resume distributing SAE events to all SRC VTAs.

Use the following statement to configure SRC VTA database optimization:

```
shared vta group name processor db-engine {
  sessions-terminated-frequently ;
}
```

To configure SRC VTA database optimization:

1. From configuration mode, access the statement used to configure the db-engine processor. For example, to set up SRC VTA database optimization for the SRC VTA group called **vta1**:

```
[edit]
user@host# edit shared vta group vta1 processor db-engine
```

2. Set the **sessions-terminated-frequently** option.

```
[edit shared vta group vta1 processor db-engine]
user@host# set sessions-terminated-frequently
```

Variables Used to Define the Interim Accounting Interval for Services

Current Service Variables

Use the variables described in this section to define a formula for the interim accounting interval.

lastInterimTime

- Last interim time interval.
- Value—Number of seconds in the range 1–2147483647

sessionLength

- Length of the current session.
- Value—Number of seconds in the range 0–2147483647; value is 0 when the SRC VTA is calculating the interim time of start events. For other events, value is set by the PA_SESSION_TIME attribute.

maxUsageRate

- Maximum rate at which the subscriber can use network resources according to the formula described in [Table 52 on page 658](#).
- Value—Integer in the range 0–9223372036854775807
- Guidelines—This formula corresponds to the usage formula for the same service as the interim formula.

The maxUsageRate variable is calculated for a service by means of the following values for the variables in the corresponding usage formula:

- upStreamBytes=PA_UPSTREAM_BANDWIDTH
 - downStreamBytes=PA_DOWNSTREAM_BANDWIDTH
 - interimTime=lastInterimTime
 - upStreamPackets=0
 - downStreamPackets=0

If you use the parameters upStreamPackets (PA_IN_PACKETS) and downStreamPackets (PA_OUT_PACKETS) in the usage formula and at the same time maxUsageRate in the interim interval formula, the maxUsageRate is not accurate, because the values for maximum upStreamPackets and downStreamPackets are unknown.

averageUsageRate

- Average rate at which the subscriber is consuming volume in units per second. The unit can be a value such as dollars, bytes, or packets. The type of unit depends on the value specified in the formula. Measurement begins when the service starts.
- Value—Integer in the range 0–9223372036854775807; the value is 0 when the SRC VTA is calculating the interim time of start events.

For other events, the value is the usage formula divided by PA_SESSION_TIME. The usage formula is calculated from PA_IN_PACKETS, PA_OUT_PACKETS, PA_OUT_OCTETS, PA_IN_OCTETS, and PA_SESSION_TIME.

latestUsageRate

- Rate of service usage since the last usage report.
- Value—Integer in the range 0–9223372036854775807; the value is 0 when the SRC VTA is calculating the interim time of start events.

The value is calculated by using the result of the usage formula divided by the length of the service session since the previous usage report for the same service.

Other Service Variables

Use the variables described in this section to define an interim accounting formula that depends on usage of another service tracked by the SRC VTA.

System requirements to calculate service usage, in the form of the averageUsageRate and the sessionLength variables, can affect system performance. Using a longer interim interval means that there are fewer interim events to process, which requires fewer system resources.

averageUsageRate_<serviceName>

- Average rate at which the service is consuming volume in units per second. The unit can be a value such as dollars, bytes, or packets. The type of unit depends on the value specified in the interim accounting formula. Measurement begins when the service starts.
- Value—Integer in the range 0–9223372036854775807; the value is 0 when the SRC VTA is calculating the interim time of start events.
- Guidelines—Service names can contain alphanumeric characters and dashes (-).

sessionLength_<serviceName>

- Length of a service session for the service.
- Value—Integer in the range 0–2147483647; the value is 0 when the SRC VTA is calculating the interim time of start events.
- Guidelines—Service names can contain alphanumeric characters and dashes (–).

Account Balance Variable

Use the variable described in this section to obtain balance information from each of the subscriber's accounts.

balance_<accountName>

- Balance for the specified account before the new usage value is applied.
- Value—Integer in the range 0–9223372036854775807
- Example—balance_PeriodicQuota refers to the balance for the PeriodicQuota account.

Sample Formulas for Interim Accounting Interval

Table 52 on page 658 provides examples of formulas to dynamically adjust the interim accounting interval for a service.

Table 52: Examples of Interim Accounting Interval

Formula	Description
return 900	Accounting interval is fixed at 900 seconds (15 minutes).
return (<balance_Periodic> + <balance_Bought>) /<maxUsageRate>	Minimum time required for the subscriber to empty the periodic and bought accounts.
return <sessionLength> >= 60*15 ? (<balance_Periodic> + <balance_Bought>) /<averageUsageRate>/2 : (<balance_Periodic> + <balance_Bought>) /<maxUsageRate>	Half the time required for the subscriber to empty the accounts at the current average rate, or the minimum time if the session is shorter than 15 minutes. Because the average rate may not be representative early in the session, check when the account is half empty.

Variables Used to Define the Usage Metric for Services

A usage metric is a formula that calculates usage based on an accounting event for the specified service. The formula is specified in the form of a JavaScript program and it can specify variables. When you configure a service, you need to specify the interim interval and the usage metric.

For the usage metric, you define a formula that determines the use of network resources for a service. Each service in the SRC VTA can use a different formula. You can configure the SRC VTA software to evaluate this formula for every accounting event it receives from the SAE for each quota service. The SRC VTA can then debit the result from the accounts.

Use the variables described in this section to define the formula. See [“Sample Formulas for Usage Metrics for the SRC VTA” on page 660](#) for examples of usage metric formulas.

downStreamBytes

- Amount of data that the subscriber downloaded from the network since the last accounting event.
- Value—Number of bytes in the range 0–9223372036854775807

downStreamPackets

- Number of data packets that the subscriber downloaded from the network since the last accounting event.
- Value—Integer in the range 0–9223372036854775807
- Guidelines—Do not use `downStreamPackets` in a usage formula and `maxUsageRate` in the interim interval formula for the same service at the same time.

interimTime

- Time since the last accounting event.
- Value—Number of seconds in the range 0–2147483647
- Guidelines—Generally, this value equals the interim accounting interval; however, it may exceed the interim accounting interval if an accounting event is lost. Similarly, the value may be less than the interim accounting interval if a stop event occurs in the middle of an accounting interval.

upStreamBytes

- Amount of data that the subscriber uploaded to the network since the last accounting event.
- Value—Number of bytes in the range 0–9223372036854775807

upStreamPackets

- Number of data packets that the subscriber uploaded to the network since the last accounting event.
- Value—Integer in the range 0–9223372036854775807
- Guidelines—Do not use upStreamPackets in a usage formula and maxUsageRate in the interim interval formula for the same service at the same time.

Sample Formulas for Usage Metrics for the SRC VTA

Table 53 on page 660 provides examples of usage formulas.

Table 53: Examples of Formulas That Calculate Use of Network Resources

Formula	Description	Function
return <upStreamBytes> + <downStreamBytes>	Number of bytes sent and received by the subscriber.	Tracks volume of data that the subscriber transfers.
return 2*<upStreamBytes> + <downStreamBytes>	Twice the number of sent bytes plus the number of received bytes.	Allows higher charges for subscribers who are operating servers.
return <interimTime>	Time the subscriber is connected.	Tracks time that the subscriber connects rather than volume of data transfer.
return <downStreamBytes>/<interimTime>	Rate of downstream data transfer.	Allows higher charges for higher transfer rates.
QuotaInternet formula: return <upStreamBytes> + <downStreamBytes> – (<upStreamPackets> + <downStreamPackets>)*20 QuotaLocal formula: return (<upStreamBytes> + <downStreamBytes> – (<upStreamPackets> + <downStreamPackets>)*20)/2	Formulas for separate, complementary services in a single SRC VTA. The following expression returns the total number of bytes in the IP headers of packets uploaded and downloaded by the service, and as such is not subscriber data. It is not counted as usage. (<upStreamPackets> + <downStreamPackets>)*20	Provides support for two services: QuotaInternet for Internet service and QuotaLocal for local service. Allows higher charges for Internet service than for local service. By allocating a fixed usage limit for both services to each subscriber, the formula encourages subscribers to access local resources due to decreased cost.

Related Documentation

- [Adjusting the Interim Accounting Interval Overview on page 614](#)

- [Managing SRC VTA Accounts and Sessions Overview on page 613](#)
- [How the SRC VTA Works on page 596](#)

Configuring the Mailer Processor for the SRC VTA Group (SRC CLI)

Configuring the mailer processor involves the following tasks:

- [Configuring the SRC VTA Mailer Processor to Send E-Mail Notifications \(SRC CLI\) on page 661](#)
- [Configuring the SRC VTA to Send E-Mail Notifications \(SRC CLI\) on page 662](#)

Configuring the SRC VTA Mailer Processor to Send E-Mail Notifications (SRC CLI)

Use the SRC VTA mailer processor to specify the SMTP server to use for e-mail messages that the SRC VTA sends to subscribers.

Use the following statement to configure the mailer processor:

```
shared vta group name processor mailer {
  smtp-server smtp-server;
  smtp-server-password smtp-server-password;
  smtp-server-port smtp-server-port ;
  smtp-server-user smtp-server-user;
}
```

1. From configuration mode, access the statement that configures the SRC VTA mailer processor. For example, to configure the mailer processor for the SRC VTA group called `vta1`:

```
[edit]
user@host# edit shared vta group vta1 processor mailer
```

2. Configure the SMTP server used for outgoing e-mail by specifying the host name or IP address of the SMTP server. For example, set the SMTP server to a host called `mail.example.com`:

```
[edit shared vta group vta1 processor mailer]
user@host# set smtp-server mail.example.com
```

3. Configure the password used to access the SMTP server.

```
[edit shared vta group vta1 processor mailer]
user@host# set smtp-server-password smtp-server-password
```

4. Configure the port used to connect to the SMTP server.

```
[edit shared vta group vta1 processor mailer]
user@host# set smtp-server-port smtp-server-port
```

5. Configure the username for accessing the SMTP server. `mail.example.com`:

```
[edit shared vta group vta1 processor mailer]
user@host# set smtp-server-user smtp-server-user
```

6. Commit your configuration.

```
[edit shared vta group vta1 processor mailer]
user@host# commit
```

Configuring the SRC VTA to Send E-Mail Notifications (SRC CLI)

To configure the SRC VTA to send e-mail notifications, you need to configure an event handler that specifies an action that calls the *mailer-send* function.

1. From configuration mode, access the statement that configures an action for the SRC VTA event handler. For example, to create an action called *act1* for the SRC VTA group called *vta1*:

```
[edit]
user@host# edit shared vta group vta1 action act1
```

2. Configure the action to call the *mailer-send* function.

```
[edit shared vta group vta1 action act1]
user@host# set function mailer-send
```

3. Configure the parameters for the *mailer-send* function.

```
[edit shared vta group vta1 action act1]
user@host# set parameter
```

Specify the following parameters for the *mailer-send* function:

- Recipient—Address of the e-mail recipient
 - From—Address of the e-mail sender
 - Subject—Subject of the e-mail
 - Text—Text of the e-mail
4. Configure what to do if an error occurs. For example, to configure the event handler to go to the next action if an error occurs:

```
[edit shared vta group vta1 action act1]
user@host# set on-error go-to-next-action
```

5. Configure the event handler and specify the action you configured in the previous steps of this procedure.

See [“Configuring Event Handlers \(SRC CLI\)” on page 648](#).

Configuring the SRC VTA Scripts Processor (SRC CLI)

The scripts processor can invoke external executable scripts or JavaScript programs. We recommend using JavaScript programs for better performance.

- External scripts are executable programs, such as shell scripts, that are available on the SRC VTA's host. Each external script can perform a task and return a value. If the script returns a value, the value can be added to the current event as an event attribute.
- JavaScript programs are used to process SRC VTA event attributes. For example, a JavaScript program can convert SRC VTA event attributes in a timestamp to a date string and add them to the event as a new attribute. The attributes can then be used for subsequent actions, such as sending an e-mail notification to the subscriber. The JavaScript program can refer to any attributes of the event being processed and return a value.

Configuring the SRC VTA scripts processor involves the following tasks:

- [Configuring the SRC VTA to Run Scripts on page 663](#)
- [Configuring JavaScript Programs on page 663](#)
- [Configuring External Scripts on page 664](#)

Configuring the SRC VTA to Run Scripts

To configure the scripts processor:

1. Configure a JavaScript program or an external script.
 - See [“Configuring JavaScript Programs” on page 663](#)
 - See [“Configuring External Scripts” on page 664](#)
2. Configure an action for the event handler that calls the appropriate script function.

The action specifies a function that the scripts processor performs on events. You can set up an action to run either an external script or a JavaScript program.

- Specify the *scripts-run-javascript* function if the script is a JavaScript.
- Specify the *scripts-run-external-script* function if the script is an external script.

When you specify these functions, you need to specify the name of a script you previously configured by using the **shared vta group id processor scripts** statement. If the script configuration contains a return-attribute and return-type, an attribute with that name and type are added to the event after the script is executed.

See [“Configuring Actions for SRC VTA Event Handlers \(SRC CLI\)” on page 647](#)

3. Configure an event handler and specify the action you configured in Step 2.

See [“Configuring Event Handlers \(SRC CLI\)” on page 648](#)

Configuring JavaScript Programs

Use the following statement to configure a JavaScript program for the SRC VTA:

```
shared vta group name processor scripts javascript name {
  script script;
  return-type return-type;
  return-attribute return-attribute;
}
```

1. From configuration mode, access the statement that configures a JavaScript script and specify the name of the JavaScript program. For example, to configure a JavaScript script called `js1`:

```
[edit]
user@host# edit shared vta group vta1 processor scripts javascript js1
```

2. Specify a function body in JavaScript.

```
[edit shared vta group vta1 processor scripts javascript js1]
user@host# set script script
```

To refer to the event attributes being processed, include the attribute name delimited by angle brackets (`<and>`). The JavaScript program can verify whether the event has the referenced attribute. If a referenced attribute does not exist in the event, the attribute's value is null. The JavaScript program must return a value.

3. (Optional) Specify the Java type of the return attribute.

```
[edit shared vta group vta1 processor scripts javascript js1]
user@host# set return-type return-type
```

Where the **return-type** is one of the following:

- Integer
- Long
- Float
- Double
- String
- Boolean

4. (Optional) Specify the name of the return attribute.

```
[edit shared vta group vta1 processor scripts javascript js1]
user@host# set return-attribute return-attribute
```

Specify an attribute that provides the return value of the script as a valid Java identifier that subsequent actions and event handlers can refer to. If this attribute is not set, the return value is not added to the event as an event attribute.

JavaScript scripts must return an attribute value. The name of a return attribute cannot start with an underscore (`_`) because these event attributes are reserved for internal use.

Configuring External Scripts

Use the following statement to configure an external script for the SRC VTA:

```
shared vta group name processor scripts external-script name {
  full-path full-path;
  parameters parameters;
```



```

return-type return-type;
return-attribute return-attribute;
}

```

1. From configuration mode, access the statement that configures an external script and specify the name of the external script. For example, to configure an external script called script1:

```

[edit]
user@host# edit shared vta group vta1 processor scripts external-script script1

```

2. Specify the full path name to the external script.

```

[edit shared vta group vta1 processor scripts external-script script1]
user@host# set full-path full-path

```

You must specify a fully qualified path to an executable program on the local file system.

