

SRC PE Software

Application Services Gateway Configuration Guide

Release

4.0.x



Published: 2010-05-18

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

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.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

SRC PE Software Application Services Gateway Configuration Guide

Release 4.0.x

Copyright © 2010, Juniper Networks, Inc.

All rights reserved. Printed in USA.

Writing: Linda Creed, Justine Kangas, Betty Lew, Helen Shaw

Editing: Fran Mues

Illustration: Nathaniel Woodward

Cover Design: Edmonds Design

Revision History

May 2010—Revision 1

The information in this document is current as of the date listed in the revision history.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. The 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

READ THIS END USER LICENSE AGREEMENT ("AGREEMENT") BEFORE DOWNLOADING, INSTALLING, OR USING THE SOFTWARE. BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE OR OTHERWISE EXPRESSING YOUR AGREEMENT TO THE TERMS CONTAINED HEREIN, YOU (AS CUSTOMER OR IF YOU ARE NOT THE CUSTOMER, AS A REPRESENTATIVE/AGENT AUTHORIZED TO BIND THE CUSTOMER) CONSENT TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT OR CANNOT AGREE TO THE TERMS CONTAINED HEREIN, THEN (A) DO NOT DOWNLOAD, INSTALL, OR USE THE SOFTWARE, AND (B) YOU MAY CONTACT JUNIPER NETWORKS REGARDING LICENSE TERMS.

1. **The Parties.** The parties to this Agreement are (i) Juniper Networks, Inc. (if the Customer's principal office is located in the Americas) or Juniper Networks (Cayman) Limited (if the Customer's principal office is located outside the Americas) (such applicable entity being referred to herein as "Juniper"), and (ii) the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").

2. **The Software.** In this Agreement, "Software" means the program modules and features of the Juniper or Juniper-supplied software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller, or which was embedded by Juniper in equipment which Customer purchased from Juniper or an authorized Juniper reseller. "Software" also includes updates, upgrades and new releases of such software. "Embedded Software" means Software which Juniper has embedded in or loaded onto the Juniper equipment and any updates, upgrades, additions or replacements which are subsequently embedded in or loaded onto the equipment.

3. **License Grant.** Subject to payment of the applicable fees and the limitations and restrictions set forth herein, Juniper grants to Customer a non-exclusive and non-transferable license, without right to sublicense, to use the Software, in executable form only, subject to the following use restrictions:

- a. Customer shall use Embedded Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller.
- b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees; provided, however, with respect to the Steel-Belted Radius or Odyssey Access Client software only, Customer shall use such Software on a single computer containing a single physical random access memory space and containing any number of processors. Use of the Steel-Belted Radius or IMS AAA software on multiple computers or virtual machines (e.g., Solaris zones) requires multiple licenses, regardless of whether such computers or virtualizations are physically contained on a single chassis.
- c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, realms, devices, links, ports or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.
- d. For any trial copy of the Software, Customer's right to use the Software expires 30 days after download, installation or use of the Software. Customer may operate the Software after the 30-day trial period only if Customer pays for a license to do so. Customer may not extend or create an additional trial period by re-installing the Software after the 30-day trial period.
- e. The Global Enterprise Edition of the Steel-Belted Radius software may be used by Customer only to manage access to Customer's enterprise network. Specifically, service provider customers are expressly prohibited from using the Global Enterprise Edition of the Steel-Belted Radius software to support any commercial network access services.

The foregoing license is not transferable or assignable by Customer. No license is granted herein to any user who did not originally purchase the applicable license(s) for the Software from Juniper or an authorized Juniper reseller.

4. **Use Prohibitions.** Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not: (a) modify, unbundle, reverse engineer, or create derivative works based on the Software; (b) make unauthorized copies of the Software (except as necessary for backup purposes); (c) rent, sell, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party; (d) remove any proprietary notices, labels, or marks on or in any copy of the Software or any product in which the Software is embedded; (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market; (f) use any 'locked' or key-restricted feature, function, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the

Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use Embedded Software on non-Juniper equipment; (j) use Embedded Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) use the Software in any manner other than as expressly provided herein.

5. **Audit.** Customer shall maintain accurate records as necessary to verify compliance with this Agreement. Upon request by Juniper, Customer shall furnish such records to Juniper and certify its compliance with this Agreement.

6. **Confidentiality.** The Parties agree that aspects of the Software and associated documentation are the confidential property of Juniper. As such, Customer shall exercise all reasonable commercial efforts to maintain the Software and associated documentation in confidence, which at a minimum includes restricting access to the Software to Customer employees and contractors having a need to use the Software for Customer's internal business purposes.

7. **Ownership.** Juniper and Juniper's licensors, respectively, retain ownership of all right, title, and interest (including copyright) in and to the Software, associated documentation, and all copies of the Software. Nothing in this Agreement constitutes a transfer or conveyance of any right, title, or interest in the Software or associated documentation, or a sale of the Software, associated documentation, or copies of the Software.

8. **Warranty, Limitation of Liability, Disclaimer of Warranty.** The warranty applicable to the Software shall be as set forth in the warranty statement that accompanies the Software (the "Warranty Statement"). Nothing in this Agreement shall give rise to any obligation to support the Software. Support services may be purchased separately. Any such support shall be governed by a separate, written support services agreement. TO THE MAXIMUM EXTENT PERMITTED BY LAW, JUNIPER SHALL NOT BE LIABLE FOR ANY LOST PROFITS, LOSS OF DATA, OR COSTS OR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, THE SOFTWARE, OR ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. IN NO EVENT SHALL JUNIPER BE LIABLE FOR DAMAGES ARISING FROM UNAUTHORIZED OR IMPROPER USE OF ANY JUNIPER OR JUNIPER-SUPPLIED SOFTWARE. EXCEPT AS EXPRESSLY PROVIDED IN THE WARRANTY STATEMENT TO THE EXTENT PERMITTED BY LAW, JUNIPER DISCLAIMS ANY AND ALL WARRANTIES IN AND TO THE SOFTWARE (WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE), INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT DOES JUNIPER WARRANT THAT THE SOFTWARE, OR ANY EQUIPMENT OR NETWORK RUNNING THE SOFTWARE, WILL OPERATE WITHOUT ERROR OR INTERRUPTION, OR WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK. In no event shall Juniper's or its suppliers' or licensors' liability to Customer, whether in contract, tort (including negligence), breach of warranty, or otherwise, exceed the price paid by Customer for the Software that gave rise to the claim, or if the Software is embedded in another Juniper product, the price paid by Customer for such other product. Customer acknowledges and agrees that Juniper has set its prices and entered into this Agreement in reliance upon the disclaimers of warranty and the limitations of liability set forth herein, that the same reflect an allocation of risk between the Parties (including the risk that a contract remedy may fail of its essential purpose and cause consequential loss), and that the same form an essential basis of the bargain between the Parties.

9. **Termination.** Any breach of this Agreement or failure by Customer to pay any applicable fees due shall result in automatic termination of the license granted herein. Upon such termination, Customer shall destroy or return to Juniper all copies of the Software and related documentation in Customer's possession or control.

10. **Taxes.** All license fees payable under this agreement are exclusive of tax. Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software. If applicable, valid exemption documentation for each taxing jurisdiction shall be provided to Juniper prior to invoicing, and Customer shall promptly notify Juniper if their exemption is revoked or modified. All payments made by Customer shall be net of any applicable withholding tax. Customer will provide reasonable assistance to Juniper in connection with such withholding taxes by promptly: providing Juniper with valid tax receipts and other required documentation showing Customer's payment of any withholding taxes; completing appropriate applications that would reduce the amount of withholding tax to be paid; and notifying and assisting Juniper in any audit or tax proceeding related to transactions hereunder. Customer shall comply with all applicable tax laws and regulations, and Customer will promptly pay or reimburse Juniper for all costs and damages related to any liability incurred by Juniper as a result of Customer's non-compliance or delay with its responsibilities herein. Customer's obligations under this Section shall survive termination or expiration of this Agreement.

11. **Export.** Customer agrees to comply with all applicable export laws and restrictions and regulations of any United States and any applicable foreign agency or authority, and not to export or re-export the Software or any direct product thereof in violation of any such restrictions, laws or regulations, or without all necessary approvals. Customer shall be liable for any such violations. The version of the Software supplied to Customer may contain encryption or other capabilities restricting Customer's ability to export the Software without an export license.