3. Specify the parameters required by the script as [a b c]. When an action invokes the script, the action must supply values for these parameters in the same order as they are defined here.

```

[edit shared vta group vta1 processor scripts external-script script1]
user@host# set parameters parameters

```

4. (Optional) Specify the Java type of the return attribute.

```

[edit shared vta group vta1 processor scripts external-script script1]
user@host# set return-type return-type

```

Where return-type is one of the following:

- Integer
- Long
- Float
- Double
- String
- Boolean

5. (Optional) Specify the name of the return attribute.

```

[edit shared vta group vta1 processor scripts external-script script1]
user@host# set return-attribute return-attribute

```

Specify an attribute that provides the return value of the script as a valid Java identifier that subsequent actions and event handlers can refer to. If this attribute is not set, the return value is not used by the event. The external script returns the value by printing to standard output.



NOTE: The name of a return attribute cannot start with an underscore (_) because these event attributes are reserved for internal use.

- Related Documentation**
- [Configuring Scripts That Update Accounts \(SRC CLI\) on page 653](#)
 - [Configuring the SRC VTA to Run Scripts on page 663](#)

Configuring SRC VTA Logging (SRC CLI)

The SRC VTA generates messages that you can save in logs—either by writing the messages to text files or by using the system log facilities. Configuring SRC VTA logging involves the following tasks:

- [Logging SRC VTA Messages to a Text File on page 666](#)
- [Logging SRC VTA Messages to a System Logging Server on page 670](#)

Logging SRC VTA Messages to a Text File

Use this procedure to configure the SRC VTA to save messages, such as event messages, SRC VTA startup messages, errors, and so on, in a text file.

Use the following statement to configure the SRC VTA to save messages in text files:

```
shared vta group name logger name file {  
    filename filename;  
    filter filter;  
    maximum-file-size maximum-file-size;  
    rollover-filename rollover-filename;  
}
```

1. From configuration mode, access the statement that configures logging to a text file. For example, to configure a logger called `vta1-logger` for the SRC VTA group called `vta1`:

```
[edit]  
user@host# edit shared vta group vta1 logger vta1-logger file
```

2. Specify the path and filename of the current log file. For example:

```
[edit shared vta group vta1 logger vta1-logger file]  
user@host# set filename pathname/filename.log
```

Make sure you have write access to the folder.

3. (Optional) Specify a filter that determines the type of messages that this log file contains.

```
[edit shared vta group vta1 logger vta1-logger file]  
user@host# set filter filter
```

The filter is specified in an expression. The software filters events by evaluating each subexpression from left to right. When the software finds a match, it logs or ignores the message accordingly. You can specify an unlimited number of subexpressions. The order in which you specify the subexpressions affects the result. Expressions have the format:

singlematch [,singlematch]

Where

singlematch—[!] ([<category> | ([<category>]/[<severity>] |
[<minimumSeverity>]-[<maximumSeverity>]))

- !—Do not log matching events.
- <category>—SRC component that generated the event message. To log only events in a specific category, you can define the category, which is a text string that matches the name of a category. The text string is not case sensitive. For the names of categories, view a log file for a default filter. Juniper Networks Technical Assistance Center (JTAC) can also provide category names.
- [<severity>] | [<minimumSeverity>]-[<maximumSeverity>]—Name or number in the range 1–127. A higher number indicates a higher severity level. [Table 54 on page 667](#) shows common severity levels that you can specify by name.

Table 54: Named Severity Levels

Name	Severity Level
logmin	1
debug	10
info	20
notice	30
warning	40
error	50
crit	60
alert	70
emerg	80
panic	90
logmax	127

You can set up *event trail* logging for the SRC VTA, which is helpful when debugging the SRC VTA configuration. Event trail logging logs messages that show you exactly how the SRC VTA is handling each event as it passes through the SRC VTA. You can follow a particular event as it passes through the SRC VTA and see exactly how each event handler and action handles that event.

To set up event trail logging, set the filter option to **EventTrail/9-**. Enabling event trail logging generates messages such as the following:

```

14:19:59.008 EST 16.11.2011 [Thread-21
(group:HornetQ-client-global-threads-1542419015)] [EventTrail] [9]
Event trail afe51007-7e35-465b-a515-3be12735afd6
VTA received event:
Event type = SAEPluginEvent
Event name = USERSTART
Attributes:
PA_ACCT_INTERIM_TIME = 0
PA_DOMAIN = vtatest.com
PA_EVENT_TIME = 1321471198
PA_EVENT_TIME_MILLISECOND = 1321471198995
PA_INTERFACE_NAME = ip10.227.1.96
PA_INTERFACE_SPEED = 0
PA_IN_OCTETS = 0
PA_IN_PACKETS = 0
PA_LAC_IP = 0
PA_LOGIN_NAME = vtatest1@vtatest.com
PA_NAS_INET_ADDRESS = 0.0.0.0
PA_NAS_IP = 0.0.0.0
PA_NAS_PORT = 0
PA_OUT_OCTETS = 0
PA_OUT_PACKETS = 0
PA_PRIMARY_USER_NAME = vtatest1@vtatest.com
PA_ROUTER_NAME = sim
PA_ROUTER_TYPE = junos
PA_SESSION_ID = KcVGVRM60AwLOAAU
PA_SESSION_TIME = 0
PA_SESSION_TIMEOUT = -1
PA_SSP_HOST = kimberley
PA_TERMINATE_CAUSE = 0
PA_TUNNEL_ID = 0
PA_TUNNEL_SESSION_ID = 0
PA_USER_DN =
uniqueId=vtatest1,ou=group01,retailerName=vtatest,o=Users,o=UMC
PA_USER_INET_ADDRESS = 10.227.1.96
PA_USER_IP_ADDRESS = 10.227.1.96
PA_USER_SESSION_HANDLE = SAE:KcVGVRM60AwLOAAU@sim
PA_USER_SESSION_ID = KcVGVRM60AwLOAAU
PA_USER_TYPE = AUTHINTF
PA_VPN_ID =
currentTime = 1321471199007
processingUuid = afe51007-7e35-465b-a515-3be12735afd6
subscriberId = vtatest1@vtatest.com
SAE information:
SAE references =
[IOR:000000000000003549444C3A736D67742E6A756E697065722E6E657
42F7361652F5365727669636541637469766174696F6E456E67696E653A312E3000000000000000100
000000000006C000102000000000D31302E3232372E362E31323600002261000000146B696D6265726
C65792F736165504F412F5341450000000200000000000000080000000004A4143000000000100000020
0000000000010001000000010501000100010109000000020501000100010100]
NIC proxy namespace = /nicProxies/np1
VTA subscriber ID solution = [Subscriber ID = PA_LOGIN_NAME, NIC key
= PA_LOGIN_NAME, SAE key = LOGIN_NAME]
SAE subscriber ID = LOGIN_NAME
SAE subscriber references: 0
14:19:59.008 EST 16.11.2011 [Thread-21
(group:HornetQ-client-global-threads-1542419015)] [EventTrail] [9]
Event trail afe51007-7e35-465b-a515-3be12735afd6
VTA ignoring event.
.
.

```

```

.
.
14:19:59.039 EST 16.11.2011 [Thread-19
(group:HornetQ-client-global-threads-1542419015)] [EventTrail] [9]
  Event trail cea6f3b3-efc7-46c1-9e8a-b26b721f03cc
  Event handler NoQuota: Received event.
14:19:59.039 EST 16.11.2011 [Thread-19
(group:HornetQ-client-global-threads-1542419015)] [EventTrail] [9]
  Event handler NoQuota: Ignoring event cea6f3b3-efc7-46c1-9e8a-b26b721f03cc
14:19:59.040 EST 16.11.2011 [Thread-19
(group:HornetQ-client-global-threads-1542419015)] [EventTrail] [9]
  Event trail cea6f3b3-efc7-46c1-9e8a-b26b721f03cc
  VTA completed processing event.

```

This is only an excerpt of the log and does not show all messages. Notice that each message starts with a unique ID generated when the SRC VTA starts processing an event—for example, “Event trail afe51007-7e35-465b-a515-3be12735afd6.”

Enabling debug log messages has a negative effect on system performance. Do not enable debug log messages unless JTAC instructs you to do so.

You can define a severity level as follows:

- Specify an explicit severity. For example:
warning—Defines only warning messages
- Specify a minimum severity and a maximum severity. For example:
info-warning—Defines messages of minimum severity level of info and a maximum severity level of warning
- Accept the default minimum (logmin) or maximum (logmax) severity by omitting the minimum or maximum severity. For example:
info—Defines messages of minimum severity level info and maximum severity level logmax
-warning—Defines messages of minimum severity level logmin and maximum severity level warning
- Specify no severity to log all event messages.

- [Table 55 on page 670](#) shows some examples of filters.

Table 55: Examples of Filters for Event Messages

Syntax	Event Messages Saved
/	All event messages
/info-	Event messages of level info and higher from all categories
vta/debug	Debug events from the SRC VTA category only
!vta,/debug	All debug events except those from the SRC VTA category
!VtaMsg/info-,vtaMsg,vta	All messages from the SRC VTA category, except those from the VtaMsg category with level lower than info

- Specify the maximum file size. This option disables or enables and sets the maximum size of the log file and the rollover file.

```
[edit shared vta group vta1 logger vta1-logger file]
user@host# set maximum-file-size maximum-file-size
```



NOTE: The maximum file size is specified in KB. Maximum size of the log file is 10,000,000 KB.

Do not set the maximum file size to a value greater than the available disk space.

- Specify the rollover filename.

```
[edit shared vta group vta1 logger vta1-logger file]
user@host# set rollover filename rollover filename
```

Specify the path and filename of the rollover log file. When the log file reaches the maximum size, the software closes the log file and renames it with the name you specify for the rollover file. If a previous rollover file exists, the software overwrites it. The software then reopens the log file and continues to save event messages in it.

For example:

```
[edit shared vta group vta1 logger vta1-logger file]
user@host# set rollover filename vta_debug.alt
```

Logging SRC VTA Messages to a System Logging Server

Use this procedure to configure the SRC VTA to save messages, such as event messages, SRC VTA startup messages, errors, and so on, on a system logging server.

Use the following statement to configure the SRC VTA to save messages on a system logging server:

```
shared vta group name logger name syslog {
  filter filter;
  host host;
}
```

1. From configuration mode, access the statement that configures logging to the system log facility. For example, to configure a logger called vta1-logger for the SRC VTA group called vta1:

```
[edit]
user@host# edit shared vta group vta1 logger vta1-logger syslog
```

2. (Optional) Specify a filter that determines the type of messages that this log file contains.

```
[edit shared vta group vta1 logger vta1-logger syslog]
user@host# set filter filter
```

The filter is specified in an expression. The software filters events by evaluating each subexpression from left to right. When the software finds a match, it logs or ignores the message accordingly. You can specify an unlimited number of subexpressions. The order in which you specify the subexpressions affects the result. Expressions have the format:

```
singlematch [,singlematch]
```

Where

```
singlematch—[!] ( <category> | ([<category>]/[<severity>] |
[<minimumSeverity>]-[<maximumSeverity>] ))
```

- **!**—Do not log matching events.
- **<category>**—SRC component that generated the event message. To log only events in a specific category, you can define the category, which is a text string that matches the name of a category. The text string is not case sensitive. For the names of categories, view a log file for a default filter. Juniper Networks Technical Assistance Center (JTAC) can also provide category names.
- **[<severity>] | [<minimumSeverity>]-[<maximumSeverity>]**—Name or number in the range 1–127. A higher number indicates a higher severity level.

[Table 56 on page 671](#) shows common severity levels that you can specify by name.

Table 56: Named Severity Levels

Name	Severity Level
logmin	1
debug	10
info	20
notice	30

Table 56: Named Severity Levels (*continued*)

Name	Severity Level
warning	40
error	50
crit	60
alert	70
emerg	80
panic	90
logmax	127

For the SRC VTA, you can perform *event trail* logging, which is helpful when debugging the SRC VTA configuration. Event trail logging logs messages that show you exactly how the SRC VTA is handling each event as it passes through the SRC VTA. You can follow a particular event as it passes through the SRC VTA and see exactly how each event handler and action handles that event.

To set up event trail logging, set the filter option to **filter EventTrail/9-**. Enabling event trail logging generates messages such as the following:

```
14:19:59.008 EST 16.11.2011 [Thread-21
(group:HornetQ-client-global-threads-1542419015)] [EventTrail] [9]
Event trail afe51007-7e35-465b-a515-3be12735afd6
VTA received event:
Event type = SAEPluginEvent
Event name = USERSTART
Attributes:
PA_ACCT_INTERIM_TIME = 0
PA_DOMAIN = vtatest.com
PA_EVENT_TIME = 1321471198
PA_EVENT_TIME_MILLISECOND = 1321471198995
PA_INTERFACE_NAME = ip10.227.1.96
PA_INTERFACE_SPEED = 0
PA_IN_OCTETS = 0
PA_IN_PACKETS = 0
PA_LAC_IP = 0
PA_LOGIN_NAME = vtatest1@vtatest.com
PA_NAS_INET_ADDRESS = 0.0.0.0
PA_NAS_IP = 0.0.0.0
PA_NAS_PORT = 0
PA_OUT_OCTETS = 0
PA_OUT_PACKETS = 0
PA_PRIMARY_USER_NAME = vtatest1@vtatest.com
PA_ROUTER_NAME = sim
PA_ROUTER_TYPE = junos
PA_SESSION_ID = KcVGVRM60AwL0AAU
PA_SESSION_TIME = 0
PA_SESSION_TIMEOUT = -1
```



```

PA_SSP_HOST = kimberley
PA_TERMINATE_CAUSE = 0
PA_TUNNEL_ID = 0
PA_TUNNEL_SESSION_ID = 0
PA_USER_DN =
uniqueId=vtatest1,ou=group01,retailerName=vtatest,o=Users,o=UMC
PA_USER_INET_ADDRESS = 10.227.1.96
PA_USER_IP_ADDRESS = 10.227.1.96
PA_USER_SESSION_HANDLE = SAE:KcVGVRM60AwLOAAU@sim
PA_USER_SESSION_ID = KcVGVRM60AwLOAAU
PA_USER_TYPE = AUTHINTF
PA_VPN_ID =
currentTime = 1321471199007
processingUuid = afe51007-7e35-465b-a515-3be12735afd6
subscriberId = vtatest1@vtatest.com
SAE information:
  SAE references =
[IOR:00000000000003549444C3A736D67742E6A756E697065722E6E657
42F7361652F5365727669636541637469766174696F6E456E67696E653A312E3000000000000000100
000000000006C000102000000000D31302E3232372E362E31323600002261000000146B696D6265726
C65792F736165504F412F5341450000000200000000000008000000004A414300000000100000020
000000000010001000000010501000100010109000000020501000100010100]
  NIC proxy namespace = /nicProxies/np1
  VTA subscriber ID solution = [Subscriber ID = PA_LOGIN_NAME, NIC key
= PA_LOGIN_NAME, SAE key = LOGIN_NAME]
  SAE subscriber ID = LOGIN_NAME
  SAE subscriber references: 0
14:19:59.008 EST 16.11.2011 [Thread-21
(group:HornetQ-client-global-threads-1542419015)] [EventTrail] [9]
Event trail afe51007-7e35-465b-a515-3be12735afd6
VTA ignoring event.
.
.
.
.
14:19:59.039 EST 16.11.2011 [Thread-19
(group:HornetQ-client-global-threads-1542419015)] [EventTrail] [9]
Event trail cea6f3b3-efc7-46c1-9e8a-b26b721f03cc
Event handler NoQuota: Received event.
14:19:59.039 EST 16.11.2011 [Thread-19
(group:HornetQ-client-global-threads-1542419015)] [EventTrail] [9]
Event handler NoQuota: Ignoring event cea6f3b3-efc7-46c1-9e8a-b26b721f03cc
14:19:59.040 EST 16.11.2011 [Thread-19
(group:HornetQ-client-global-threads-1542419015)] [EventTrail] [9]
Event trail cea6f3b3-efc7-46c1-9e8a-b26b721f03cc
VTA completed processing event.