12. **Commercial Computer Software.** The Software is "commercial computer software" and is provided with restricted rights. Use, duplication, or disclosure by the United States government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7201 through 227.7202-4, FAR 12.212, FAR 27.405(b)(2), FAR 52.227-19, or FAR 52.227-14 (ALT III) as applicable.

13. **Interface Information.** To the extent required by applicable law, and at Customer's written request, Juniper shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of applicable fee, if any. Customer shall observe strict obligations of confidentiality with respect to such information and shall use such information in compliance with any applicable terms and conditions upon which Juniper makes such information available.

14. **Third Party Software.** Any licensor of Juniper whose software is embedded in the Software and any supplier of Juniper whose products or technology are embedded in (or services are accessed by) the Software shall be a third party beneficiary with respect to this Agreement, and such licensor or vendor shall have the right to enforce this Agreement in its own name as if it were Juniper. In addition, certain third party software may be provided with the Software and is subject to the accompanying license(s), if any, of its respective owner(s). To the extent portions of the Software are distributed under and subject to open source licenses obligating Juniper to make the source code for such portions publicly available (such as the GNU General Public License ("GPL") or the GNU Library General Public License ("LGPL")), Juniper will make such source code portions (including Juniper modifications, as appropriate) available upon request for a period of up to three years from the date of distribution. Such request can be made in writing to Juniper Networks, Inc., 1194 N. Mathilda Ave., Sunnyvale, CA 94089, ATTN: General Counsel. You may obtain a copy of the GPL at <http://www.gnu.org/licenses/gpl.html>, and a copy of the LGPL at <http://www.gnu.org/licenses/lgpl.html>.

15. **Miscellaneous.** This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. The provisions of the U.N. Convention for the International Sale of Goods shall not apply to this Agreement. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous agreements relating to the Software, whether oral or written (including any inconsistent terms contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement. This Agreement and associated documentation has been written in the English language, and the Parties agree that the English version will govern. (For Canada: Les parties aux présentes confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattache, soient rédigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).

Table of Contents

	About the Documentation	xvii
	SRC Documentation and Release Notes	xvii
	Audience	xvii
	Documentation Conventions	xvii
	Documentation Feedback	xix
	Requesting Technical Support	xix
	Self-Help Online Tools and Resources	xx
	Opening a Case with JTAC	xx
Part 1	Accessing SRC Software in a Business-to-Business Environment	
Chapter 1	Overview of the Web Services Gateway	3
	Overview of the Web Services Gateway	3
	Terminology	4
Chapter 2	Activating Services with SOAP	7
	Overview of Dynamic Service Activator	7
	Dynamic Service Activator Operation	8
	Dynamic Service Activator in a Redundant Environment	9
Chapter 3	Configuring Dynamic Service Activator (SRC CLI)	11
	Configuration Statements for Dynamic Service Activator	11
	Before You Use Dynamic Service Activator	14
	Enabling Dynamic Service Activator on a Web Application Server (SRC CLI)	15
	Creating Grouped Configurations for Dynamic Service Activator (SRC CLI)	15
	Configuring Dynamic Service Activator (SRC CLI)	16
	Configuring Local Properties for Dynamic Service Activator (SRC CLI)	16
	Configuring Basic Local Properties for Dynamic Service Activator	16
	Configuring Initial Properties for Dynamic Service Activator	17
	Configuring Initial Directory Connection Properties for Dynamic Service Activator	18
	Configuring Initial Directory Eventing Properties for Dynamic Service Activator	18
	Configuring Dynamic Service Activator Properties (SRC CLI)	19
	Configuring General Properties for Dynamic Service Activator	19
	Configuring Subscriber Types for Dynamic Service Activator	20
	Configuring Session Handles for Dynamic Service Activator	21
	Configuring the NIC Proxies for Dynamic Service Activator	21

	Configuring Access to Methods and Scripts for Dynamic Service Activator	22
	Configuring Access to Methods for Dynamic Service Activator	23
	Configuring Methods	23
	Configuring Access to Methods	26
	Configuring Access to Scripts for Dynamic Service Activator	26
	Configuring Scripts	27
	Configuring Access to Scripts	27
	Restricting Access to Service Sessions for Dynamic Service Activator	28
	Configuring Access to Attributes for Dynamic Service Activator	28
	Configuring the SAE to Send Tracking Events to Dynamic Service Activator	29
	Configuring Dynamic Service Activator to Publish Events to SOAP Applications	30
	Configuring External SOAP Applications	30
	Configuring Event Subscriptions	31
	Configuring the Logging Destinations for Dynamic Service Activator	32
	Configuring Logging Destinations to Store Messages in a File	32
	Configuring Logging Destinations to Send Messages to the System Logging Facility	33
	Deploying Dynamic Service Activator on the Virtual Host (SRC CLI)	33
	Starting Dynamic Service Activator (SRC CLI)	34
	Sample Data for Dynamic Service Activator	34
	Methods, Scripts, and Clients	34
	PCMM Available Services	35
Chapter 4	Configuring Dynamic Service Activator (C-Web Interface)	37
	Enabling Dynamic Service Activator on a Web Application Server (C-Web Interface)	37
	Creating Grouped Configurations for Dynamic Service Activator (C-Web Interface)	38
	Configuring Dynamic Service Activator (C-Web Interface)	39
	Configuring Local Properties for Dynamic Service Activator (C-Web Interface)	39
	Configuring Basic Local Properties for Dynamic Service Activator	39
	Configuring Initial Properties for Dynamic Service Activator	39
	Configuring Initial Directory Connection Properties for Dynamic Service Activator	39
	Configuring Initial Directory Eventing Properties for Dynamic Service Activator	40
	Configuring Dynamic Service Activator Properties (C-Web Interface)	40
	Configuring General Properties for Dynamic Service Activator	40
	Configuring Subscriber Types for Dynamic Service Activator	41
	Configuring the NIC Proxies for Dynamic Service Activator	41
	Configuring Access to Methods and Scripts for Dynamic Service Activator	42
	Configuring Access to Methods for Dynamic Service Activator	42
	Configuring Access to Scripts for Dynamic Service Activator	43
	Configuring the Logging Destinations to Store Log Messages in a File	43

	Configuring the Logging Destinations to Send Log Messages to the System	
	Logging Facility	44
	Deploying Dynamic Service Activator on the Virtual Host (C-Web Interface) . . .	44
	Starting Dynamic Service Activator (C-Web Interface)	45
Chapter 5	Monitoring Dynamic Service Activator (SRC CLI)	47
	SRC CLI Commands to Monitor Dynamic Service Activator	47
	Output Control Keys for monitor Command	48
	Viewing Statistics for Dynamic Service Activator (SRC CLI)	48
	Viewing Information About SOAP Operations (SRC CLI)	48
	Monitoring SOAP Operations (SRC CLI)	49
	Viewing Information About NIC Proxies (SRC CLI)	49
	Monitoring NIC Proxies (SRC CLI)	49
Chapter 6	Monitoring Dynamic Service Activator (C-Web Interface)	51
	Viewing Statistics for Dynamic Service Activator (C-Web Interface)	51
	Viewing Information About SOAP Operations (C-Web Interface)	51
	Viewing Information About NIC Proxies (C-Web Interface)	52
Chapter 7	Testing Dynamic Service Activator (SRC CLI)	53
	Testing the Web Application Gateway Client	53
	Configuring the Test Environment for Dynamic Service Activator Services	54
	Configuring Settings for Dynamic Service Activator Services (SRC CLI)	54
	Configuring the Subscriber URI for Dynamic Service Activator Services (SRC CLI)	55
	Verifying Settings for Dynamic Service Activator Services (SRC CLI)	56
	Deleting Settings for Dynamic Service Activator Services (SRC CLI)	56
	Running Methods and Scripts for Dynamic Service Activator Services (SRC CLI)	57
	Testing Subscriber Logins and Logouts (SRC CLI)	58
	Testing Subscriber Access to Subscriptions (SRC CLI)	58
	Testing Subscription Activations and Deactivations (SRC CLI)	59
	Testing Subscription Modifications (SRC CLI)	59
	Testing Script Invocations (SRC CLI)	60
	Testing Gateway Extension Invocations (SRC CLI)	60
	Testing Access to Attributes for Subscriber Sessions (SRC CLI)	60
	Example: Testing Subscriber Access to Subscriptions	61
	Configuring the Test Environment for On-Demand Video Services	61
	Configuring Settings for On-Demand Video Services (SRC CLI)	61
	Verifying Settings for On-Demand Video Services (SRC CLI)	62
	Deleting Settings for On-Demand Video Services (SRC CLI)	62
	Running Methods for On-Demand Video Services (SRC CLI)	62
	Testing Queries for Active Sessions (SRC CLI)	63
	Testing Resource Release Requests (SRC CLI)	63
	Testing Resource Reservation Requests (SRC CLI)	64
	Configuring the Test Environment for Personalized Video Services	65
	Configuring Settings for Personalized Video Services (SRC CLI)	65
	Verifying Settings for Personalized Video Services (SRC CLI)	65
	Deleting Settings for Personalized Video Services (SRC CLI)	66