```

This is only an excerpt of the log and does not show all messages. Notice that each message starts with a unique ID generated when the SRC VTA starts processing an event—for example, “Event trail afe51007-7e35-465b-a515-3be12735afd6.”

Enabling debug log messages has a negative effect on system performance. Do not enable debug log messages unless JTAC instructs you to do so.

You can define a severity level as follows:

- Specify an explicit severity. For example:
warning—Defines only warning messages
- Specify a minimum severity and a maximum severity. For example:

info-warning—Defines messages of minimum severity level of info and a maximum severity level of warning

- Accept the default minimum (logmin) or maximum (logmax) severity by omitting the minimum or maximum severity. For example:

info—Defines messages of minimum severity level info and maximum severity level logmax

-warning—Defines messages of minimum severity level logmin and maximum severity level warning

- Specify no severity to log all event messages.
- Example—[Table 57 on page 674](#) shows some examples of filters.

Table 57: Examples of Filters for Event Messages

Syntax	Event Messages Saved
/	All event messages
/info-	Event messages of level info and higher from all categories
vta/debug	Debug events from the SRC VTA category only
!vta,/debug	All debug events except those from the SRC VTA category
!VtaMsg/info-,vtaMsg,vta	All messages from the SRC VTA category, except those from the VtaMsg category with level lower than info

3. Specify the host information for the system log server.

```
[edit shared vta group vta1 logger vta1-logger syslog]
user@host# set host host
```

Specify an IP address or name of a host that collects event messages with a standard system logging daemon.

Related Documentation

- [Logging SRC VTA Messages to a System Logging Server on page 670](#)
- [SRC VTA Overview on page 593](#)
- [How the SRC VTA Works on page 596](#)
- [Logging SRC VTA Messages to a Text File on page 666](#)

Enabling, Disabling, and Restarting the SRC VTA (SRC CLI)

After you create and commit an SRC VTA shared configuration, a new top-level SRC component called *vta-name* automatically appears in the SRC CLI. This component is visible when you execute the following SRC CLI operational commands:

- **show component**
- **enable component vta-name**
- **disable component vta-name**
- **restart component vta-name**

Where the **vta-name** is the name you specified for the group name under the **edit shared vta group name** configuration statement. You can configure multiple SRC VTA instances (groups), each with a unique name.

After you delete an SRC VTA group and commit the configuration, the SRC component called **vta-name** automatically disappears from the SRC CLI.

When you issue the **show component** command, a component called VTA is always visible. However, this component cannot be enabled or disabled or restarted. It is only displayed so that you can view the version number of the SRC VTA. Only SRC VTA group components with names like **vta-name** can be enabled and disabled and restarted.

To enable an SRC VTA:

- From operational mode, enable the SRC VTA.
`user@host> enable component vta-name`

To disable an SRC VTA:

- From operational mode, enable the SRC VTA.
`user@host> disable component vta-name`

To restart an SRC VTA:

- From operational mode, enable the SRC VTA.
`user@host> restart component vta-name`

**Related
Documentation**

- [SRC VTA Overview on page 593](#)
- [How the SRC VTA Works on page 596](#)

Enabling the SOAP Interface for an SRC VTA Group (SRC CLI)

To use the SOAP interface, you need to enable it for the SRC VTA group. You can also specify whether basic HTTP authentication is required when using the SOAP interface. By default, authentication is required. If you disable authentication, the CLI editing level must be set to expert.



NOTE: If you enable authentication, you must configure each user within the shared application server and set the user account role to *VTA-group name*.

After you install the SRC VTA, you can access a Web Services Definition Language (WSDL) file for the application. The WSDL file defines the SOAP properties that you or your customers can use to develop a client. The URL for the SRC VTA WSDL file is:

http://<host-name>:8080/<vta-group>/api/soap/services/vtaAPI?wsdl

- **<host-name>**—Name of the C Series Controller that is running the SRC VTA
- **<vta-group>**—Name of the SRC VTA group

Use the following configuration statements to enable the SOAP interface to configure user accounts at the **[edit]** hierarchy level:

```
shared vta group name api {  
    enable-soap-api;  
    disable-client-authentication  
}
```

1. From configuration mode, access the configuration statement that enables the SOAP interface. For example, to enable the SOAP interface for the SRC VTA group called Quota:

```
user@host# edit shared vta group Quota api
```

2. Enable the SOAP interface.

```
[edit shared vta group Quota api]  
user@host# set enable-soap-api
```

3. To disable client authentication when using the SOAP interface:

```
[edit shared vta group Quota api]  
user@host# set disable-client-authentication
```

Related Documentation

- [SRC VTA SOAP Interface on page 616](#)
- [Viewing SRC VTA SOAP API Statistics \(SRC CLI\) on page 709](#)
- [Methods for the SRC Volume Tracking Application SOAP Interface on page 677](#)
- [Configuring User Accounts for Web Applications \(SRC CLI\) on page 627](#)

Methods for the SRC Volume Tracking Application SOAP Interface

This topic describes the methods associated with the Volume Tracking Application SOAP interface and provides information additional to that in the Web Services Definition Language (WSDL) file.

openAccount

- Open an account and optionally create a balance change for the account as part of one database transaction.
- Arguments:
 - `accountData`—Account data object carrying account information.
 - `writeBalanceChange`—If true, an account balance change record is written to the database.
 - `description`—Description of the account balance change record.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

closeAccount

- Close an account and optionally record the change to the balance change table as part of one database transaction. This method only sets the status of the account to closed. If the account is already closed, do nothing.
- Arguments:
 - `subscriberID`—ID of the subscriber to whom the account belongs.
 - `accountName`—Name of the account to be closed.
 - `writeBalanceChange`—If true, an account balance change record is written to the database.
 - `description`—Description of the account balance change record.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

closeAccountsOfSubscriber

- Close accounts for a subscriber and optionally create balance changes for the accounts as part of one database transaction. This method only sets the status of the accounts to closed. Do nothing for closed accounts.
- Arguments:
 - `subscriberID`—ID of the subscriber to whom the account belongs.
 - `writeBalanceChange`—If true, an account balance change record is written to the database.
 - `description`—Description of the account balance change record.

- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

closeAccountsByName

- Close the named accounts for all subscribers. This method only sets the status of the accounts to closed. Do nothing for closed accounts.
- Arguments:
 - `accountName`—Name of the account to be closed.
 - `writeBalanceChange`—If true, an account balance change record is written to the database.
 - `description`—Description of the account balance change record.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

changeStatus

- Change the status of a specified account and optionally create a balance change record as part of one database transaction. This method does not perform any status check. It can be used to set a closed account to active again.
- Arguments:
 - `subscriberID`—ID of the subscriber to whom the account belongs.
 - `accountName`—Name of the account.
 - `status`—New status of the account.
 - `writeBalanceChange`—If true, an account balance change record is written to the database.
 - `description`—Description of the account balance change record.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

changeBalance

- Add the given amount to the balance (decreasing it if the amount is negative) and optionally create a balance change record for the account as part of one database transaction.
- Arguments:
 - `subscriberID`—ID of the subscriber to whom the account belongs.
 - `accountName`—Name of the account.
 - `status`—New status of the account.
 - `writeBalanceChange`—If true, an account balance change record is written to the database.
 - `description`—Description of the account balance change record.

- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

topUpBalance

- Top up the named accounts with the given amount for all subscribers. This method optionally creates balance change records for the accounts as part of one database transaction. Do nothing for closed accounts.
- Arguments:
 - subscriberID—ID of the subscriber to be topped up.
 - accountName—Name of the account to be topped up.
 - amount—Amount to be added to the balance of each account.
 - date—Date that the account's lastupdatetime field, in milliseconds since January 1, 1970 UTC, is to be set.
 - writeBalanceChange—If true, an account balance change record is written to the database.
 - description—Description of the account balance change record.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

topUpBalance

- Top up the named accounts with the given amount for all subscribers. This method optionally creates balance change records for the accounts as part of one database transaction. Do nothing for closed accounts.
- Arguments:
 - subscriberID—ID of the subscriber to be topped up.
 - accountName—Name of the account to be topped up.
 - status—Status of the account to be topped up.
 - amount—Amount to be added to the balance of each account.
 - date—Date that the account's lastupdatetime field, in milliseconds since January 1, 1970 UTC, is to be set.
 - writeBalanceChange—If true, an account balance change record is written to the database.
 - description—Description of the account balance change record.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

getAccount

- Get the named account for the given subscriber.
- Arguments:

- subscriberID—ID of the subscriber to whom the account belongs.
- accountName—Name of the account.

SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

getAccountsOfSubscriber

- Search for accounts with the given status for a subscriber.
- Arguments:
 - subscriberID—ID of the subscriber to whom the account belongs.
 - status—Account status. If no status is specified, it returns all accounts for the subscriber regardless of their statuses.
- Expected output—Collection of AccountData objects.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

getAccountsByName

- Search for accounts by name and status.
- Arguments:
 - accountName—Name of the account.
 - status—Account status. If no status is specified, it returns all accounts for the subscriber regardless of their statuses.
- Expected output—Collection of AccountData objects.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

getBalanceChanges

- Search for balance changes in a given time period for an account. If both subscriberID and accountName are null, it returns balance changes in a time period for all accounts. If both start and end times are zero, it returns all balance changes for a given account.
- Arguments:
 - subscriberID—ID of the subscriber.
 - accountName—Name of the account.
 - start—Start time of the period to query.
 - end—End time of the period to query.
- Expected output—Collection of BalanceChangeData objects.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

getSessionBalanceChange

- Given a session ID and a qualifier, return the session balance change record for a named account.
- Arguments:
 - subscriberID—ID of the subscriber.
 - accountName—Name of the account.
 - sessionId—ID of the session to query.
 - qualifier—Qualifier of the session to query.
- Expected output—SessionBalanceChangeData object.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

getSessionBalanceChange

- Given a session ID and a qualifier, return the session balance change value objects.
- Arguments:
 - sessionId—ID of the session to query.
 - qualifier—Qualifier of the session to query.
- Expected output—Collection of SessionBalanceChangeData objects.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

getSessionBalanceChanges

- Search for session balance changes in a given time period for an account. If subscriberID and accountName are null, it returns session balance changes in a time period for all accounts. If the start and end times are zero, it returns session balance changes for a given account.
- Arguments:
 - subscriberID—ID of the subscriber.
 - accountName—Name of the account.
 - start—Start time of the period to query.
 - end—End time of the period to query.
- Expected output—Collection of SessionBalanceChangeData objects.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

getSessionBalanceChanges

- Given a session ID and qualifier, get the related session balance changes on the named accounts.
- Arguments:

- sessionId—ID of the session to query.
- qualifier—Qualifier of the session to query.
- subscriberID—ID of the subscriber.
- accountNames—Names of the accounts.
- Expected output—Map from the account name to the SessionBalanceChangeData object. If accountNames is null, it returns the related session balance change records on all accounts for the subscriber.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

getAllBalanceChanges

- Search for balance changes and session balance changes in a given time period for an account. This method converts the balance changes to session balance changes and merges the balance changes and session balance changes.
- Arguments:
 - subscriberID—ID of the subscriber.
 - accountName—Name of the account.
 - start—Start time of the period to query.
 - end—End time of the period to query.
- Expected output—Collection of SessionBalanceChangeData and BalanceChangeData sorted by date field.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

terminateSessions

- Terminate sessions for the subscriber.
- Arguments:
 - subscriberIDs—IDs of the subscribers whose sessions will be terminated.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

getSession

- Given a subscriber ID, session ID, and a qualifier, return the session value object.
- Arguments:
 - subscriberID—ID of the subscriber.
 - sessionId—ID of the session to query.
 - qualifier—Qualifier of the session to query.

- Expected output—SessionData of the session that is found.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

getSessions

- Given the session ID and status, return a collection of session value objects. If no status is specified, it returns sessions with the session ID regardless of their status.
- Arguments:
 - sessionId—ID of the session to query.
 - status—Status of the session to query.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

getSessions

- Get sessions with the given status starting within a given time period for a subscriber. If the start and end times are zero, it returns sessions with the given status for the subscriber. If status is null, it returns all the relevant sessions regardless of their status.
- Arguments:
 - subscriberID—ID of the subscriber.
 - status—Status of the session to query.
 - start—Start time of the period to query.
 - end—End time of the period to query.
- Expected output—Collection of SessionData objects.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

deleteAccount

- Delete an account.
- Arguments:
 - subscriberID—ID of the subscriber.
 - accountName—Name of the account.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

deleteAccountsOfSubscriber

- Delete accounts for a subscriber. Only closed accounts are removed. For each unclosed account, an error message is logged in the debug log.
- Arguments:

- subscriberID—ID of the subscriber.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

deleteAccountsByName

- Delete named accounts for all subscribers. Only closed accounts are deleted. For each unclosed account, an error message is logged in the debug log.
- Arguments:
 - accountName—Name of the account.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

deleteBalanceChanges

- Delete balance changes in a given time period for a set of accounts. If no account names are specified, it deletes balance changes in the given time period on all subscriber accounts. If no time period is specified, it deletes all balance changes for the set of accounts. If no account names or time period are specified, it deletes all balance changes on all subscriber accounts. If both subscriberID and accountNames are not specified, it deletes the balance changes in the given time period for all subscribers and accounts.
- Arguments:
 - subscriberID—ID of the subscriber.
 - accountNames—Names of accounts.
 - start—Start time of the period to query.
 - end—End time of the period to query.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

deleteSessionBalanceChanges

- Delete session balance changes in a given time period for a set of accounts. If no account names are specified, it deletes session balance changes in the given time period on all subscriber accounts. If no time period is specified, it deletes all session balance changes for the set of accounts. If no account names or time period are specified, it deletes all balance changes on all the subscriber accounts.
- Arguments:
 - subscriberID—ID of the subscriber.
 - accountNames—Names of accounts.
 - start—Start time of the period to query.
 - end—End time of the period to query.

- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

deleteSession

- Given a session IdD and qualifier, delete the session.
- Arguments:
 - subscriberID—ID of the subscriber.
 - sessionId—ID of the session to query.
 - qualifier—Qualifier of the session to query.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

deleteSessions

- Given a session ID, delete the related sessions. Only closed sessions are deleted. For each unclosed session, an error message is logged in the debug log.
- Arguments:
 - sessionId—ID of the session to query.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

deleteSessions

- Delete sessions starting within a given time period for a subscriber. If no time period is specified, it deletes all sessions for the subscriber. If no subscriber is specified, it deletes sessions in a given time period for all subscribers. Only closed sessions are deleted.
- Arguments:
 - subscriberID—ID of the subscriber.
 - start—Start time of the period to query.
 - end—End time of the period to query.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

cleanup

- Clean up accounts whose last update time is earlier than the given time and whose status is equal to the given status. If no subscriber is specified, it performs the cleanup for all subscribers. If no status is specified, it only deletes closed accounts.
- Arguments:
 - subscriberID—ID of the subscriber.
 - time—Earliest last update time of accounts to clean up.
 - status—Status of the accounts to clean up.

- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

cleanup

- Clean up sessions, sessionBalanceChanges, and balanceChanges whose last update time is earlier than the given time. If no subscriber is specified, it performs the cleanup for all subscribers.
- Arguments:
 - subscriberID—ID of the subscriber.
 - time—Earliest last update time of accounts to clean up.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

updateAccounts

- Modify accounts selected by subscriberId, accountName, and accountStatus arguments. In those selected accounts, it:
 - Overwrites the account status with newStatus (if newStatus is not null).
 - Overwrites the account balance with balanceAmount (if balanceAmount is not null and amountIsDelta is false).
 - Changes the account balance by balanceAmount (if balanceAmount is not null and amountIsDelta is true).
 - If the balanceChangeDescription is not null, also creates a balance change record containing the specified description, the current time, and the amount by which the account balance was changed.
 - If terminateSessions is true, generates a Callback:TERMINATESESSIONS event for every subscriber associated with the specified accounts.
- Arguments:
 - subscriberID—ID of the subscriber.
 - accountName—Name of the account.
 - accountStatus—Status of the account.
 - newStatus—New status of the account.
 - balanceAmount—Amount to set the balance to.
 - amountIsDelta—If true, sets the balance to the value specified by balanceAmount.
 - balanceChangeDescription—Description of the account balance change record.
 - terminateSessions—If true, generates a Callback:TERMINATESESSIONS event for every subscriber associated with the specified accounts.

- Expected output—Message describing the number of accounts that matched the subscriberId, accountName, and accountStatus arguments, the number of subscribers for those accounts, and the number of accounts that were modified.
- SOAP fault codes—See [“SOAP Fault Codes for the SRC Volume Tracking Application” on page 687](#).

SOAP Fault Codes for the SRC Volume Tracking Application

When the SRC Volume Tracking Application (VTA) receives a Simple Object Access Protocol (SOAP) request that it cannot handle, it returns a SOAP fault message to the gateway client. This message contains a text string that specifies the SOAP fault code and a text string that provides additional information about the fault code.

[Table 58 on page 688](#) lists SOAP fault codes that the SRC VTA returns to the gateway client.

Table 58: SRC VTA Methods and SOAP Fault Codes

Method Name	Fault Code	Fault Description	Call Classification	Action	Notes
open Account	soapenv:Server.userException	java.lang.IllegalArgumentException: To open an account, an account value object is needed.	Failed	Verify that you passed appropriate values (as specified in the Web Services Description Language [WSDL] file) in the AccountData object.	This exception occurs when the AccountData object is empty.
		Rollback failed:	Failed	Contact the gateway administrator.	This exception occurs when the database connection has timed out.
		Transaction failed	Failed	Contact the gateway administrator.	This exception occurs because the balance parameter is not set in the AccountData object or the database was down when the system was processing the transaction.
		javax.ejb.DuplicateKeyException: Entity with primary key [.jane2@virneo.net..] already exists	Failed	Verify that the combination of the subscriberID and accountName is unique because it is the primary key.	Indicates that the client attempted to insert a name or subscriberID that is already in the database.
		javax.ejb.CreateException: Error checking if entity exists:org.jboss.util.NestedSQLException: Unable to get managed connection for vta-Quota/datasource; - nested throwable: (javax.resource.ResourceException: Unable to get managed connection for vta-Quota/datasource)	Failed	Contact the gateway administrator.	Indicates that the database is not running.
			Failed		

Table 58: SRC VTA Methods and SOAP Fault Codes (*continued*)

Method Name	Fault Code	Fault Description	Call Classification	Action	Notes
		javax.ejb.Create Exception: Could not create entity: com.mysql. jdbc.exceptions.jdbc4. MySQLIntegrityConstraint ViolationException: Column 'LAST_UPDATE_TIME' cannot be null		Verify that the value of the last updated time is not null.	
		javax.ejb.EJBException: Failed to update database. The transaction was rolled back. Abandoning unprocessed event: net.juniper.smgmt.vta. quotadp.UpdateFailed Exception: Db-engine failed to process event: javax.ejb.Transaction RolledbackLocal Exception: To get an account, a subscriberID is needed.	Failed	Contact the gateway administrator.	Indicates that the subscriberID was missing in the AccountData object when the client was creating an account.

Table 58: SRC VTA Methods and SOAP Fault Codes (*continued*)

Method Name	Fault Code	Fault Description	Call Classification	Action	Notes
change Balance	soapenv: Server.user Exception	java.lang.IllegalArgumentException Exception: To change the balance of an account, a subscriberID is needed	Failed	Verify and pass the existing subscriberID and accountName to update the balance of an account.	
		javax.ejb.ObjectNotFoundException: No such entity!	Failed	Verify and pass the existing or valid subscriberID and accountName.	
		net.juniper.smgmt.vta.data mgr.im.ChangeAccount Exception: Cannot change balance for closed account	Failed	Contact the gateway administrator.	Indicates that the account is already closed.
		javax.ejb.DuplicateKeyException: Create balance change failed. DuplicateKeyException: subscriberID= Test accountName= BoughtQuota date= 1363847398695	Failed	Contact the gateway administrator.	This exception occurs when an entity EJB object cannot be created because an object with the same key already exists.
		Rollback failed:	Failed	Contact the gateway administrator.	This exception occurs when the database connection has timed out.
		Transaction failed	Failed	Contact the gateway administrator.	This exception occurs if the database was down when the system was processing the request message.
		javax.ejb.EJBException: Failed to update database. The transaction was rolled back. Abandoning unprocessed event: net.juniper.smgmt.vta.quotadp.UpdateFailed Exception: Db-engine failed to process event: javax.ejb.TransactionRolledbackLocal Exception: To get an account, a subscriberID is needed.	Failed	Contact the gateway administrator.	Indicates that the subscriberID was missing when the client was modifying the account balance.

Table 58: SRC VTA Methods and SOAP Fault Codes (*continued*)

Method Name	Fault Code	Fault Description	Call Classification	Action	Notes
change Status	soapenv:Server.userException	java.lang.IllegalArgumentException:To change the status of an account, a subscriberID/ an accountName object is needed.	Failed	Verify and pass the existing accountName and subscriberID.	
		Rollback failed:	Failed	Contact the gateway administrator.	This exception occurs when the database connection has timed out.
		Transaction failed	Failed	Contact the gateway administrator.	This exception occurs if the database was down when the system was processing the request message.
		javax.ejb.ObjectNotFoundException: No such entity!	Failed	Verify and pass the existing or valid subscriberID and accountName.	
cleanUp	soapenv:Server.userException	net.juniper.smgmt.vta.data mgr.im.DeleteSession Exception : The following closed sessions could not be deleted because they have not subsequently been stopped (format is 'subscriberId:sessionId: sessionIdQualifier'):	Failed	Verify that the session is already closed or stopped.	
		net.juniper.smgmt.vta.data mgr.im.DeleteSession Exception : The following sessions could not be deleted because they are neither closed nor stopped (format is 'subscriberId:sessionId: sessionIdQualifier= status'):	Failed	Verify that the session is already closed or stopped.	

Table 58: SRC VTA Methods and SOAP Fault Codes (*continued*)

Method Name	Fault Code	Fault Description	Call Classification	Action	Notes
close Account	soapenv:Server.userException	java.lang.IllegalArgumentException Exception: To change an account, a subscriberID/ an accountName object is needed.	Failed	Verify and pass the existing accountName and subscriberID.	
		javax.ejb.ObjectNotFoundException Exception: No such entity!	Failed	Verify and pass the existing or valid subscriberID and accountName.	
		Internally calling "changeStatus" Method Name	Failed		See the fault codes for the change Status method for more information.
close Account ByName	soapenv:Server.userException	Internally calling "changeStatus" Method Name	Failed		See the fault codes for the change Status method for more information.
close Account OfSubscriber	soapenv:Server.userException	java.lang.IllegalArgumentException Exception: To close accounts for a subscriber, a subscriberID is needed	Failed	Verify and pass the existing accountName and subscriberID.	
		Internally calling "changeStatus" Method Name	Failed		See the fault codes for the change Status method for more information.
delete Account	soapenv:Server.userException	java.lang.IllegalArgumentException Exception: To change an account, a subscriberID/ an accountName object is needed.	Failed	Verify and pass the existing accountName and subscriberID.	
		javax.ejb.ObjectNotFoundException Exception: No such entity!	Failed	Verify and pass the existing or valid subscriberID and accountName.	
		net.juniper.smgmt.vta.data mgr.im.DeleteAccount Exception: Cannot delete unclosed account.vta-Quota/AccountLocal: [.JJ.ACC.]	Failed	Close the account before deleting it.	

Table 58: SRC VTA Methods and SOAP Fault Codes (*continued*)

Method Name	Fault Code	Fault Description	Call Classification	Action	Notes
delete Accounts ByName	soapenv: Server.user Exception	java.lang.IllegalArgumentException Exception: To delete accounts by accountName, an accountName is needed	Failed	Verify and pass the existing accountName to delete an account.	
delete Accounts OfSubscriber	soapenv: Server.user Exception	java.lang.IllegalArgumentException Exception: To delete accounts by subscriber, a subscriberID is needed	Failed	Verify and pass the existing subscriberID to delete an account.	
delete Balance Changes	soapenv: Server.user Exception	java.lang.IllegalArgumentException Exception: To delete balance changes, start time cannot be greater than end time.	Failed	Verify that the start time is not later than the end time.	
		java.lang.IllegalArgumentException Exception: To delete balance changes, account or time period should be specified.	Failed	Verify and pass the existing accountName or time period to delete the balance of an account.	
		java.lang.IllegalArgumentException Exception: To delete balance changes, When account names are specified, subscriberID cannot be null.	Failed	Verify and pass the valid subscriberID when the account names are specified.	
delete Session	soapenv: Server.user Exception	java.lang.IllegalArgumentException Exception: To delete a session, a subscriberID is needed	Failed	Verify and pass the existing subscriberID to delete the session.	
		java.lang.IllegalArgumentException Exception: To delete a session, a sessionId is needed	Failed	Verify and pass the existing sessionId to delete the session.	
		net.juniper.smgmt.vta.data mgr.im.DeleteSession Exception : The following sessions could not be deleted because they are neither closed nor stopped (format is 'subscriberId:sessionId:sessionIdQualifier=status'):	Failed	Verify that the sessions are neither closed nor stopped before deleting them.	

Table 58: SRC VTA Methods and SOAP Fault Codes (*continued*)

Method Name	Fault Code	Fault Description	Call Classification	Action	Notes
delete Session Balance Changes	soapenv:Server.userException	java.lang.IllegalArgumentException Exception: To delete session balance changes, start time cannot be greater than end time.	Failed	Verify that the start time is not later than the end time.	
		java.lang.IllegalArgumentException Exception: To delete session balance changes, an account or time period should be specified.	Failed	Verify and pass the existing accountName or time period.	
		java.lang.IllegalArgumentException Exception: To delete session balance changes, when account names are specified, subscriberID cannot be null.	Failed	Verify and pass the valid subscriberID when the account names are specified.	
delete Sessions	soapenv:Server.userException	java.lang.IllegalArgumentException Exception: To delete sessions by sessionId, a sessionId is needed.	Failed	Verify that the valid sessionId was passed to delete the closed sessions.	
		net.juniper.smgmt.vta.data mgr.im.DeleteSession Exception : The following sessions could not be deleted because they are neither closed nor stopped (format is 'subscriberId:sessionId:sessionIdQualifier=status'):	Failed	Verify that the sessions are neither closed nor stopped before deleting them.	
		java.lang.IllegalArgumentException Exception: To delete session balance changes, start time cannot be greater than end time.	Failed	Verify that the start time is not later than the end time.	
		java.lang.IllegalArgumentException Exception: To delete sessions, a subscriberID or a time period should be specified.	Failed	Verify and pass the existing subscriberID or time period.	
get Account	soapenv:Server.userException	java.lang.IllegalArgumentException Exception: To get an account, a subscriberID/ accountName is needed.	Failed	Verify and pass the existing or valid subscriberID and accountName.	

Table 58: SRC VTA Methods and SOAP Fault Codes (*continued*)

Method Name	Fault Code	Fault Description	Call Classification	Action	Notes
getAccountsByName	soapenv:Server.userException	java.lang.IllegalArgumentException Exception: To get an account, an accountName is needed.	Failed	Verify and pass the existing or valid accountName.	
getAccountsOfSubscriber	soapenv:Server.userException	java.lang.IllegalArgumentException Exception: To get accounts of a subscriber, a subscriberID is needed	Failed	Verify and pass the existing or valid subscriberID.	
getAllBalanceChanges	soapenv:Server.userException	java.lang.IllegalArgumentException Exception: To get balance changes, subscriberID and accountName should both be null or not null.	Failed	Verify that both the subscriberID and accountName are null or not null.	
		java.lang.IllegalArgumentException Exception: To get balance changes, start time cannot greater than end time.	Failed	Verify that the start time is not later than the end time.	
		java.lang.IllegalArgumentException Exception: To get balance changes, an account or a time period should be specified.	Failed	Verify and pass the existing or valid accountName or time period to get the balance changes.	
getBalanceChanges	soapenv:Server.userException	Internally calling "getAllBalanceChanges" Method name	Failed		See the fault codes for the getAllBalanceChanges method for more information.
getSession	soapenv:Server.userException	javax.ejb.ObjectNotFoundException Exception: No such entity!	Failed	Verify and pass a valid accountName to get the session details.	
		java.lang.IllegalArgumentException Exception: To change an account, a subscriberID/ an accountName object is needed.	Failed	Verify and pass the existing or valid subscriberID and accountName to get the session details.	

Table 58: SRC VTA Methods and SOAP Fault Codes (*continued*)

Method Name	Fault Code	Fault Description	Call Classification	Action	Notes
get Session Balance Change	soapenv: Server.user Exception	java.lang.IllegalArgumentException: To get a specific session balance change, a subscriberID/ an accountName /a sessionId object is needed.	Failed	Verify and pass a valid subscriberID or accountName or sessionId to get the session balance change details.	
		java.lang.IllegalArgumentException: To get session balance changes by a session, a sessionId is needed.	Failed	Verify and pass the existing or valid sessionId to get the session balance change details.	
get Session Balance Changes	soapenv: Server.user Exception	Internally calling "getAllBalanceChanges" Method name	Failed		See the fault codes for the getAllBalanceChanges method for more information.
get Sessions	soapenv: Server.user Exception	java.lang.IllegalArgumentException: To get sessions, a sessionId is needed.	Failed	Verify and pass the existing or valid sessionId.	
		java.lang.IllegalArgumentException: To get sessions for a subscriber, a subscriberID is needed.	Failed	Verify and pass the existing or valid subscriberID.	
		java.lang.IllegalArgumentException: To get sessions, start time cannot greater than end time.	Failed	Verify that the start time is not later than the end time.	
terminate Sessions	soapenv: Server.user Exception	javax.ejb.EJBException: Failed to update database. The transaction was rolled back. Abandoning unprocessed event: net.juniper.smgmt.vta.quotadp.UpdateFailed Exception: Db-engine failed to process event: javax.ejb.TransactionRolledbackLocalException: To get an account, a subscriberID is needed.	Failed	Verify that the valid subscriberID was passed to terminate the subscriber sessions.	

Table 58: SRC VTA Methods and SOAP Fault Codes (*continued*)

Method Name	Fault Code	Fault Description	Call Classification	Action	Notes
topUp Balance	soapenv:Server.userException	Internally calling "terminateSessions" Method Name	Failed		See the fault code for the terminate Sessions method for more information.
update Accounts	Server.userException	java.lang.IllegalArgumentException: accountName must not be null.	Failed	Verify and pass the existing or valid accountName to update the account.	
		Internally calling "terminateSessions" Method Name	Failed		See the fault code for the terminate Sessions method for more information.

Table 59 on page 697 lists fault codes that are common to the methods associated with the SRC Volume Tracking Application SOAP interface.

Table 59: VTA SOAP Fault Codes Common to Methods

Fault Code	Call Classification	Action	Notes
javax.ejb.RemoveException	Failed	Contact the gateway administrator.	This exception occurs if the enterprise bean or container does not allow you to remove the EJB object. For more details, see http://docs.oracle.com/javaee/6/api/javax/ejb/RemoveException.html .
Server.userException	Failed	Verify that you passed correct arguments in the API.	This exception occurs when the arguments passed in the API does not match the actual arguments in the API.

Table 59: VTA SOAP Fault Codes Common to Methods (*continued*)

javax.ejb.Finder Exception	Failed	Contact the gateway administrator.	Indicates that the client failed to find the requested EJB object. For more details, see http://docs.oracle.com/javaee/6/api/javax/ejb/FinderException.html .
-------------------------------	--------	---------------------------------------	--

Related Documentation • [Methods for the SRC Volume Tracking Application SOAP Interface on page 677](#)

Using One SRC VTA Account for Multiple Subscriber Sessions

The SRC VTA allows multiple subscriber sessions to share the same SRC VTA account. The SRC VTA debits usage for all the subscriber sessions from the account. When the account is empty, service sessions for all subscribers are stopped. When the account is refilled, the SRC VTA starts services for all subscriber sessions that share the account.

To use this feature, you use the subscriberId event attribute to map a group of subscribers to the SRC VTA account. You then use the accounting-id attribute as the **subscriber-id-solution** parameter under the **shared vta group name** configuration statement. For information about this option, see “[Creating and Configuring an SRC VTA Shared Group Configuration \(SRC CLI\)](#)” on page 639. You also set up the NIC to use the accounting ID to look up the SAE that manages a subscriber.

To set up the SRC software to use one SRC VTA account for multiple subscribers:

1. In the subscriber classifier script, assign a value to the accountingUserId attribute. For example, you could assign it to the userName, interfaceName, loginName, or a combination of classification criteria. The purpose of the assignment is to allow the SRC VTA to identify subscribers by many different subscriber attributes using accountingUserId as a wrapper.

For example, the following subscriber classifier script assigns the value of the userName to the accountingUserId attribute:

```
[<-retailerDn->?accountingUserId=<-userName->?sub?(uniqueID=<-userName->)]
```

2. Configure the SAE to publish the PA_ACCOUNTING_ID plug-in attribute in subscriber-tracking events to the NIC SAE agent plug-in.

See *Configuring Internal Plug-Ins (SRC CLI)*.

3. Configure the NIC to use the OnePopAcctId NIC scenario.

See “[Configuring the NIC \(SRC CLI\)](#)” on page 152.

4. Configure the SRC VTA shared group configuration in the SRC CLI as follows:

- For the **subscriber-id-solution** option enter **accounting-id**.

- For the **nic-proxy** option, enter the name of the NIC proxy configuration you entered for the **shared vta nic-proxy-configuration name** statement. Be sure to configure the NIC proxy so that it uses a NIC that maps from the accounting ID to the SAE.

See [“Creating and Configuring an SRC VTA Shared Group Configuration \(SRC CLI\)” on page 639](#).

5. (Optional) Set up an action to apply functions to all subscriber sessions that share the same SRC VTA account. For example, the following action starts services for all subscriber sessions that have the same subscriber ID.

```
[edit shared vta group vta1]
user@host# edit action xyz
[edit shared vta group vta1 action xyz]
user@host#y# set function sae-start-service
[edit shared vta group pear action xyz]
user@host## show
```

```
function sae-start-service;
parameter;
```

Make sure that the **current-subscriber-only** option is set to false (not present). If it is set, delete it.

Related Documentation

- [SRC VTA Overview on page 593](#)
- [Managing SRC VTA Accounts and Sessions on page 614](#)
- [Configuring Subscribers and Subscriptions to SRC VTA Services on page 635](#)

Managing the SRC VTA (SRC CLI)

Each SRC VTA account has a status. You can configure the initial status of each SRC VTA account type in the db-engine processor.



NOTE: The SRC VTA requires accounts to have the correct status to execute delete and modify commands. For example, to delete an account, the status of the account must be closed. If the account has any other status, the SRC VTA will not delete the account. Conversely, to modify an account, the status of the account cannot be closed. For example, to change the balance of an account, the status of the account cannot be closed. If the account status is closed, the SRC VTA will not execute the change to the account.

- [Deleting Balance Change History Records from the Database \(SRC CLI\) on page 701](#)
- [Deleting Session History Records from the Database \(SRC CLI\) on page 702](#)
- [Deleting Subscriber SRC VTA Accounts \(SRC CLI\) on page 702](#)
- [Modifying SRC VTA Accounts and Service Sessions \(SRC CLI\) on page 702](#)
- [Terminating Sessions \(SRC CLI\) on page 704](#)
- [Deleting Event Queues for SRC VTA Groups \(SRC CLI\) on page 704](#)

Deleting Balance Change History Records from the Database (SRC CLI)

Use the **request vta group vtaName delete balance-changes before date** command to delete old balance change records from the SRC VTA database. This command deletes balance change records from the SRC VTA database that have a last updated time before the specified date.

- From operational mode, execute:

```
user@host> request vta group vtaName delete balance-changes before date
```

Specify the date in the format yyyy-mm-dd.

- Related Documentation**
- [Deleting Session History Records from the Database \(SRC CLI\) on page 702](#)
 - [Deleting Subscriber SRC VTA Accounts \(SRC CLI\) on page 702](#)

Deleting Session History Records from the Database (SRC CLI)

Use the **request vta group *vtaName* delete sessions before *date*** command to delete old session history records from the SRC VTA database. This command deletes session history records from the SRC VTA database that have a last updated time before the specified date.

- From operational mode, execute:

```
user@host> request vta group vtaName delete sessions before date
```

Specify the date in the format yyyy-mm-dd.

Related Documentation

- [Deleting Balance Change History Records from the Database \(SRC CLI\) on page 701](#)
- [Deleting Subscriber SRC VTA Accounts \(SRC CLI\) on page 702](#)

Deleting Subscriber SRC VTA Accounts (SRC CLI)

Use the **request vta group *vtaName* delete subscriber *subscriber-id*** command to delete a specified subscriber's SRC VTA accounts and subscriber's sessions that have a status of "Closed." This command also deletes all balance changes (from administrative actions) and session balance changes (from session usage) associated with those accounts.

- From operational mode, execute:

```
user@host> request vta group vtaName delete subscriber subscriber-id
```

Related Documentation

- [Deleting Balance Change History Records from the Database \(SRC CLI\) on page 701](#)
- [Deleting Session History Records from the Database \(SRC CLI\) on page 702](#)

Modifying SRC VTA Accounts and Service Sessions (SRC CLI)

Use the **request vta group *vtaName* update-accounts *account-name* *account-name*** command to modify subscriber SRC VTA accounts and service sessions. By using this command, you can modify either all service sessions in an account or a specific subscriber's service sessions in the account.

You can overwrite the status, description, last update time, or balance in a set of accounts. You can also change the existing account balance by a specific positive or negative amount.

- To modify all service sessions in an account, execute the following command from operational mode:

```
user@host> request vta group vtaName update-accounts account-name account-name
```

- To modify a specific subscriber's service sessions in the account, execute the following command from operational mode:

```
user@host> request vta group vtaName update-accounts account-name account-name
subscriber-id subscriber-id
```

You can optionally add the **account-status** option if you only want to update accounts with a particular status.

Use one or more of the options described in [Table 60 on page 703](#) to modify the account.

Table 60: Arguments Used to Modify Accounts

Argument	Description
new-status <i>newAccountStatus</i>	Changes the status of the account or specified subscriber's service sessions.
new-balance <i>newAccountBalance</i>	Sets the account balance to the amount specified by the <i>newAccountBalance</i> . For example, an account starts with a balance of 10. You set the <i>newAccountBalance</i> to 2. The account now has a balance of 2.
balance-change <i>accountBalanceChangeAmount</i>	Adds the amount specified for the <i>accountBalanceChangeAmount</i> to the existing account balance. For example, an account starts with a balance of 10. You set the <i>accountBalanceChangeAmount</i> to 2. The account now has a balance of 12.
balance-change-description <i>description</i>	(Optional) Add a description for the balance change. If you specify this option, a new balance change record is created in the database that contains the specified description, as well as the amount by which the account balance was changed. If the account balance was not changed, this amount will be zero.
terminate-sessions	(Optional) Trigger a callback:terminatesessions event for each specified subscriber. The SRC VTA ignores these events unless you configure an event-handler to process them.

When you execute either of these commands, the SRC VTA takes the following steps:

- If you specify the **terminate-sessions** option, the SRC VTA generates a **callback:terminatesessions** event for every subscriber that owns a selected account.
- The SRC VTA updates all service sessions in the account, or all service sessions for the specified subscriber as requested. This is purely a database operation.
- For each account that is updated, the SRC VTA creates an **account-update** event. This event contains the relevant account's old and new balances, and other information. The SRC VTA ignores these events unless you configure an event handler to process this event type.

Related Documentation

- [Managing SRC VTA Accounts and Sessions Overview on page 613](#)
- [SRC VTA Overview on page 593](#)
- [Terminating Sessions \(SRC CLI\) on page 704](#)
- [How the SRC VTA Works on page 596](#)
- [Deleting Subscriber SRC VTA Accounts \(SRC CLI\) on page 702](#)

- [Deleting Session History Records from the Database \(SRC CLI\) on page 702](#)

Terminating Sessions (SRC CLI)

For better integration with billing systems, it is sometimes desirable to terminate a set of subscribers' SRC VTA sessions at the end of a billing period.

- To terminate SRC VTA sessions, from operational mode, execute:

```
user@host> request vta group vtaName terminate-sessions subscriber-id subscriber-id
```



NOTE: To use `request vta group vtaName terminate-sessions subscriber-id subscriber-id` to terminate sessions, you need to configure an event handler to process `callback:terminatesessions` events by invoking an action that in turn invokes the `db-engine-terminate-session` function. If you do not configure the event handler to process these events in this manner, executing this command does nothing.

Related Documentation

- [Managing SRC VTA Accounts and Sessions Overview on page 613](#)
- [SRC VTA Overview on page 593](#)
- [Modifying SRC VTA Accounts and Service Sessions \(SRC CLI\) on page 702](#)
- [How the SRC VTA Works on page 596](#)

Deleting Event Queues for SRC VTA Groups (SRC CLI)

Use the `request vta delete event-queue group vtaName` to delete the event queue for the SRC VTA group. Use this command to delete empty event queues after you disable the SRC VTA using the `disable component vta-xxx` command. You should execute this command on each system where the SRC VTA group was previously running.

- From operational mode, execute:

```
user@host> request vta delete event-queue group vtaName
```

Related Documentation

- [Configuring Event Queues for SRC VTA Groups \(SRC CLI\) on page 650](#)

CHAPTER 40

Monitoring and Testing the SRC VTA

- [Viewing a Subscriber's SRC VTA Accounts \(SRC CLI\) on page 705](#)
- [Viewing Balance Change History for a Subscriber \(SRC CLI\) on page 705](#)
- [Viewing a Subscriber's Session History \(SRC CLI\) on page 706](#)
- [Viewing SRC VTA Performance Statistics \(SRC CLI\) on page 706](#)
- [Viewing SRC VTA SOAP API Statistics \(SRC CLI\) on page 709](#)
- [Testing the SRC VTA Configuration Overview on page 710](#)
- [Testing SRC VTA Events \(SRC CLI\) on page 713](#)

Viewing a Subscriber's SRC VTA Accounts (SRC CLI)

Purpose View the list of all specified SRC VTA accounts of a subscriber.

Action To view information about a subscriber's SRC VTA account:

```
user@host> show vta accounts group name subscriber-id subscriberId account-name  
account-name
```

The **group** and **subscriber-id** options are mandatory. The **account-name** option is optional.

Related Documentation

- [Viewing Balance Change History for a Subscriber \(SRC CLI\) on page 705](#)
- [Viewing a Subscriber's Session History \(SRC CLI\) on page 706](#)

Viewing Balance Change History for a Subscriber (SRC CLI)

Purpose View a list of the specified SRC VTA accounts and all associated balance changes and session balance changes if the changes have a timestamp between the *from* and *to* dates. Balance changes (from administrative actions) and session balance changes (from service usage) are displayed in a single list and ordered by their timestamps from the oldest to most recent.

Action To view a list of the specified SRC VTA accounts and all associated balance changes and session balance changes, from operational mode, execute:

```
user@host> show vta balance-changes group name subscriber-id subscriber-id account-name  
account-name from from to to
```

Specify the date in the format yyyy-mm-dd.

The system displays both balance changes from administrative actions and session balance changes from service usage. The from and to dates are optional. If you omit the to date, the system defaults to the infinite future. If you omit the from date, the system defaults to the previous day.

- Related Documentation**
- [Viewing a Subscriber's SRC VTA Accounts \(SRC CLI\) on page 705](#)
 - [Viewing a Subscriber's Session History \(SRC CLI\) on page 706](#)

Viewing a Subscriber's Session History (SRC CLI)

Purpose View a subscriber's SRC VTA-tracked session history.

Action To view a list of the subscriber's SRC VTA-tracked session history, from operational mode, execute:

```
user@host> show vta sessions group name from from to to subscriber-id subscriber-id
```

Specify the date in the format yyyy-mm-dd.

The system displays a list of the subscriber's SRC VTA-tracked sessions that have a last update time on or between the from and to dates. The list is ordered by session start time (not last update time), from the oldest to most recent.

The from and to dates are optional. If you omit the to date, the system defaults to the infinite future. If you omit the from date, balance changes are displayed for the last six days. Balance changes are displayed from the oldest to most recent.

To display only the subscriber ID, session ID, status, up bytes, and down bytes, execute:

```
user@host> show vta sessions group name from from to to subscriber-id subscriber-id terse
```

- Related Documentation**
- [Viewing a Subscriber's SRC VTA Accounts \(SRC CLI\) on page 705](#)
 - [Viewing Balance Change History for a Subscriber \(SRC CLI\) on page 705](#)

Viewing SRC VTA Performance Statistics (SRC CLI)

Purpose View performance statistics for a SRC VTA's queue, event handlers, and actions.