	Running Methods for Personalized Video Services (SRC CLI)	66
	Testing Resource Allocation Requests (SRC CLI)	66
	Testing Queries for Active Sessions (SRC CLI)	68
	Testing Resource Release Requests (SRC CLI)	68
	Configuring the Test Environment for PCMM Services	69
	Configuring Settings for PCMM Services (SRC CLI)	69
	Verifying Settings for PCMM Services (SRC CLI)	69
	Deleting Settings for PCMM Services (SRC CLI)	70
	Running Methods for PCMM Services (SRC CLI)	70
	Testing Resource Requests (SRC CLI)	71
	Testing Resource Release Requests (SRC CLI)	71
	Testing Queries for Subscriber Contexts (SRC CLI)	72
	Testing Queries for Available Services (SRC CLI)	72
Chapter 8	Developing Gateway Clients	75
	API for Dynamic Service Activator	75
	Public SOAP Interfaces of Web Applications	75
	Methods for the Dynamic Service Activator Web Service Interface	76
	Format of the Subscriber's URI	83
	Subscription Attributes	85
	SOAP Fault Codes for Dynamic Service Activator	86
Chapter 9	Activating Video Services with SOAP	91
	Overview of Web Service Interface for Video Services	91
	Web Service Interface Methods for On-Demand Video Services	92
	Web Service Interface Methods for Personalized Video Services	93
	Configuring Dynamic Service Activator for Video Services	95
	Configuring Dynamic Service Activator as an External Plug-In (SRC CLI)	96
	Configuring the Video Policy Manager (SRC CLI)	96
	Configuring the Video Server	96
	Configuring the Video Service Mappings (SRC CLI)	97
	Configuring the Session Database (SRC CLI)	99
	Configuring the Session Database Cluster Server (SRC CLI)	99
	Configuring the Session Database Topology (SRC CLI)	99
	Deleting Session Database Configurations	100
	Monitoring the Session Database (SRC CLI)	100
	Viewing the Session Database Status (SRC CLI)	100
	Viewing Active Service Sessions (SRC CLI)	100
Chapter 10	Activating PCMM Services with SOAP	101
	Overview of Web Service Interface for PCMM	101
	SRC PCMM Web Service Interface Methods	101
	Configuring PCMM Policies and Parameter Substitutions (SRC CLI)	104
	Configuring Classify-Traffic Conditions for Dynamic Service Activator	104
	Configuring FlowSpec Actions for Dynamic Service Activator	105
	Configuring Service Class Name Actions for Dynamic Service Activator	105
	Configuring DOCSIS Actions for Dynamic Service Activator	105
	Configuring Services That Are Available for PCMM Clients (SRC CLI)	106

Part 2	Providing Services in IMS Networks	
Chapter 11	Providing Services in IMS Networks	111
	Overview of an IMS Environment	111
	IMS and ETSI References	112
	Abbreviations	113
	IMS Layers	114
	Signaling Protocol	115
	ETSI-TISPAN Architecture	115
	RACS Layer	115
	Rq Interface	116
	SPDF	116
	A-RACF	116
	SRC Software in the ETSI-TISPAN Architecture	117
	SRC Software in the IMS Environment	117
	State Synchronization	118
	Redundancy	118
Chapter 12	Providing Services in IMS Networks (SRC CLI)	121
	Configuration Statements for IMS Support	121
	Configuring the IMS Software (SRC CLI)	123
	Configuring Initial Properties for IMS (SRC CLI)	123
	Configuring Directory Connection Properties for IMS (SRC CLI)	124
	Configuring Initial Directory Eventing Properties for IMS (SRC CLI)	125
	Configuring the Local Diameter Peer (SRC CLI)	126
	Configuring the Remote Diameter Peer (SRC CLI)	127
	Configuring Logging Destinations to Store Messages in a File (SRC CLI)	128
	Configuring Logging Destinations to Send Messages to the System Logging Facility (SRC CLI)	129
	Creating Grouped Configurations for IMS (SRC CLI)	130
	Configuring the Subscriber Type (SRC CLI)	131
	Configuring a NIC Proxy for IMS (SRC CLI)	132
	Configuring Resolution Information for a NIC Proxy	132
	Changing the Configuration for the NIC Proxy Cache	133
	Configuring a NIC Proxy for NIC Replication	134
	Configuring NIC Test Data	136
	Configuring IMS for Failover (SRC CLI)	137
	Configuring the SAE for IMS	138
	Configuring IMS as an External Plug-In	138
	Configuring Event Publishers	138
	Managing IMS (SRC CLI)	139
	Starting the IMS Process (SRC CLI)	139
	Restarting the IMS Process (SRC CLI)	139
	Stopping the IMS Process (SRC CLI)	139
	Displaying IMS Status (SRC CLI)	140
	Monitoring IMS (SRC CLI)	140
	Viewing Server Process Information	140
	Viewing Statistics for the Rq Interface	140
	Viewing Information About Peers	141

	Monitoring IMS (C-Web Interface)	141
	Viewing Statistics for the Server Process	141
	Viewing Statistics for the A-RACF Rq Interface	142
	Example: Configuring JUNOS Policies for IMS (SRC CLI)	143
	Enabling Expansion of JUNOS Classify-Traffic Conditions	143
Chapter 13	Testing IMS Service Sessions (SRC CLI)	145
	Testing Service Sessions for IMS	145
	Configuring the Test Environment for IMS Services (SRC CLI)	145
	Configuring Settings for AAR Messages (SRC CLI)	145
	Configuring the Globally Unique Address (SRC CLI)	146
	Configuring Service Information for Media Types (SRC CLI)	147
	Configuring IP Flows for Media Types (SRC CLI)	148
	Testing Service Sessions (SRC CLI)	149
	Testing Session Activations (SRC CLI)	149
	Testing Session Modifications (SRC CLI)	150
	Testing Session Deactivations (SRC CLI)	150
Part 3	Index	
	Index	153

List of Figures

Part 1	Accessing SRC Software in a Business-to-Business Environment	
Chapter 1	Overview of the Web Services Gateway	3
	Figure 1: Web Services Gateway Architecture	4
Chapter 2	Activating Services with SOAP	7
	Figure 2: Dynamic Service Activator Operation	9
Part 2	Providing Services in IMS Networks	
Chapter 11	Providing Services in IMS Networks	111
	Figure 3: A Simplified IMS Converged Network (Service Focus)	112
	Figure 4: High-Level View of the IMS Architecture	114
	Figure 5: High-Level View of the ETSI-TISPAN Architecture	115
	Figure 6: SRC Software in the ETSI-TISPAN Architecture	117
	Figure 7: Juniper Networks IMS Architecture	118

List of Tables

	About the Documentation	xvii
	Table 1: Notice Icons	xviii
	Table 2: Text Conventions	xviii
Part 1	Accessing SRC Software in a Business-to-Business Environment	
Chapter 1	Overview of the Web Services Gateway	3
	Table 3: Web Services Gateway Terms	4
Chapter 5	Monitoring Dynamic Service Activator (SRC CLI)	47
	Table 4: Commands to Monitor Dynamic Service Activator	47
	Table 5: Output Control Keys for the monitor Command	48
Chapter 10	Activating PCMM Services with SOAP	101
	Table 6: Parameter Names for Classify-Traffic Conditions	104
	Table 7: Parameter Names for FlowSpec Actions	105
	Table 8: Parameter Names for Service Class Actions	105
	Table 9: Parameter Names for DOCSIS Actions	106
Part 2	Providing Services in IMS Networks	
Chapter 11	Providing Services in IMS Networks	111
	Table 10: Abbreviations in the IMS and ETSI-TISPAN Environments	113