Action To view performance statistics for a SRC VTA group, from operational mode, execute:

```
user@host> show vta statistics performance group name
```

```
Uptime
Uptime (seconds) 44
Up Since          Tue Oct 18 14:53:04 EDT 2011

Event Queue
Configured max queue size      5000
Current queue size             0
Events received                 0
```

```

Events rejected due to full queue          0
Events dispatched                         0
Events received in last 60s: Min time in SAE 0
Events received in last 60s: Avg time in SAE 0
Events received in last 60s: Max time in SAE 0
Events dispatched in last 60s: Min time in queue 0
Events dispatched in last 60s: Avg time in queue 0
Events dispatched in last 60s: Max time in queue 0
Events received/second in last 60s        0.0
Events dispatched/second in last 60s      0.0

Event Handler
Event handler name                        EndofBilling
Events received                          0
Events ignored                           0
Events processed                          0
Event processing failures                  0
Successful processing in last 60s: Min time 0
Successful processing in last 60s: Avg time 0
Successful processing in last 60s: Max time 0
Successfully processed events/second in last 60s 0.0

Event Handler
Event handler name                        GetQuota
Events received                          0
Events ignored                           0
Events processed                          0
Event processing failures                  0
Successful processing in last 60s: Min time 0
Successful processing in last 60s: Avg time 0
Successful processing in last 60s: Max time 0
Successfully processed events/second in last 60s 0.0

Event Handler
Event handler name                        NoQuota
Events received                          0
Events ignored                           0
Events processed                          0
Event processing failures                  0
Successful processing in last 60s: Min time 0
Successful processing in last 60s: Avg time 0
Successful processing in last 60s: Max time 0
Successfully processed events/second in last 60s 0.0

Event Handler
Event handler name                        QuotaRefilled
Events received                          0
Events ignored                           0
Events processed                          0
Event processing failures                  0
Successful processing in last 60s: Min time 0
Successful processing in last 60s: Avg time 0
Successful processing in last 60s: Max time 0
Successfully processed events/second in last 60s 0.0

Event Handler
Event handler name                        RecordUsage
Events received                          0
Events ignored                           0
Events processed                          0
Event processing failures                  0

```

Successful processing in last 60s: Min time 0
Successful processing in last 60s: Avg time 0
Successful processing in last 60s: Max time 0
Successfully processed events/second in last 60s 0.0

Event Handler
Event handler name SetInterim
Events received 0
Events ignored 0
Events processed 0
Event processing failures 0
Successful processing in last 60s: Min time 0
Successful processing in last 60s: Avg time 0
Successful processing in last 60s: Max time 0
Successfully processed events/second in last 60s 0.0

Action
Action name CalcUsage
Events received 0
Events processed 0
Event processing failures 0
Successful processing in last 60s: Min time 0
Successful processing in last 60s: Avg time 0
Successful processing in last 60s: Max time 0
Successfully processed events/second in last 60s 0.00

Action
Action name CalculateInterim
Events received 0
Events processed 0
Event processing failures 0
Successful processing in last 60s: Min time 0
Successful processing in last 60s: Avg time 0
Successful processing in last 60s: Max time 0
Successfully processed events/second in last 60s 0.00

Action
Action name DebitAccounts
Events received 0
Events processed 0
Event processing failures 0
Successful processing in last 60s: Min time 0
Successful processing in last 60s: Avg time 0
Successful processing in last 60s: Max time 0
Successfully processed events/second in last 60s 0.00

Action
Action name GetAccountBalances
Events received 0
Events processed 0
Event processing failures 0
Successful processing in last 60s: Min time 0
Successful processing in last 60s: Avg time 0
Successful processing in last 60s: Max time 0
Successfully processed events/second in last 60s 0.00

Action
Action name SetInterim
Events received 0
Events processed 0
Event processing failures 0

```

Successful processing in last 60s: Min time      0
Successful processing in last 60s: Avg time     0
Successful processing in last 60s: Max time     0
Successfully processed events/second in last 60s 0.00

Action
Action name                                     StartptspService
Events received                                0
Events processed                               0
Event processing failures                      0
Successful processing in last 60s: Min time     0
Successful processing in last 60s: Avg time     0
Successful processing in last 60s: Max time     0
Successfully processed events/second in last 60s 0.00

Action
Action name                                     StopptspService
Events received                                0
Events processed                               0
Event processing failures                      0
Successful processing in last 60s: Min time     0
Successful processing in last 60s: Avg time     0
Successful processing in last 60s: Max time     0
Successfully processed events/second in last 60s 0.00

Action
Action name                                     TerminateSession
Events received                                0
Events processed                               0
Event processing failures                      0
Successful processing in last 60s: Min time     0
Successful processing in last 60s: Avg time     0
Successful processing in last 60s: Max time     0
Successfully processed events/second in last 60s 0.00

Action
Action name                                     test
Events received                                0
Events processed                               0
Event processing failures                      0
Successful processing in last 60s: Min time     0
Successful processing in last 60s: Avg time     0
Successful processing in last 60s: Max time     0
Successfully processed events/second in last 60s 0.00

user@host>

```

- Related Documentation**
- [Viewing a Subscriber's SRC VTA Accounts \(SRC CLI\) on page 705](#)
 - [Viewing Balance Change History for a Subscriber \(SRC CLI\) on page 705](#)

Viewing SRC VTA SOAP API Statistics (SRC CLI)

Purpose View statistics for the SRC VTA SOAP API.

Action To view statistics for the SRC VTA SOAP API, from operational mode, execute:

```
user@host> show vta statistics soap-api group name
```

For example, to view statistics for the SRC VTA group called Quota:

```
user@host> show vta statistics soap-api group Quota
SOAP API Statistics
State                The vta group Quota is running
Start time
Number of errors      0
Number of soap resssquests 13282

user@host>
```

**Related
Documentation**

- [SRC VTA SOAP Interface on page 616](#)
- [Enabling the SOAP Interface for an SRC VTA Group \(SRC CLI\) on page 676](#)
- [Methods for the SRC Volume Tracking Application SOAP Interface on page 677](#)
- [Viewing a Subscriber's SRC VTA Accounts \(SRC CLI\) on page 705](#)

Testing the SRC VTA Configuration Overview

You can use SRC VTA test commands to simulate events and test a specific SRC VTA group configuration. You can simulate subscriber-tracking events, service-tracking events, and callback events.

There are two steps involved in configuring and testing the SRC VTA configuration.

- First, you need to define the SRC VTA test events. You define a test event by assigning values to a set of attributes. The attributes you can assign to the test event depends on the event type you specify. SRC VTA test event configurations are stored in the Juniper Networks database so that you can reference them when executing SRC VTA test commands from any C Series Controller.
- After you define a test event, you can reference the test event when you execute test commands.

[Table 61 on page 710](#) lists the event types and the configuration statements you use to define SRC VTA test events.

Table 61: SRC VTA Event Types and Test Event Configuration Statements

Test Events	Configuration Statement Used to Define the Test Event	Event Type
-------------	---	------------

Table 61: SRC VTA Event Types and Test Event Configuration Statements (*continued*)

Subscriber-and service-tracking test events	<p>Use the following statement to configure subscriber- and service-tracking test events:</p> <pre>shared vta test-events event-name type type attributes {</pre> <p>The <i>event-name</i> variable is an arbitrary name you specify for the event. When you execute a test command, you specify this name for the <i>test-event</i> variable.</p> <p>See Table 62 on page 711 for setting the <i>attribute-name</i> variable for either subscriber- or service-tracking test events.</p>	<p>You can define the following event types for the <i>event-type</i> variable when defining subscriber-tracking events:</p> <ul style="list-style-type: none"> • user-start • user-interim • user-stop <p>You can define the following event types for the <i>event-type</i> variable when defining service-tracking events:</p> <ul style="list-style-type: none"> • service-start:service-name • service-interim:service-name • service-stop:service-name
Callback test events	<p>Use the following statement to configure callback test events:</p> <pre>shared vta test-events type call-back:name callback-attributes {</pre> <p>You can define the following attribute type for callback test events:</p> <ul style="list-style-type: none"> • Boolean • Long • Integer • Double • Float • String <p>The <i>event-name</i> variable is an arbitrary name you specify for the event. When you execute a test command, you specify this name for the <i>test-event</i> variable.</p>	<p>The only event type you can define for callback test events is callback.</p>

Table 62: Attributes for Subscriber- and Service-Tracking Test Events

Attributes Supported for Subscriber-Tracking Events	Attributes Supported for Service-Tracking Events
PA_ACCOUNTING_ID	PA_ACCOUNTING_ID
PA_AUTH_USER_ID	PA_AGGR_ACCOUNTING_ID
PA_DHCP_PACKET	PA_AGGR_AUTH_USER_ID
PA_EVENT_TIME	PA_AGGR_LOGIN_NAME
PA_EVENT_TIME_MILLISECOND	PA_AGGR_SESSION_ID
PA_IF_RADIUS_CLASS	PA_AGGR_USER_DN
PA_IF_SESSION_ID	PA_AGGR_USER_INET_ADDRESS

Table 62: Attributes for Subscriber- and Service-Tracking Test Events (*continued*)

PA_INTERFACE_ALIAS	PA_AUTH_USER_ID
PA_INTERFACE_DESCR	PA_DHCP_PACKET
PA_INTERFACE_NAME	PA_DOWNSTREAM_BANDWIDTH
PA_LOGIN_ID	PA_EVENT_TIME
PA_LOGIN_NAME	PA_EVENT_TIME_MILLISECOND
PA_NAS_INET_ADDRESS	PA_IF_RADIUS_CLASS
PA_NAS_IP	PA_IF_SESSION_ID
PA_NAS_PORT	PA_IN_OCTETS
PA_OPERATIONAL	PA_IN_PACKETS
PA_PORT_ID	PA_INTERFACE_ALIAS
PA_PRIMARY_USER_NAME	PA_INTERFACE_DESCR
PA_RADIUS_CLASS	PA_INTERFACE_NAME
PA_ROUTER_NAME	PA_LOGIN_ID
PA_SESSION_TIMEOUT	PA_LOGIN_NAME
PA_SSP_HOST	PA_NAS_INET_ADDRESS
PA_SUBSCRIPTION_NAME	PA_NAS_IP
PA_TERMINATE_CAUSE	PA_NAS_PORT
PA_USER_DN	PA_OPERATIONAL
PA_USER_INET_ADDRESS	PA_OUT_OCTETS
PA_USER_IP_ADDRESS	PA_OUT_PACKETS
PA_USER_MAC_ADDRESS	PA_PORT_ID
PA_USER_RADIUS_CLASS	PA_PRIMARY_USER_NAME
PA_USER_TYPE	PA_RADIUS_CLASS
	PA_ROUTER_NAME
	PA_SERVICE_NAME

Table 62: Attributes for Subscriber- and Service-Tracking Test Events (*continued*)

PA_SERVICE_SCOPE
PA_SERVICE_SESSION_NAME
PA_SERVICE_SESSION_TAG
PA_SESSION_ID
PA_SESSION_TIME
PA_SSP_HOST
PA_SUBSCRIPTION_NAME
PA_TERMINATE_CAUSE
PA_UPSTREAM_BANDWIDTH
PA_USER_DN
PA_USER_INET_ADDRESS
PA_USER_IP_ADDRESS
PA_USER_MAC_ADDRESS
PA_USER_RADIUS_CLASS
PA_USER_TYPE

Related Documentation • [Configuring Event Handlers Overview on page 602](#)

Testing SRC VTA Events (SRC CLI)

You can test SRC VTA events for subscriber- and service-tracking events, as well as callback events.

- From operational mode, execute:

```
user@host> test vta events subscriber-id subscriber-identifier event-name event-name
group group
```

For the *event-name* variable, specify the event name you configured when you defined the test event with either the **shared vta test-events *name* callback-attributes *name*** or **shared vta test-events *name* attributes *name*** statement.

- Related Documentation**
- [Configuring Event Handlers Overview on page 602](#)
 - [Testing the SRC VTA Configuration Overview on page 710](#)

PART 10

Index

- [Index on page 717](#)

Index

A

access DNSs.....	203
accounting	
SAE, description.....	13
ACP (Admission Control Plug-In)	
redundancy	
monitoring.....	327
ACP congestion point usage trap	
configuring	294
ACP. <i>See</i> SRC ACP	
action congestion points.....	251
configuring	286
monitoring	
C-Web interface.....	336, 338
SRC CLI.....	324
address pools. <i>See</i> IP address pools	
Admission Control Plug-In. <i>See</i> SRC ACP	
agents <i>See</i> NIC agents	
allocating bandwidth to applications not controlled	
by SRC ACP.....	254
APIs	
SRC ACP.....	258
APIs (application programming interfaces)	
CORBA remote API.....	12
NIC.....	197
provided with SAE.....	11
SAE core API.....	12
application programming interfaces. <i>See</i> APIs	
applications	
executing with SRC ACP.....	251
external for use with SRC ACP.....	251, 255
assigned IP subscribers	
third-party devices.....	115
IP address pools.....	115
assigning	
edge congestion points to subscribers.....	283
interfaces to backbone congestion point	
profiles.....	293
interfaces to subscribers.....	283
ATM access network, using with SRC ACP.....	252
attributes.....	542

authentication plug-ins	
virtual routers	
SRC CLI.....	58, 82
authentication target	
configuration	
SRC CLI.....	510
authorizing and tracking services.....	255

B

backbone congestion point profiles	
configuring.....	293
backbone congestion points.....	268
configuring.....	286
configuring for services.....	287
defining applications in.....	251
deriving.....	253
DNs of.....	253
monitoring	
SRC CLI.....	314, 315
running applications from.....	286
backbone network.....	249
backbone network management with SRC ACP	
configuring.....	285
background bandwidth.....	254
bandwidth	
allocating to applications not controlled by SRC	
ACP.....	254
background.....	254
configuring	
for services.....	284, 287
for subscribers.....	282
downstream.....	250
upstream.....	250
bandwidths and congestion points for subscribers	
configuring.....	282
basic group	
configuration	
SRC CLI.....	520
configuration summary	
SRC CLI.....	472
configuring management of RADIUS-enabled	
devices for the SIC	
SRC CLI.....	474
Diameter configuration summary	
SRC CLI.....	474

RADIUS configuration summary	
SRC CLI.....	472
RADIUS dynamic authorization configuration summary	
SRC CLI.....	473
BEEP, devices running Junos OS.....	4
configuring port	
SRC CLI.....	87
BEEP, Devices running Junos OS	
connection.....	78
Blocks Extensible Exchange Protocol. <i>See</i> BEEP	
C	
certificate authority (CA).....	88
classification scripts	
congestion point classification	
configuring.....	298
criteria.....	297, 300
description.....	297
how it works.....	297
targets.....	297, 299
clients	
gateway	
Web Services Gateway.....	616
Common Object Request Broker Architecture. <i>See</i> CORBA	
community manager	
configuring, third-party devices	
SRC CLI.....	122
component interactions	
devices running Junos OS and SAE.....	4
configuration group, devices running Junos OS.....	78, 97
configuration manager, instantiating for NIC.....	198
congestion point applications	
SPI for ACP.....	285
congestion point classification.....	253
congestion point classification scripts. <i>See</i> classification scripts	
congestion point expressions.....	253, 304
congestion point profiles.....	253
congestion point expressions.....	304
defining.....	304
congestion points.....	249, 250
configuring.....	282
defining applications in.....	251
deriving.....	251
deriving from congestion point expressions.....	253
deriving from profile.....	252
managing.....	254
modifying.....	308
monitoring.....	326, 327
retrieving information about.....	255
congestion points by IP address and associated service sessions	
monitoring	
SRC CLI.....	315
congestion points by login name and associated service sessions	
monitoring	
SRC CLI.....	321
congestion points by session ID and associated service sessions	
monitoring	
SRC CLI.....	318
conventions	
notice icons.....	xxxiii
text.....	xxxiii
COPS (Common Open Policy Service)	
connection with JunosE routers.....	53
configuring SAE, SRC CLI.....	59
disabling on router.....	70
enabling on router.....	70
COPS-PR versus COPS XDR.....	4
JunosE router connection.....	3
CORBA (Common Object Request Broker Architecture)	
IOR location.....	194
remote API.....	12
CORBA interfaces	
SRC ACP.....	274
CORBA-based plug-in SPI. <i>See</i> plug-ins, external	
customer support.....	xxxv
contacting JTAC.....	xxxv
customized interface modules.....	13
D	
database accounting method	
configuration	
SRC CLI.....	508
deriving congestion points.....	251
device drivers	
Junos	
configuring, SRC CLI.....	86
viewing state, C-Web interface.....	101
viewing state, SRC CLI.....	99