About the Documentation

- SRC Documentation and Release Notes on page xvii
- Audience on page xvii
- Documentation Conventions on page xvii
- Documentation Feedback on page xix
- Requesting Technical Support on page xix

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® 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 xviii defines the notice icons used in this guide. Table 2 on page xviii 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

Accessing SRC Software in a Business-to-Business Environment

- Overview of the Web Services Gateway on page 3
- Activating Services with SOAP on page 7
- Configuring Dynamic Service Activator (SRC CLI) on page 11
- Configuring Dynamic Service Activator (C-Web Interface) on page 37
- Monitoring Dynamic Service Activator (SRC CLI) on page 47
- Monitoring Dynamic Service Activator (C-Web Interface) on page 51
- Testing Dynamic Service Activator (SRC CLI) on page 53
- Developing Gateway Clients on page 75
- Activating Video Services with SOAP on page 91
- Activating PCMM Services with SOAP on page 101

CHAPTER 1

Overview of the Web Services Gateway

- Overview of the Web Services Gateway on page 3
- Terminology on page 4

Overview of the Web Services Gateway

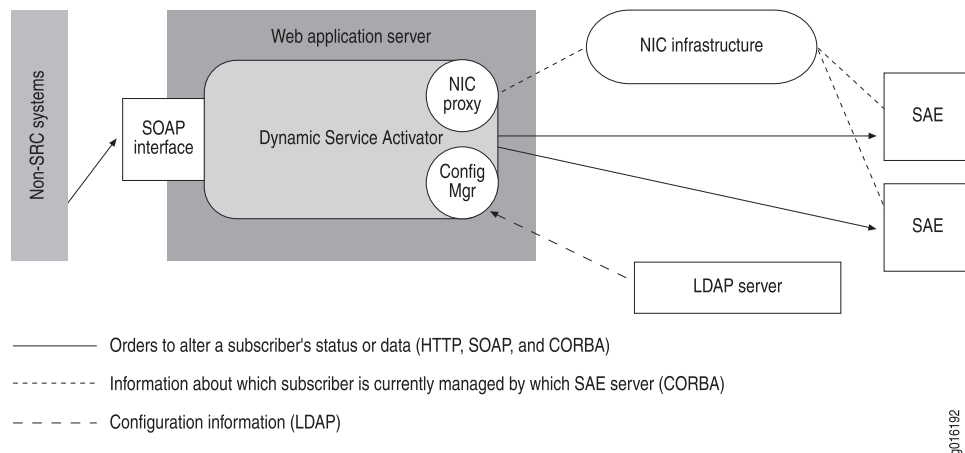
The Web Services Gateway allows a *gateway client*—an application that is not part of the SRC network—to interact with SRC components through a Simple Object Access Protocol (SOAP) interface. This feature is useful for business-to-business (B2B) situations, such as wholesaler-retailer environments. Typically, the wholesaler owns and administers the SRC components, whereas the retailer maintains a database of subscribers. Retailers purchase services from one or more wholesalers and sell the services to their subscribers.

The Web Services Gateway supports Web applications that allow gateway clients to interact with the SRC network. The SRC owner installs, configures, and administers the Web applications. Using information provided by the SRC owner, the business partner creates gateway clients to communicate with the SRC components.

The Web Services Gateway offers the Dynamic Service Activation Web application (subsequently known as Dynamic Service Activator). The Dynamic Service Activator allows a gateway client to dynamically activate and deactivate services for subscribers and to run scripts that manage the SAE.

Figure 1 on page 4 shows the architecture for the Web Services Gateway.

Figure 1: Web Services Gateway Architecture



- Related Topics**
- Overview of Dynamic Service Activator on page 7
 - Dynamic Service Activator in a Redundant Environment on page 9
 - Starting Dynamic Service Activator (SRC CLI) on page 34
 - Terminology on page 4

Terminology

Table 3 on page 4 provides a list of terms and corresponding definitions that are used in the gateway documentation.

Table 3: Web Services Gateway Terms

Term	Definition
Argument	Value that a gateway client passes to a Web application; the application uses the value to perform an action on behalf of the gateway client
Business partner	Organization that manages a database of subscribers and uses an SRC owner's equipment to provision services
Dynamic properties	Properties that the software changes in the directory
Gateway client	Software application that submits requests to a gateway Web application
Gateway extension	Servlet deployed on the Web Services Gateway Web application server
Gateway Web application	Software application that belongs to the SRC owner and allows business partners to interact with the SRC network
Configuration namespace	Entry in the directory that defines a list of properties and inherits properties from parent objects in the hierarchy

Table 3: Web Services Gateway Terms (*continued*)

Term	Definition
SRC owner	Organization that owns the SRC components and software and offers the use of this equipment to business partners who manage a database of subscribers
Static properties	Properties that you change in the directory
Web application server	Software application that supports Web applications and serves Web pages to browsers through HTTP

- Related Topics**
- Overview of the Web Services Gateway on page 3
 - Overview of Dynamic Service Activator on page 7
 - Dynamic Service Activator in a Redundant Environment on page 9

CHAPTER 2

Activating Services with SOAP

- Overview of Dynamic Service Activator on page 7
- Dynamic Service Activator in a Redundant Environment on page 9

Overview of Dynamic Service Activator

Dynamic Service Activator enables business partners or their subscribers to dynamically activate services or run scripts on an SRC owner's SAE through the SAE's CORBA remote interface.

For managing services, Dynamic Service Activator supports a fixed set of methods and uses the SAE access interface module to access the SAE core API. For invoking scripts, Dynamic Service Activator uses the remote Java scripts interface module. These scripts can perform any function offered by the SAE's core Java APIs.

For access control, Dynamic Service Activator requires the Juniper Networks database to be running on the same host.

The SRC owner is responsible for:

- Deciding how to control clients' access to methods and scripts. You can allow clients to access all methods and scripts in the directory or restrict clients' access to specific methods and scripts.
- Configuring Dynamic Service Activator. If you restrict clients' access to specific methods and scripts, this task involves configuring a set of access controls between a client and each method or script that the client can use.
- Creating Java scripts that Dynamic Service Activator will invoke on an SAE (see the SAE CORBA Remote API documentation on the Juniper Networks Web site at <http://www.juniper.net/techpubs/software/management/src/api-index.html>).

The business partner is responsible for:

- Creating the gateway clients that communicate with the gateway.
- Optionally, providing a way for subscribers to activate services; for example, through a portal.

Dynamic Service Activator Operation

The following steps explain how Dynamic Service Activator interacts with other components to enable the gateway client to execute a method or script on a particular SAE. Figure 2 on page 9 illustrates the processes.

1. The gateway client sends a SOAP message to the Web application server through HTTP.

The request includes:

- Name of the method or script that the gateway client wants to activate.
 - Arguments that the gateway client wants to pass to the method or script.
 - Type-value arguments that the gateway client passes to the method or script for one of the following:
 - Subscriber's DN
 - Name with which the subscriber logs in
 - Name of the interface and name of the virtual router to which the subscriber connects
 - SNMP index of the interface and name of the virtual router to which the subscriber connects
 - Subscriber's IP address, name of the managed interface, and name of the virtual router to which the subscriber connects
 - Subscriber's primary username
 - Subscriber's session handle
2. The Web application server authenticates the gateway client's identity.
 3. The Web application passes the SOAP request to Dynamic Service Activator.
 4. Dynamic Service Activator checks that:
 - a. The Web application server has authenticated the gateway client and refused any requests from an unauthenticated gateway client.
 - b. The gateway client is allowed to access the specified method or script.
 - c. The arguments supplied by the gateway client satisfy any restrictions specified in the Dynamic Service Activator configuration that apply to the gateway client for the requested method or script.
 5. If the gateway client satisfies these requirements, Dynamic Service Activator passes an argument, such as a subscriber's IP address specified in the Dynamic Service Activator configuration, to the network information collector (NIC).
 6. The NIC uses the argument to determine the SAE on which Dynamic Service Activator should execute the method or script.

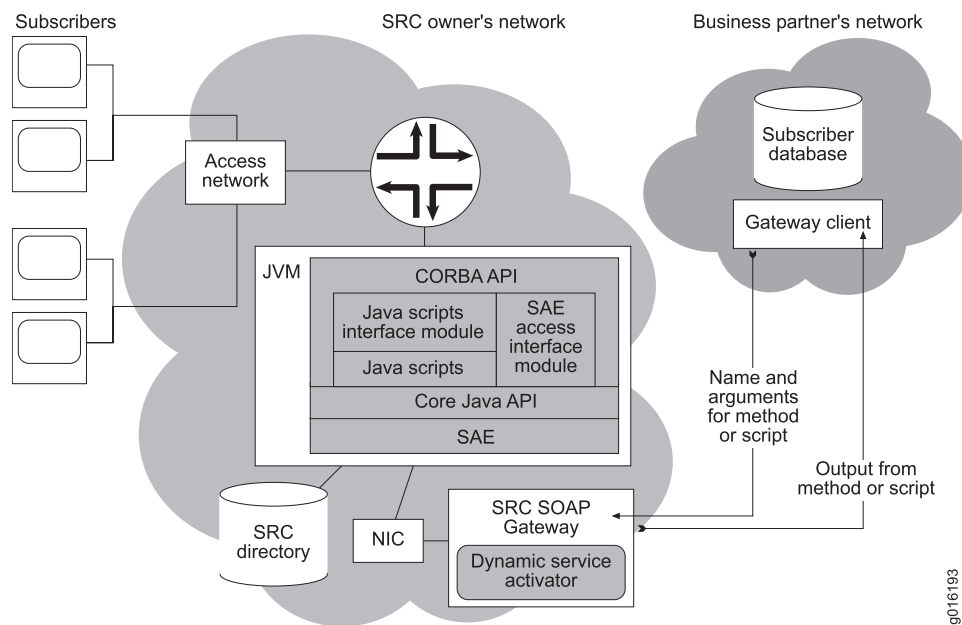
7. Dynamic Service Activator passes the name of the method or script and the associated arguments to the SAE through CORBA.
8. The SAE executes the method or script and returns the expected output or SOAP fault codes through CORBA to Dynamic Service Activator.

The expected output from the method or script depends on the values that the method or script is programmed to return. Some methods and scripts return no values; others may return a short indicator of the success or failure of the operation, an HTML page, or a complex data structure in a format the gateway client understands.

For information about the SOAP fault codes that the methods and scripts return, see “SOAP Fault Codes for Dynamic Service Activator” on page 86.

9. Dynamic Service Activator returns an output from the method or script to the gateway client through a SOAP response.

Figure 2: Dynamic Service Activator Operation



- Related Topics**
- API for Dynamic Service Activator on page 75
 - Dynamic Service Activator in a Redundant Environment on page 9
 - Before You Use Dynamic Service Activator on page 14
 - Starting Dynamic Service Activator (SRC CLI) on page 34
 - Configuration Statements for Dynamic Service Activator on page 11

Dynamic Service Activator in a Redundant Environment

Based on the availability requirements for the Dynamic Service Activator, you can set up Dynamic Service Activator to run in a redundant environment in the following ways:

- Use load-balancing software to manage load for Dynamic Service Activator between two or more instances of it that run on different systems.

The Web application server on each system may already be installed in an environment that uses load-balancing software.

- Install and activate an instance of Dynamic Service Activator on two systems, and configure services to send SOAP requests from client applications to both instances of the Dynamic Service Activators at the same time.

Each SOAP request tries to activate or deactivate the same service session at the same time. This scenario gives better subscriber response time than spaced requests from the service.

For a configuration in which services send SOAP requests to two Dynamic Service Activators, you can optimize the code to wait only for the first call. As soon as one of the SOAP requests is completed, the service application can continue and not wait for the second call to be completed. To detect a failure of the Web Services Gateway or the client's network connection, the application needs to wait for both calls to time out before the request fails.

Duplicate requests and responses places a higher load on the system that runs the service application and on the system running the SAE.

- Related Topics**
- Overview of the Web Services Gateway on page 3
 - Overview of Dynamic Service Activator on page 7
 - Terminology on page 4

CHAPTER 3

Configuring Dynamic Service Activator (SRC CLI)

- Configuration Statements for Dynamic Service Activator on page 11
- Before You Use Dynamic Service Activator on page 14
- Enabling Dynamic Service Activator on a Web Application Server (SRC CLI) on page 15
- Creating Grouped Configurations for Dynamic Service Activator (SRC CLI) on page 15
- Configuring Dynamic Service Activator (SRC CLI) on page 16
- Configuring Local Properties for Dynamic Service Activator (SRC CLI) on page 16
- Configuring Dynamic Service Activator Properties (SRC CLI) on page 19
- Deploying Dynamic Service Activator on the Virtual Host (SRC CLI) on page 33
- Starting Dynamic Service Activator (SRC CLI) on page 34
- Sample Data for Dynamic Service Activator on page 34

Configuration Statements for Dynamic Service Activator

Use the following configuration statements to configure the operating properties for Dynamic Service Activator at the [edit] hierarchy level.

```
slot number dsa {  
    shared shared;  
}  
  
slot number dsa deploy {  
    virtual-host virtual-host;  
}  
  
slot number dsa initial {  
    base-dn base-dn;  
    static-dn static-dn;  
    dynamic-dn dynamic-dn;  
}  
  
slot number dsa initial directory-connection {  
    url url;  
    backup-urls [backup-urls...];  
    principal principal;  
    credentials credentials;  
    protocol protocol;
```

```
    timeout timeout;  
    check-interval check-interval;  
    blacklist;  
    snmp-agent;  
}  
  
slot number dsa initial directory-eventing {  
    eventing;  
    signature-dn signature-dn;  
    polling-interval polling-interval;  
    event-base-dn event-base-dn;  
    dispatcher-pool-size dispatcher-pool-size;  
}
```

Use the following configuration statements to configure Dynamic Service Activator at the [edit] hierarchy level.

shared dsa group *name*

```
shared dsa group name configuration {  
    disable-access-control-mechanism;  
}
```

```
shared dsa group name configuration client name {  
    restricted;  
}
```

```
shared dsa group name configuration client name application application-id {  
    disabled;  
    listener-url listener-url;  
    http-id http-id;  
    http-password http-password;  
    jms-queue-size jms-queue-size;  
}
```

```
shared dsa group name configuration client name application application-id  
event-subscription event-subscription-name {  
    disabled;  
    subject-id subject-id;  
    public-interface-id public-interface-id;  
    event-type-filter [(user-start | user-interim | user-stop | service-start | service-interim  
        | service-stop | interface-start | interface-interim | interface-stop)...];  
    service-name-filter [service-name-filter...];  
    event-filter event-filter;  
    attribute-names [attribute-names...];  
}
```

```
shared dsa group name configuration client name permissions attributes {  
    service [service...];  
    subscription [subscription...];  
    subscriber [subscriber...];  
}
```

shared dsa group *name* configuration client *name* permissions method *name*

shared dsa group *name* configuration client *name* permissions method *name* constraints
argument-index

shared dsa group *name* configuration client *name* permissions script *name*


```

shared dsa group name configuration client name permissions script name constraints
argument-index

shared dsa group name configuration logger name file {
  filter filter;
  filename filename;
  rollover-filename rollover-filename;
  maximum-file-size maximum-file-size;
}

shared dsa group name configuration logger name syslog {
  filter filter;
  host host;
  facility facility;
  format format;
}

shared dsa group name configuration method (allocate-resource | commit-resources |
  invoke-gateway-extension | invoke-script | query-available-services | query-client-status
  | query-contexts | query-status | release-network-resources | release-resource |
  release-resources | reserve-network-resource | subscriber-activate-service |
  subscriber-deactivate-service | subscriber-login | subscriber-logout |
  subscriber-modify-service | subscriber-read-subscription | subscribers-activate-service
  | subscribers-deactivate-service | subscribers-login | subscribers-logout |
  subscribers-modify-service | subscribers-read | subscribers-read-subscriber |
  subscribers-read-subscription) constraints argument-index

shared dsa group name configuration nic-proxy-configuration name

shared dsa group name configuration nic-proxy-configuration name cache {
  cache-size cache-size;
  cache-cleanup-interval cache-cleanup-interval;
  cache-entry-age cache-entry-age;
}

shared dsa group name configuration nic-proxy-configuration name nic-host-selection {
  groups [groups...];
  selection-criteria (roundRobin | randomPick | priorityList);
}

shared dsa group name configuration 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;
}

shared dsa group name configuration nic-proxy-configuration name resolution {
  resolver-name resolver-name;
  key-type key-type;
  value-type value-type;
  expect-multiple-values;
  constraints constraints;
}

shared dsa group name configuration nic-proxy-configuration name test-nic-bindings {
  use-test-bindings;
}