- viewing statistics, C-Web interface.....102
 - viewing statistics, SRC CLI.....100, 101
- JunosE
 - configuring, SRC CLI.....59
 - viewing state, SRC CLI.....72
 - viewing statistics, SRC CLI.....74
- devices running Junos OS
 - BEEP connection.....4
 - configuring port, SRC CLI.....87
 - configuration groups.....97
 - configuring to interact with SAE.....95
 - default virtual router.....79
 - disabling interactions with SAE.....97
 - enabling interactions with SAE.....97
 - monitoring interactions with SAE.....98
 - SAE interactions.....4
 - SRC software process.....78, 105
 - troubleshooting.....98
- Devices running Junos OS
 - configuration groups.....78
 - SRC software process.....78
- directory
 - services for SRC ACP.....278
 - subscribers for SRC ACP.....277
- directory blacklist, deleting.....40, 47
- distinguished name. *See* DN
- DMI
 - managing Junos DMI-enabled devices
 - overview.....106
 - managing routers running Junos OS.....78, 105
 - overview of managing DMI-enabled routers
 - running Junos OS through Junos Space
 - SRC CLI.....107
- DMI driver
 - migrating
 - SRC CLI.....110
- DMI network device
 - adding
 - SRC CLI.....108
- DMI, devices running Junos OS
 - connection.....78, 105
- DN (distinguished name)
 - NIC resolution.....203
- DNs
 - backbone congestion points.....253
 - edge congestion points.....252
- documentation
 - comments on.....xxxv
- domain maps
 - reloading on SAE.....47
- downstream bandwidth.....250
- downstream RADIUS network elements and
 - accounting targets
 - configuration statements
 - SRC CLI.....491
- downstream RADIUS network elements and
 - authentication targets
 - configuration statements
 - SRC CLI.....491
- E**
- edge congestion points
 - assigning to subscribers.....283
 - deriving.....251
 - DNs of.....252
 - monitoring
 - SRC CLI.....312, 313
- edge network.....249, 281
- edge network management, configuring.....281
- equipment registration
 - deleting.....41, 48
- event notification, PCMM network
 - configuration statements.....124
 - properties, configuring
 - SRC CLI.....124
- event notification, third-party devices
 - description.....116
- events, publishing.....267
- external applications
 - displaying information from.....325
 - interaction with NIC.....197
 - monitoring
 - C-Web interface.....341
- external plug-ins
 - configuring SRC ACP as.....249
- external plug-ins. *See* plug-ins
- External Subscriber Monitor
 - acting as pseudo RADIUS server, C-Web
 - interface.....347
 - agent process statistics, viewing
 - SRC CLI.....362
 - configuring.....348
 - configuring basic local properties.....348
 - configuring client secret.....355
 - configuring directory connection
 - properties.....349
 - configuring event notification.....356

configuring eventing properties.....	350
configuring initial properties.....	349
configuring logging destinations.....	350
event notifications, monitoring	
SRC CLI.....	362
event notifications, viewing	
SRC CLI.....	360
IP address manager.....	347
C-Web Interface.....	347
overview, C-Web interface.....	347
starting.....	357
statistics, monitoring	
SRC CLI.....	360
statistics, viewing	
SRC CLI.....	359
stopping.....	357
F	
failover parameters, SAE.....	41, 49
fault recovery, SRC ACP.....	257
files	
ACP data.....	271
G	
group	
creation	
SRC CLI.....	476
groups, NIC hosts.....	137
H	
hosted internal plug-in.....	267
hosted plug-ins. See plug-ins	
I	
interactions between SRC ACP and other	
components.....	254, 256
interface classification scripts	
reloading on SAE.....	39, 47
interface modules, SAE.....	12
interfaces, assigning to backbone congestion point	
profiles.....	293
internal plug-ins. See plug-ins	
IOR	
router initialization scripts.....	63, 125
IP address pools	
local address pools, configuring	
SRC CLI.....	58, 82
static pools, configuring	
SRC CLI.....	58, 82
IP pools	
updating	
Junos VR.....	83
JunosE VR.....	67, 68
J	
JunosE routers	
accessing router CLI.....	69
COPS connection.....	3
configuring, SRC CLI.....	59
integration overview.....	53
monitoring interactions with SAE.....	71
router objects, adding	
SRC CLI.....	55
SRC client.....	53
starting.....	70
stopping.....	70
troubleshooting.....	71
VR objects	
adding individually, SRC CLI.....	57, 81
discovering, SRC CLI.....	55
L	
LDAP access. See SAE (service activation engine),	
configuring	
local properties	
configuration	
SRC CLI.....	474
logging properties	
configuring for SRC ACP.....	269
login names.....	203
login process	
assigned IP subscribers, third-party	
devices.....	116
event notification method, third-party	
devices.....	117
login registration	
deleting.....	40, 48
M	
managing	
congestion points.....	254
edge network with SRC ACP.....	281
manuals	
comments on.....	xxxv
methods	
SRC Volume-Tracking Application.....	677
modifying congestion points.....	308

monitoring	
backbone congestion points.....	268
SRC ACP	
C-Web interface.....	329
SRC CLI.....	311
N	
NAS port ID.....	252
network devices	
SNMP communities, configuring	
SRC CLI.....	129
network information collector. <i>See</i> NIC	
network interfaces.....	281, 285
network publisher <i>See</i> NIC	
NIC (network information collector).....	133
API.....	197
configuration prerequisites	
C Series Controllers.....	151
configuration statements.....	149
configuration, changing.....	169
configuration, verifying.....	165
data mapping.....	135
default operating properties, viewing.....	153
factory interface.....	197, 198
logging	
changing configuration.....	158
default.....	158
monitors	
example.....	235
network publisher	
overview.....	171
prerequisites.....	173
running.....	179
troubleshooting.....	181
operating properties, changing.....	153
overview.....	133
planning implementation.....	139
realms	
overview.....	203
replication	
groups.....	137
overview.....	137
SAE plug-in agents.....	163
replication, configuring.....	155
resolution processes.....	203, 204
resolvers	
constraints.....	206
overview.....	136
restarting.....	167
roles.....	204
starting.....	166
stopping.....	166
testing	
any key.....	193
examples.....	167
resolution.....	166
test data.....	193
viewing	
configuration.....	210
<i>See also</i> other NIC entries	
NIC agents	
configuration overview.....	144
directory, configuring	
SRC CLI.....	159
overview.....	136
restarting.....	168
sae client agents, configuring.....	161
sae plug-in agents, configuring.....	162
NIC configuration scenarios	
changing.....	169
SRC CLI.....	156
MultiPop.....	240
OnePop.....	210
OnePopAcctId.....	223
OnePopAllRealms.....	234
OnePopDnSharedIp.....	230
OnePopDynamicIp.....	214
OnePopLogin.....	225
OnePopLoginPull.....	227
OnePopPcmm.....	212
OnePopPrefixIp.....	239
OnePopPrimaryUser.....	228
OnePopSharedIp.....	216
OnePopStaticRoutelp.....	171, 218
OnePopTunnel.....	238
OnePopVrflp.....	171, 221
overview.....	140, 209
scenario-name.....	169
NIC hosts	
configuration prerequisites.....	152
groups.....	137
overview.....	135
redundancy	
example.....	234
starting.....	166
stopping.....	167, 168

NIC locators	
external applications.....	195, 196
overview.....	133, 135
NIC proxies	
cache, configuring	
SRC CLI.....	190
configuration overview.....	185
configuration prerequisites.....	186
instantiating.....	199
logging.....	199
NIC replication, configuring	
SRC CLI.....	191
overview.....	135
prerequisites.....	186
removing instances.....	202
requirements.....	197
resolution information, configuring	
SRC CLI.....	188
resolution requests.....	200
NIC Proxy for Pseudo-RADIUS server	
configuring.....	352
NIC proxy for Pseudo-RADIUS server	
changing configuration.....	352
configuring for NIC replication.....	353
configuring resolution.....	352
NIC resolvers	
restarting.....	168
nic-network-publisher-configuration-statements....	172
notice icons.....	xxxi
O	
operation	
SRC ACP, configuring.....	271
P	
PacketCable Multimedia. <i>See</i> PCMM	
PCMM (PacketCable Multimedia)	
SAE connection.....	4
plug-ins	
architecture.....	10
authentication	
virtual routers, SRC CLI.....	58, 82
external.....	10
hosted.....	11
hosted internal plug-in.....	267
internal.....	10
SRC ACP.....	249
tracking	
virtual routers, SRC CLI.....	58, 82
types.....	10
preventing	
service activation.....	256
priorityList.....	192
properties	
SRC ACP.....	269
proxy RADIUS accounting method	
configuration	
SRC CLI.....	509
pseudo-RADIUS server	
configuring External Subscriber Monitor.....	354
publishing events.....	267
R	
RADIUS accounting listener	
configuring	
SRC CLI.....	480, 482
RADIUS accounting listener queue limits	
configuration	
SRC CLI.....	480
RADIUS accounting listener transport	
configuration	
SRC CLI.....	481
RADIUS authentication listener queue limits	
configuration	
SRC CLI.....	482
RADIUS authentication listener transport	
configuring	
SRC CLI.....	483
randomPick.....	192
realm	
<i>See</i> NIC realms	
redundancy, SRC ACP.....	256
rendering.....	436
resolution processes	
DN to SAE reference.....	230, 234, 244
IP address to login name.....	225
IP address to SAE	
reference.....	210, 216, 230, 234, 241
login name to SAE reference.....	225
roles, NIC.....	204
roundRobin.....	192
router initialization scripts	
developing.....	63, 125
iorPublisher.....	63, 125
isePoolPublisher.....	63, 126

- Junos
 - configuring location, SRC CLI.....94
 - JunosE
 - configuring location, SRC CLI.....66
 - example.....65, 127
 - JunosE Software.....63, 125
 - poolPublisher.....63, 126
 - specifying for NIC.....146
 - router object
 - adding for third-party devices
 - SRC CLI.....119
 - routers
 - accessing router CLI.....69
 - adding devices running Junos OS
 - SRC CLI.....79
 - adding JunosE
 - SRC CLI.....54
 - configuring SAE to communicate with Junos
 - Space
 - SRC CLI.....109
 - integrating devices running Junos OS.....78, 105
 - integrating JunosE.....53
 - SNMP communities, configuring
 - SRC CLI.....129
- S**
- SAE (service activation engine)
 - accounting.....13
 - APIs. See APIs
 - BEEP connection, devices running Junos OS.....4
 - COPS
 - JunosE router connection.....4
 - deleting directory blacklist.....40, 47
 - device running Junos OS client.....95
 - disabling interactions with device running Junos
 - OS.....97
 - enabling interactions with device running Junos
 - OS.....97
 - failover parameters.....41, 49
 - monitoring interactions
 - device running Junos OS.....98
 - JunosE routers.....71
 - NIC replication, configuring
 - SRC CLI.....163
 - overview.....3
 - PCMM environment.....4
 - plug-ins See plug-ins
 - reloading configuration.....38, 45, 46
 - role.....3
 - router initialization scripts. See router
 - initialization scripts
 - session store
 - C Series Controllers.....29
 - starting
 - SRC client on JunosE router.....70
 - stopping
 - SRC client on JunosE router.....70
 - SAE (service activation engine), configuring
 - BEEP connection
 - SRC CLI.....87
 - COPS connection
 - SRC CLI.....59
 - directory eventing, SAE configuration data
 - SRC CLI.....27
 - event notification API properties
 - SRC CLI.....124
 - LDAP access, SRC CLI
 - device data.....26
 - directory data.....20
 - persistent login cache data.....25
 - policy data.....24
 - service data.....23
 - subscriber data.....21
 - router initialization script location
 - SRC CLI.....66, 94
 - serialized data compression
 - SRC CLI.....35
 - session job manager
 - SRC CLI.....35
 - session store
 - SRC CLI.....31
 - SRC ACP.....267
 - SAE (service activation engine), configuring
 - to monitor backbone congestion points.....267
 - SAE communities
 - configuring, third-party devices
 - SRC CLI.....122
 - description, third-party devices.....114
 - SAE remote interface
 - customized interface modules.....13
 - script services
 - for third-party devices.....114
 - serialized data compression, configuring
 - SRC CLI.....35
 - service activation engine. See SAE

services		
configuring bandwidth for.....	284, 287	
monitoring		
C-Web interface.....	332	
SRC CLI.....	313	
preventing activation.....	256	
reloading on SAE.....	39, 46	
session job manager, configuring		
SRC CLI.....	35	
session state registrar See SSR		
session store		
C Series Controllers.....	29	
configuring, SRC CLI		
compressing session objects.....	35	
global parameters.....	33	
in third-party networks.....	114	
SIC		
device models in network element		
configure.....	493	
failover mode		
configure.....	499	
failover mode and policy		
configure.....	498, 499	
fast fail options for the failover policy		
configure.....	500	
retry options for the failover policy		
configure.....	501	
SIC (subscriber information collector)		
accounting listener		
overview.....	456	
accounting methods		
overview.....	446	
authentication methods		
overview.....	450	
configuring basic SIC group		
.....	520	
database accounting method		
overview.....	446	
overview mapping attributes.....	447, 448	
default attributes in tagged attribute group,		
configuring		
SRC CLI.....	560	
device capabilities, configuring		
SRC CLI.....	546	
device models		
overview.....	463	
device models, configuration		
SRC CLI.....	480	
device templates, configuring		
SRC CLI.....	545	
Diameter server		
overview.....	456	
statements.....	486	
Diameter server identity, configuring		
SRC CLI.....	487	
Diameter server peer, configuring		
SRC CLI.....	488	
Diameter server, configuring		
SRC CLI.....	486	
dictionaries		
overview.....	463	
dictionaries, configuration		
SRC CLI.....	478	
downstream network element		
overview.....	456	
dynamic authorization		
how the process works.....	437	
overview.....	435, 441	
editing rules		
overview.....	453	
editing rules, configuration		
SRC CLI.....	505	
editing rules, configuration statements		
SRC CLI.....	502	
event logging		
event level.....	464	
log file.....	464	
log stream.....	464	
overview.....	464	
event logging, configuration		
SRC CLI.....	515	
explicit authentication routing, configuration		
statements		
SRC CLI.....	511	
explicit routing rules		
overview.....	450	
explicit routing, configuration		
SRC CLI.....	484, 512	
explicit routing, configuration statements		
SRC CLI.....	510	
failover mode		
overview.....	460	
failover policy		
overview.....	460	
global service template default attributes,		
configuring		
SRC CLI.....	569	