```

```
shared dsa group name configuration nic-proxy-configuration name test-nic-bindings
  key-values name
```

```
shared dsa group name configuration script name
```

```
shared dsa group name configuration script name constraints argument-index
```

```
shared dsa group name configuration session-handle {
  strong-encoding;
  encoding-key encoding-key;
}
```

```
shared dsa group name configuration subscriber-types name {
  subscriber-id-type (address | dn | login-name | interface-name | interface-index |
    address-interface-name | primary-user-name | session-handle | tunnel-session |
    global-address | global-login-name);
  nic-proxy nic-proxy;
}
```

- Related Topics**
- Before You Use Dynamic Service Activator on page 14
 - Configuring Dynamic Service Activator Properties (SRC CLI) on page 19
 - For detailed information about each configuration statement, see the *SRC PE CLI Command Reference*

Before You Use Dynamic Service Activator

Before you use Dynamic Service Activator, you must:

- Deploy a working SRC network.
- Configure and start the application server.
See Configuring the Web Application Server (SRC CLI).
- Configure a NIC that identifies the SAE reference for each subscriber type.
See Configuring the NIC (SRC CLI).
- Create the Java scripts that Dynamic Service Activator invokes (see the SAE CORBA Remote API documentation on the Juniper Networks Web site at <http://www.juniper.net/techpubs/software/management/src/api-index.html>).
- Load the sample data with the **request system ldap load dsa-configuration** command.
See “Sample Data for Dynamic Service Activator” on page 34.

- Related Topics**
- Overview of Dynamic Service Activator on page 7
 - Dynamic Service Activator in a Redundant Environment on page 9
 - Starting Dynamic Service Activator (SRC CLI) on page 34
 - SRC CLI Commands to Monitor Dynamic Service Activator on page 47

Enabling Dynamic Service Activator on a Web Application Server (SRC CLI)

Deploy one copy of Dynamic Service Activator on a Web application server to support all business partners. Before you enable Dynamic Service Activator, you must configure and start the application server.

To enable Dynamic Service Activator, perform the following tasks:

1. Creating Grouped Configurations for Dynamic Service Activator (SRC CLI) on page 15
2. Configuring Local Properties for Dynamic Service Activator (SRC CLI) on page 16
3. Configuring Dynamic Service Activator Properties (SRC CLI) on page 19
4. Deploying Dynamic Service Activator on the Virtual Host (SRC CLI) on page 33
5. Starting Dynamic Service Activator (SRC CLI) on page 34

Creating Grouped Configurations for Dynamic Service Activator (SRC CLI)

You must configure Dynamic Service Activator within a group. Dynamic Service Activator configurations are organized into a hierarchy of groups. Subordinate groups inherit configuration from superior groups.

Configuration groups allow you to share the Dynamic Service Activator configuration with different Dynamic Service Activator instances in the SRC network. You can also set up different configurations for different instances.

You can then create a grouped Dynamic Service Activator configuration that is shared with some Dynamic Service Activator instances. For example, if you create two different Dynamic Service Activator groups called `config1` and `config2` within the shared Dynamic Service Activator configuration, you could select the Dynamic Service Activator configuration that should be associated with a particular Dynamic Service Activator instance.

Use the **shared** option of the **slot *number* dsa** statement to select the group for a Dynamic Service Activator instance as part of the local configuration. Use the **shared dsa group *name* configuration** statements to configure the group.

To select and configure a group:

1. From configuration mode, select a group for a Dynamic Service Activator instance. For example, to select a group called `config1` in the root group:

```
[edit]
user@host# set slot 0 dsa shared /config1
```

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 Dynamic Service Activator configuration.

```
[edit]
user@host# edit shared dsa group config1 ?
Possible completions:
<[Enter]> Execute this command
> configuration
> group Group of configuration properties
| Pipe through a command
```

For more information, see “Configuring Dynamic Service Activator Properties (SRC CLI)” on page 19.

- Related Topics**
- Overview of Dynamic Service Activator on page 7
 - Creating Grouped Configurations for Dynamic Service Activator (C-Web Interface) on page 38
 - Configuring Local Properties for Dynamic Service Activator (SRC CLI) on page 16
 - Starting Dynamic Service Activator (SRC CLI) on page 34
 - Viewing Statistics for Dynamic Service Activator (SRC CLI) on page 48

Configuring Dynamic Service Activator (SRC CLI)

To use Dynamic Service Activator in the SRC network, you configure basic and initial properties, including directory connection and directory eventing properties. You also configure Dynamic Service Activator properties.

For information about these configuration procedures, see:

1. Configuring Local Properties for Dynamic Service Activator (SRC CLI) on page 16
2. Configuring Dynamic Service Activator Properties (SRC CLI) on page 19

Configuring Local Properties for Dynamic Service Activator (SRC CLI)

Configure basic and initial properties, including directory connection and directory eventing properties. Tasks to configure the local properties for Dynamic Service Activator are:

- Configuring Basic Local Properties for Dynamic Service Activator on page 16
- Configuring Initial Properties for Dynamic Service Activator on page 17
- Configuring Initial Directory Connection Properties for Dynamic Service Activator on page 18
- Configuring Initial Directory Eventing Properties for Dynamic Service Activator on page 18

Configuring Basic Local Properties for Dynamic Service Activator

Use the following configuration statements to configure basic local properties for Dynamic Service Activator:

```
slot number dsa {
  shared shared;
}
```

To configure basic local properties:

1. From configuration mode, access the statement that configures the local properties.

```
[edit]
user@host# edit slot 0 dsa
```

2. Specify the configuration namespace for Dynamic Service Activator as the path, relative to the root of the static configuration properties, that defines the object for the namespace.

```
[edit slot 0 dsa]
user@host# set shared shared
```

For example:

```
[edit slot 0 dsa]
user@host# set shared /sample
```

3. (Optional) Verify your configuration.

```
[edit slot 0 dsa]
user@host# show
```

Configuring Initial Properties for Dynamic Service Activator

Use the following configuration statements to configure initial properties for Dynamic Service Activator:

```
slot number dsa initial {
  base-dn base-dn;
  static-dn static-dn;
  dynamic-dn dynamic-dn;
}
```

To configure initial local properties:

1. From configuration mode, access the statement that configures the initial properties.

```
[edit]
user@host# edit slot 0 dsa initial
```

2. Specify the properties for Dynamic Service Activator.

```
[edit slot 0 dsa 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 Initial Directory Connection Properties for Dynamic Service Activator

Use the following configuration statements to configure directory connection properties for Dynamic Service Activator:

```
slot number dsa initial directory-connection {  
    url url;  
    backup-urls [backup-urls...];  
    principal principal;  
    credentials credentials;  
    protocol protocol;  
    timeout timeout;  
    check-interval check-interval;  
    blacklist;  
    snmp-agent;  
}
```

To configure directory connection properties:

1. From configuration mode, access the statement that configures the directory connection properties.

```
[edit]  
user@host# edit slot 0 dsa initial directory-connection
```

2. Specify the properties for Dynamic Service Activator.

```
[edit slot 0 dsa initial directory-connection]  
user@host# set ?
```

Configuring Initial Directory Eventing Properties for Dynamic Service Activator

Use the following configuration statements to configure directory eventing properties for Dynamic Service Activator:

```
slot number dsa 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 statement that configures the local properties.

```
[edit]  
user@host# edit slot 0 dsa initial directory-eventing
```

2. Specify the initial directory eventing properties for Dynamic Service Activator.

```
[edit slot 0 dsa 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](#).

- Related Topics**
- Overview of Dynamic Service Activator on page 7
 - Configuring Dynamic Service Activator Properties (SRC CLI) on page 19
 - Configuring Local Properties for Dynamic Service Activator (C-Web Interface) on page 39
 - Configuring the Test Environment for Dynamic Service Activator Services on page 54
 - Viewing Statistics for Dynamic Service Activator (SRC CLI) on page 48

Configuring Dynamic Service Activator Properties (SRC CLI)

Tasks to configure the Dynamic Service Activator are:

- Configuring General Properties for Dynamic Service Activator on page 19
- Configuring Subscriber Types for Dynamic Service Activator on page 20
- Configuring Session Handles for Dynamic Service Activator on page 21
- Configuring the NIC Proxies for Dynamic Service Activator on page 21
- Configuring Access to Methods and Scripts for Dynamic Service Activator on page 22
- Configuring Access to Methods for Dynamic Service Activator on page 23
- Configuring Access to Scripts for Dynamic Service Activator on page 26
- Restricting Access to Service Sessions for Dynamic Service Activator on page 28
- Configuring Access to Attributes for Dynamic Service Activator on page 28
- Configuring the SAE to Send Tracking Events to Dynamic Service Activator on page 29
- Configuring Dynamic Service Activator to Publish Events to SOAP Applications on page 30
- Configuring the Logging Destinations for Dynamic Service Activator on page 32

Configuring General Properties for Dynamic Service Activator

The general properties for Dynamic Service Activator determine the behavior of the application rather than the relationship between a gateway client and the application.

To configure general properties for Dynamic Service Activator:

1. From configuration mode, access the statement that configures the general properties. In this sample procedure, the properties are configured in the trial group.

[edit]

```
user@host# edit shared dsa group trial configuration
```

2. (Optional) Specify the type of access that gateway clients have to methods and scripts.

```
[edit shared dsa group trial configuration]
```

user@host# set disable-access-control-mechanism

Set this value only if you want gateway clients to have unrestricted access to all methods and scripts. The client still must provide a valid client name and password, and the client name must be configured to access at least one method (for Dynamic Service Activator or PCMM) to access methods of that type. By default, gateway clients have access only to methods and scripts that you specify in the configuration. Access control should be disabled only for troubleshooting purposes.

Configuring Subscriber Types for Dynamic Service Activator

You configure which types of information identify subscribers to the SAE. The subscriber types that you can configure are the same subscriber types that you can use in applications created with the SAE CORBA remote API.

To configure subscriber types:

1. From configuration mode, access the statement that configures the subscriber types. The specified name is used to construct the subscriber's URI. In this sample procedure, the properties are configured in the trial group.

[edit]

user@host# edit shared dsa group trial configuration subscriber-types *name*

2. Specify the type of information used to identify a subscriber.

[edit shared dsa group trial configuration subscriber-types *name*]

user@host# set subscriber-id-type (address | dn | login-name | interface-name | interface-index | address-interface-name | primary-user-name | session-handle | tunnel-session | global-address | global-login-name)

where:

- **address**—Subscriber's IP address
- **dn**—Distinguished name of subscriber profile
- **login-name**—Subscriber's login name
- **interface-name**—Name of the interface and name of the virtual router to which the subscriber connects
- **interface-index**—SNMP index of the interface and name of the virtual router to which the subscriber connects
- **address-interface-name**—Subscriber's IP address, name of the managed interface, and name of the virtual router to which the subscriber connects
- **primary-user-name**—Primary username
- **session-handle**—Subscriber's session handle used to reference an existing subscriber session

- **tunnel-session**—Subscriber's L2TP tunnel session identifier, tunnel identifier, and IP address of L2TP Access Concentrator
 - **global-address**—Subscriber's IP address and VPN identifier
 - **global-login-name**—Subscriber's login name and VPN identifier
3. Specify the namespace that defines the properties for the NIC proxy operations for the specified subscriber ID type. Each subscriber type must use a different NIC proxy.

```
[edit shared dsa group trial configuration subscriber-types name]
user@host# set nic-proxy nic-proxy
```

For example:

```
[edit shared dsa group trial configuration subscriber-types name]
user@host# set nic-proxy ip
```

Configuring Session Handles for Dynamic Service Activator

You configure the encoding key and encoding algorithm for the session handles to determine how the session handle URI is constructed. Session handles are encoded when returned by SOAP calls for the service provider's privacy and to prevent service provider partners who operate SOAP clients from managing subscribers with whom they do not have a relationship.

To configure encoding for session handles:

1. From configuration mode, access the statement that configures the session handles.

```
[edit]
user@host# edit shared dsa group trial configuration session-handle
```

2. Specify the private key to use for encoding a session handle.

```
[edit shared dsa group trial configuration session-handle]
user@host# set encoding-key encoding-key
```

3. (Optional) Specify that the DES algorithm with MD5 hash digested key be used to encode the session handle. If you do not set this value, an exclusive OR algorithm is used.

```
[edit shared dsa group trial configuration session-handle]
user@host# set strong-encoding
```

Configuring the NIC Proxies for Dynamic Service Activator

You create a NIC proxy for each subscriber type to be configured. The name of the NIC proxy must match the name configured for the NIC proxy namespace.

Subscriber types that have different subscriber ID types can use the same NIC proxy. For example, a subscriber type configured as `SubscriberType1` that has a subscriber ID type of `interface-name`, and a subscriber type configured as `subscriberType2` that has a

subscriber ID type of interface-index can both use the same NIC proxy. Likewise, a subscriber type configured as SubscriberType1 and a subscriber type configured as subscriberType2 that both have a subscriber ID type of address can use the same NIC proxy.

To configure NIC proxies:

1. From configuration mode, access the statement that configures the NIC proxy. In this sample procedure, the NIC proxy called ip is configured in the trial group.

[edit]

```
user@host# edit shared dsa group trial configuration nic-proxy-configuration ip
```

2. Specify the properties for the NIC proxy.

[edit shared dsa group trial configuration nic-proxy-configuration ip]

```
user@host# set ?
```

For information about configuring NIC proxies, see Configuration Statements for NIC Proxies.

Configuring Access to Methods and Scripts for Dynamic Service Activator

Configuring access to methods and scripts involves adding methods, scripts, and clients to the configuration and configuring access properties between each client and each method or script.



NOTE: Client profiles are cached by Dynamic Service Activator 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.

When permissions are configured, roles are assigned to application server user objects automatically. The first time you add a method or script for a client, the DSA role is added to the corresponding application server user, and when the last method or script is deleted, the DSA role is removed from the corresponding user. Only role and password changes take up to 30 minutes to take effect.

If you do not want to wait 30 minutes for the changes to take effect, restart the Web application server.

Dynamic Service Activator interacts with the Web application server to determine whether a gateway client has access to a method or script. The name and credentials, such as a password, that are used to authenticate the gateway client are configured on the Web application server as user accounts.

Access constraints are regular expressions that the arguments for the method or script in the SOAP request must match. If the arguments for the method or script in a particular SOAP request do not match these regular expressions, then Dynamic Service Activator rejects the request.

Configuring Access to Methods for Dynamic Service Activator

Use the following configuration statements to configure methods and access properties between each client and each method:

shared dsa group *name* configuration client *name*

shared dsa group *name* configuration client *name* permissions method *name*

shared dsa group *name* configuration client *name* permissions method *name* constraints *argument-index*

shared dsa group *name* configuration method (allocate-resource | commit-resources | invoke-gateway-extension | invoke-script | query-available-services | query-client-status | query-contexts | query-status | release-network-resources | release-resource | release-resources | reserve-network-resource | subscriber-activate-service | subscriber-deactivate-service | subscriber-login | subscriber-logout | subscriber-modify-service | subscriber-read-subscription | subscribers-activate-service | subscribers-deactivate-service | subscribers-login | subscribers-logout | subscribers-modify-service | subscribers-read | subscribers-read-subscriber | subscribers-read-subscription) constraints *argument-index*

Tasks to configure access to methods are:

- Configuring Methods on page 23
- Configuring Access to Methods on page 26

Configuring Methods

To configure methods for Dynamic Service Activator:

1. From configuration mode, access the statement that configures the method to activate on the SAE. Use the text string that exactly matches the name of the method.

[edit]