- global service template mode, configuring
 - SRC CLI.....566
- global service template normal attributes, configuring
 - SRC CLI.....567
- global service template override attributes, configuring
 - SRC CLI.....572
- global service template parameterized attributes, configuring
 - SRC CLI.....571
- global service template required attributes, configuring
 - SRC CLI.....568
- global service template variables, configuring
 - SRC CLI.....567
- global service templates, configuring
 - SRC CLI.....565
- global service templates, creating
 - SRC CLI.....566
- global service templates, overview
 - SRC CLI.....539
- implicit routing rules
 - overview.....452
- implicit routing, configuration
 - SRC CLI.....515
- local and shared configuration
 - overview.....445
- local realms
 - overview.....464
- local realms, configuration
 - SRC CLI.....501
- normal attributes in tagged attribute group, configuring
 - SRC CLI.....559
- override attributes in tagged attribute group, configuring
 - SRC CLI.....562
- overview.....433
- parameterized attributes in tagged attribute group, configuring
 - SRC CLI.....563
- primary/backup
 - overview.....460
- proxy accounting method
 - overview.....449
- proxy function
 - overview.....456
- RADIUS configuration
 - overview.....456
- RADIUS transport for server, configuring
 - SRC CLI.....485
- RADIUS transports
 - overview.....462
- request routing rules
 - overview.....450
- request routing, configuring
 - SRC CLI.....510
- required attributes in tagged attribute group, configuring
 - SRC CLI.....561
- round robin
 - overview.....460
- server instance, creation
 - SRC CLI.....477
- service template default attributes, configuring
 - SRC CLI.....553
- service template mode, configuring
 - SRC CLI.....550
- service template normal attributes, configuring
 - SRC CLI.....551
- service template override attributes, configuring
 - SRC CLI.....555
- service template parameterized attributes, configuring
 - SRC CLI.....554
- service template required attributes, configuring
 - SRC CLI.....552
- service template samples
 - SRC CLI.....573
- service template variables, configuring
 - SRC CLI.....550
- service template, configuration statements
 - SRC CLI.....548
- service template, tagged attribute configuration statements
 - SRC CLI.....557
- service templates, configuring
 - SRC CLI.....549
- service templates, creating
 - SRC CLI.....549
- service templates, overview
 - SRC CLI.....539
- SNMP support
 - overview.....467

snmp, configuring		
SRC CLI.....	519	
statistics		
accounting transactions.....	586	
authentication transactions.....	586	
client accounting requests.....	585	
client authentication requests.....	585	
Diameter host.....	588	
Diameter peer.....	588	
target accounting requests.....	587	
target authentication requests.....	587	
tagged attribute group, creating		
SRC CLI.....	558	
tagged attributes in, configuring		
SRC CLI.....	558	
upstream network element		
overview.....	456	
SIC accounting and authentication clients		
configuring		
SRC CLI.....	494	
SIC accounting and authentication targets		
configuring		
SRC CLI.....	496	
SIC accounting targets		
configuring		
SRC CLI.....	496	
SIC authentication targets		
configuring		
SRC CLI.....	497	
SIC dynamic authorization targets		
configuring		
SRC CLI.....	495	
SIC upstream and downstream network elements		
configuring		
SRC CLI.....	490	
creating		
SRC CLI.....	492	
SNMP		
retrieving information from network		
devices.....	129	
SNMP communities		
configuring		
SRC CLI.....	129	
SOAP		
interfaces.....	616	
Soap Fault Codes		
VTA		
Method	687	
SRC ACP (SRC Admission Control Plug-In).....	281	
API.....	258	
ATM access network.....	252	
authorizing and tracking services.....	255	
backbone network management,		
configuring.....	285	
classification scripts		
configuring.....	280	
configuring.....	262	
congestion points.....	249, 267	
connections to services directory,		
configuring.....	278	
connections to subscribers' directory,		
configuring.....	277	
CORBA interfaces, configuring.....	274	
data files.....	271	
data files, reorganizing.....	308	
description of.....	249	
event publishers, configuring.....	267	
external applications.....	251, 255	
external plug-in for SAE, configuring.....	249	
fault recovery.....	257	
groups, configuring.....	262, 263	
information from external applications,		
displaying.....	325	
interactions with other components.....	254	
logging properties, configuring.....	269	
monitoring		
C-Web interface.....	329	
SRC CLI.....	311	
operation, configuring.....	271	
preventing service activation.....	256	
properties.....	269	
redundancy.....	256	
configuring.....	275	
SAE, configuring for	267	
starting.....	307	
state synchronization.....	256	
configuring.....	275	
stopping.....	307	
subscribers, monitoring.....	311, 329	
supporting multiple SAEs.....	254	
using multiple SRC ACPs.....	254	
SRC ACP (SRC Admission Control Plug-In),		
congestion points.....	249, 267	
SRC Admission Control Plug-In. See SRC ACP		

-
- SRC client, JunosE routers
 - configuring.....53
 - starting.....70
 - stopping.....70
 - SRC software process, Devices running Junos
 - OS.....78
 - SRC software process, devices running Junos
 - OS.....78, 105
 - disabling.....97
 - reenabling.....97
 - SRC Volume Tracking Application
 - API.....616
 - SRC Volume-Tracking Application See SRC VTA
 - SOAP interface
 - methods.....677
 - SRC Volume-Tracking Application (SRC VTA)
 - account and session database connection,
 - configuring See SRC CLI
 - database deadlocks
 - troubleshooting.....634
 - database to store account and session data,
 - configuring See SRC CLI
 - db-engine processor
 - configuring.....651
 - group, configuring See SRC CLI
 - JDBC .jar file See installing
 - keys to specify subscriber-id-solution See SRC CLI
 - tracking events
 - configuring.....630
 - SRC Volume-Tracking Application (VTA)
 - group
 - configuring.....639
 - SRC Volume-Tracking Application. See SRC VTA
 - SRC VTA
 - accounts and service sessions
 - modifying.....702
 - balance change history records
 - deleting.....701
 - events
 - testing.....713
 - performance statistics
 - viewing.....706
 - session history records
 - deleting.....702
 - sessions
 - terminating.....704
 - SOAP API statistics
 - viewing.....709
 - subscriber accounts
 - viewing.....705
 - subscriber balance changes
 - viewing.....705
 - subscriber session history
 - viewing.....706
 - testing configuration
 - overview.....710
 - SRC VTA (SRC Volume-Tracking Application (SRC VTA)
 - enabling674
 - SRC VTA (SRC Volume-Tracking Application)
 - account and session database connection,
 - configuration
 - SRC CLI.....642
 - accounts
 - description.....594
 - interim accounting interval, setting.....654
 - service.....594
 - subscriber.....594
 - usage metric, setting.....651, 654, 659
 - actions.....598
 - overview.....606
 - architecture.....595
 - bought account.....594
 - bought quota.....594
 - example.....616
 - connections to SRC components.....595
 - database
 - optimizing.....655
 - database engine processor
 - configuring.....652
 - db-engine processor
 - configuring.....651
 - e-mail notifications, sending.....661
 - event attributes.....596
 - event handlers.....596
 - event queue
 - configuring.....650
 - events.....596
 - account update.....604
 - callback.....605
 - tracking events from SAE.....604
 - example
 - limiting subscriber access.....616
 - functions
 - overview.....606
 - group, configuring
 - SRC CLI.....639

how it works.....	596	SRC VTA (Volume-Tracking Application)	
initial account balance		actions	
setting	652	configuring.....	647
initial account status		database deadlocks	
setting	652	troubleshooting.....	634
interval accounting interval, setting.....	654, 656	event handlers	
JavaScript programs.....	615	configuring.....	648
keys to specify subscriber-id-solution		function	
SRC CLI.....	640	configuring.....	647
logging events to a text file		on-error	
configuration.....	666	configuring.....	647
configuring.....	666	related configuration tasks	
mail processor		NIC.....	636
configuring.....	661	NIC proxies.....	636
on-error		services and policies.....	635
overview.....	606	subscribers and subscriptions.....	635
operation process.....	601	SSR (session state registrar)	
overview.....	593	adding a client node to an active cluster	
periodic account.....	594	SRC CLI.....	413
periodic quota.....	594	adding a management server to a client node	
example.....	616	in an active cluster	
processors.....	596	SRC CLI.....	414
providing volume-based services.....	594	adding data nodes to an active cluster	
quota service.....	594	SRC CLI.....	412
activation upon deposit.....	615	attribute associations, configuring in an active	
related configuration tasks		cluster	
identifying subscribers, SAEs, and		SRC CLI.....	411
sessions.....	613	cluster configurations.....	373
tracking plug-ins, enterprise		cluster network requirements.....	375
subscribers.....	632	cluster status, viewing	
SAE events.....	604	SRC CLI.....	423
saving event messages on a system logging		configuration statements.....	400
server		configuring	
configuration.....	670	cluster name.....	402
configuring.....	670	cluster nodes.....	404
script runner processor		database memory size.....	406
configuring.....	663	geometry.....	403
scripts		initial cluster.....	401
external.....	663, 664	management server.....	406
JavaScript programs.....	663	subscriber sessions schema.....	407, 409
services		creating	
interim accounting interval, setting.....	656	database	420
sessions.....	594	database memory requirements, viewing	
subscriber login with IP address.....	615	SRC CLI.....	424
types.....	593	database mode, viewing	
usage metric, configuring.....	651, 654, 659	SRC CLI.....	423
		database modes.....	389
		database schema.....	384

- database schema, configuring in an active cluster
 - SRC CLI.....410
- database, viewing running configuration
 - SRC CLI.....425
- deleting
 - all subscriber and service sessions.....422
 - database420
 - subscriber sessions by IP address.....422
- disabling
 - SSR component database421
- distributing the cluster configuration.....389
- enabling
 - SSR component database421
- impact of configuration changes.....399
- making modifications to subscriber sessions
 - table.....388
- node groups.....371
- node types.....370
- overview.....369
- placing
 - SSR database into maintenance mode.....419
 - SSR database into production mode.....419
- planning the cluster topology.....391
- planning worksheets.....393
- removing
 - client node.....417
 - data nodes.....415
 - management server.....418
- restarting
 - SSR component database421
- scaling the cluster.....374
- server requirements.....372
- supported configurations.....377
- viewing, all subscriber sessions
 - SRC CLI.....426
- viewing, subscriber sessions by indexed field
 - SRC CLI.....428
- viewing, subscriber sessions by IP address
 - SRC CLI.....427
- viewing, total number of subscriber sessions
 - SRC CLI.....430
- starting
 - SRC ACP.....307
- state synchronization
 - SRC ACP.....256
- statistics, SRC ACP
 - monitoring
 - C-Web interface.....342, 343, 344
- stopping SRC ACP.....307
- subscriber information collector *See* SIC
 - authentication target, configuring.....510
 - basic group, configuring.....520
 - configuration summary
 - SRC CLI.....472
 - configuring management of RADIUS-enabled devices for the SIC
 - SRC CLI.....474
 - database accounting method, configuring.....508
 - default attributes in tagged attribute group, configuring *See* SIC
 - device capabilities, configuring *See* SIC
 - device models, configuring *See* SIC
 - device templates, configuring *See* SIC
 - Diameter configuration summary
 - SRC CLI.....474
 - Diameter server *See* statements
 - Diameter server identity, configuring *See* SIC
 - Diameter server peer, configuring *See* SIC
 - Diameter server, configuring *See* SIC
 - dictionaries, configuring *See* SIC
 - dynamic authorization
 - how the process works.....437
 - overview.....435
 - editing rules, configuration statements *See* SIC
 - editing rules, configuring *See* SIC
 - event logging, configuring *See* SIC
 - explicit authentication routing, configuration statements *See* SIC
 - explicit routing, configuration statements *See* SIC
 - explicit routing, configuring *See* SIC
 - global service template default attributes, configuring *See* SIC
 - global service template mode, configuring *See* SIC
 - global service template normal attributes, configuring *See* SIC
 - global service template override attributes, configuring *See* SIC
 - global service template parameterized attributes, configuring *See* SIC
 - global service template required attributes, configuring *See* SIC

- global service template variables, configuring
 - See SIC
- global service templates, configuring See SIC
- global service templates, creating See SIC
- global service templates, overview See SIC
- group, creating.....476
- implicit routing, configuring See SIC
- local properties
 - directory connection properties,
 - configuring.....474
- local realms, configuring See SIC
- normal attributes in tagged attribute group,
 - configuring See SIC
- outbound RADIUS transport for group,
 - configuring See SIC
- override attributes in tagged attribute group,
 - configuring See SIC
- parameterized attributes in tagged attribute
 - group, configuring See SIC
- proxy RADIUS accounting method,
 - configuring.....509
- RADIUS accounting listener queue limits,
 - configuring.....480
- RADIUS accounting listener transport,
 - configuring
 - SRC CLI.....481
- RADIUS accounting listener,
 - configuring.....480, 482
- RADIUS authentication listener queue limits,
 - configuring.....482
- RADIUS authentication listener transport,
 - configuring
 - SRC CLI.....483
- RADIUS configuration summary
 - SRC CLI.....472
- RADIUS dynamic authorization configuration
 - summary
 - SRC CLI.....473
- RADIUS transport for server, configuring See
 - SIC
- request routing, configuring See SIC
- required attributes in tagged attribute group,
 - configuring See SIC
- server instance, creating See SIC
- service template default attributes, configuring
 - See SIC
- service template mode, configuring See SIC
- service template normal attributes, configuring
 - See SIC
- service template override attributes, configuring
 - See SIC
- service template parameterized attributes,
 - configuring See SIC
- service template required attributes,
 - configuring See SIC
- service template samples See SIC
- service template variables, configuring See SIC
- service template, configuration statements
 - SRC CLI.....548
- service template, tagged attribute configuration
 - statements
 - SRC CLI.....557
- service templates, configuring See SIC
- service templates, creating See SIC
- service templates, overview See SIC
- snmp, configuring See SIC
- tagged attribute group, creating See SIC
- tagged attributes in service templates,
 - configuring See SIC
- subscriber information collector (SIC)
 - accounting and authentication clients
 - configuring.....494
 - accounting and authentication targets
 - configuring.....496
 - accounting targets
 - configuring.....496
 - authentication targets
 - configuring.....497
 - device models in network element
 - configure.....493
 - dynamic authorization targets
 - configuring.....495
 - failover mode
 - configure.....499
 - failover mode and policy
 - configure.....498, 499
 - fast fail options for the failover policy
 - configure.....500
 - retry options for the failover policy
 - configure.....501
 - upstream and downstream network elements
 - configuring.....490
 - creating.....492
- subscribers
 - assigning interfaces to.....283
 - configuring bandwidths and congestion points
 - for.....282, 283
 - IP addresses.....203

- login names.....203
- monitoring
 - C-Web interface.....329, 339
 - SRC CLI.....311, 325
- provisioned and actual bandwidths.....255
- subscriptions
 - reloading on SAE.....39, 46
- support, technical *See* technical support

T

- targets. *See* classification scripts
- technical support
 - contacting JTAC.....xxxv
- text conventions defined.....xxxiii
- third-party devices
 - creating sessions.....115
 - integrating into SRC network
 - SRC CLI.....113
 - logging in subscribers
 - assigned IP method.....115
 - overview.....115
 - provisioning with script services.....114
 - router objects, adding
 - SRC CLI.....119
 - SAE communities.....114
 - VR objects, adding
 - SRC CLI.....120
- threads
 - configuring for sessions
 - SRC CLI.....35
- tracking plug-ins
 - virtual routers
 - SRC CLI.....58, 82
- troubleshooting
 - devices running Junos OS.....98
 - JunosE routers.....71
- tuning factors for background bandwidth.....254

U

- upstream bandwidth.....250
- upstream network elements, accounting and authentication clients, and dynamic authorization targets
 - configuration statements
 - SRC CLI.....492

V

- virtual routers
 - adding for third-party devices
 - SRC CLI.....120
 - adding individually for JunosE routers
 - SRC CLI.....57, 81
 - adding operative VRs.....79
 - SRC CLI.....55
- Volume-Tracking Application (VTA)
 - prerequisites for running.....619
- VTA *See* SRC VTA
- VTA (SRC Volume-Tracking Application)
 - accounts
 - calculating interim interval.....602
 - calculating usage.....602
 - getting balances.....602
 - event attributes.....602
 - event handlers.....602
 - overview.....602
 - event queue
 - deleting.....704
 - events.....602
 - SAEEventListener.....602
 - sessions
 - closing.....602
- VTA (Volume-Tracking Application)
 - database to store account and session data, configuration
 - SRC CLI.....633
 - prerequisites for running
 - SRC CLI.....619
 - tracking events
 - configuring.....630

W

- Web application server
 - application deployment.....629
 - channel stack
 - configuring.....623
 - configuration statements
 - SRC CLI.....621
 - configuring the Web application server
 - SRC CLI.....621
 - installing Web applications inside.....629
 - local properties
 - configuring622
 - multicast-address
 - configuring.....623

node-id	
configuring.....	624
shared cluster name	
configuring.....	622
shared cluster nodes	
configuring.....	624
shared cluster properties	
configuring.....	623
starting.....	629
Web Services Definition Language. See WSDL	
Web Services Gateway	
clients	
managing.....	616
WSDL files.....	616