```
user@host# edit shared dsa group name configuration method (allocate-resource
| commit-resources | invoke-gateway-extension | invoke-script |
query-available-services | query-client-status | query-contexts | query-status |
release-network-resources | release-resource | release-resources |
reserve-network-resource | subscriber-activate-service |
subscriber-deactivate-service | subscriber-login | subscriber-logout |
subscriber-modify-service | subscriber-read-subscription |
subscribers-activate-service | subscribers-deactivate-service | subscribers-login
| subscribers-logout | subscribers-modify-service | subscribers-read |
subscribers-read-subscriber | subscribers-read-subscription)
```

where:

- **allocate-resource**—Specifies the resources that are being requested in the AllocateResource message. This method supports personalized video services.
- **commit-resources**—Specifies the resources that are being requested in the CommitResource message.
- **invoke-gateway-extension**—Invokes a servlet that has been created and deployed in the Web Services Gateway Web application server. The servlet can be a

standalone application, or it can be part of a WAR or EAR file. When deployed, servlets invoked with this method should be accessible only from the local host.

- **invoke-script**—Manages all operations involved with invoking scripts: retrieves requests to invoke scripts from the gateway client, authenticates the gateway client, verifies the arguments supplied by the gateway client, communicates with other SRC components, and returns values to the gateway client.
- **query-available-services**—Searches for the services that are available to the calling application.
- **query-client-status**—Searches for active sessions of a given video server managed by the SRC software. This method supports on-demand video services.
- **query-contexts**—Searches for the context ID and context status for a subscriber.
- **query-status**—Searches for active sessions of a given video server managed by the SRC software. This method supports personalized video services.
- **release-network-resource**—Specifies the resources that are being requested to be released in the ReleaseNetworkResources message. This method supports on-demand video services.
- **release-resource**—Specifies the resources that are being requested to be released in the ReleaseResource message. This method supports personalized video services.
- **release-resources**—Specifies the resources that are being requested to be released in the ReleaseResources message.
- **reserve-network-resource**—Specifies the resources that are being requested in the ReserveNetworkResource message. This method supports on-demand video services.
- **subscriber-activate-service**—Activates subscribers' subscriptions to services.
- **subscriber-deactivate-service**—Deactivates subscribers' subscriptions to services.
- **subscriber-login**—Logs in subscribers. This method supports only subscribers who are identified by their IP addresses. This method does not support subscribers who are identified by the names they use to log in or by their DNs.
- **subscriber-logout**—Logs out subscribers. This method supports only subscribers who are identified by their IP addresses or the names they use to log in. This method does not support subscribers who are identified by their DNs.
- **subscriber-modify-service**—Modifies subscriptions.
- **subscriber-read-subscription**—Determines whether a subscriber accesses services through the SRC owner's network and obtains all of that subscriber's subscriptions.
- **subscribers-activate-service**—Activates subscribers' subscriptions to services. This method supports all subscriber sessions for a given subscriber URI.

- **subscribers-deactivate-service**—Deactivates subscribers' subscriptions to services. This method supports all subscriber sessions for a given subscriber URI.
- **subscribers-login**—Logs in subscribers. This method supports only subscribers who are identified by their IP addresses. This method does not support subscribers who are identified by the names they use to log in or by their DNSs. This method supports all subscriber sessions for a given subscriber URI.
- **subscribers-logout**—Logs out subscribers. This method supports only multiple subscribers who are identified by their IP addresses or the names they use to log in. This method does not support subscribers who are identified by their DNSs. This method supports all subscriber sessions for a given subscriber URI.
- **subscribers-modify-service**—Modifies subscriptions. This method supports all subscriber sessions for a given subscriber URI.
- **subscribers-read**—Reads attributes for the services, subscriber sessions, and service sessions for specific subscribers. This method supports all subscriber sessions for a given subscriber URI.
- **subscribers-read-subscriber**—Reads attributes for the subscriber session. This method supports all subscriber sessions for a given subscriber URI.
- **subscribers-read-subscription**—Determines whether a subscriber accesses services through the SRC owner's network and obtains all of that subscriber's subscriptions. This method supports all subscriber sessions for a given subscriber URI.

For example:

```
user@host# edit shared dsa group trial configuration method
subscriber-read-subscription
```

2. Specify the access constraints applied to the method for all clients.

```
[edit shared dsa group trial configuration method subscriber-read-subscription]
user@host# set constraints argument-index value
```

where:

- ***argument-index***—Zero-based index of the argument used to locate the SAE on which to activate the method
- ***value***—Regular expression

For information about the regular expression syntax, see <http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html>.

For example:

```
user@host# set constraints 1 Audio-[a-zA-Z]*
```

Configuring Access to Methods

To configure access to methods for Dynamic Service Activator:

1. From configuration mode, access the statement that configures the gateway client's access to a method. You must use the same name for the gateway client that is configured on the Web application server.

If you disable the access control mechanism and you configure the Web application server to authenticate clients with any username and password, Dynamic Service Activator sends the text string "anonymous client" as the first argument to the SAE's Java scripts interface module.

```
[edit]
user@host# edit shared dsa group name configuration client name permissions
method name
```

For example:

```
user@host# edit shared dsa group trial configuration client name permissions method
subscriber-read-subscription
```

2. Specify the regular expressions that the method arguments must match for the gateway client.

```
[edit shared dsa group trial configuration client name permissions method
subscriber-read-subscription]
user@host# set constraints argument-index value
```

For example:

```
[edit shared dsa group trial configuration client name permissions method
subscriber-read-subscription]
user@host# set constraints 1 Audio-[a-zA-Z]*
```

Configuring Access to Scripts for Dynamic Service Activator

Use the following configuration statements to configure scripts and access properties between each client and each script:

```
shared dsa group name configuration client name
shared dsa group name configuration client name permissions script name
shared dsa group name configuration client name permissions script name constraints
argument-index
shared dsa group name configuration script name
shared dsa group name configuration script name constraints argument-index
```

Tasks to configure access to scripts are:

- Configuring Scripts on page 27
- Configuring Access to Scripts on page 27

Configuring Scripts

To configure scripts for Dynamic Service Activator:

1. From configuration mode, access the statement that configures the script to activate on the SAE. Use the text string that exactly matches the name of the script.

```
[edit]
user@host# edit shared dsa group name configuration script name
```

2. Specify the zero-based index of the script argument used to locate the SAE on which to invoke the script.

```
[edit shared dsa group name configuration script name]
user@host# set sae-locator-index sae-locator-index
```

3. Specify the access constraints applied to the script for all clients.

```
[edit shared dsa group name configuration script name]
user@host# set constraints argument-index value
```

where:

- *argument-index*—Zero-based index of the argument used to locate the SAE on which to activate the method
- *value*—Regular expression

For information about the regular expression syntax, see <http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html>.

For example:

```
user@host# set constraints 1 Audio-[a-zA-Z]*
```

Configuring Access to Scripts

To configure access to scripts:

1. From configuration mode, access the statement that configures the gateway client's access to a script. You must use the same name for the gateway client that is configured on the Web application server.

If you disable the access control mechanism and you configure the Web application server to authenticate clients with any username and password, Dynamic Service Activator sends the text string "anonymous client" as the first argument to the SAE's Java scripts interface module.

```
[edit]
user@host# edit shared dsa group name configuration client name permissions script name
```

2. Specify the regular expressions that the script arguments must match for the gateway client.

```
[edit shared dsa group name configuration client name permissions script name]
```

```
user@host# set constraints argument-index value
```

For example:

```
[edit shared dsa group trial configuration client name permissions script name]  
user@host# set constraints 1 Audio-[a-zA-Z]*
```

Restricting Access to Service Sessions for Dynamic Service Activator

You can restrict the service sessions to which a gateway client has access. If you do not restrict access, the client has access to all service sessions.

To control the service sessions to which a gateway client has access:

1. From configuration mode, access the statement that configures the gateway client. You must use the same name for the gateway client that is configured on the Web application server.

```
[edit]  
user@host# edit shared dsa group name configuration client name
```

2. (Optional) Specify that access is restricted to the client's own service session.

```
[edit shared dsa group trial configuration client name]  
user@host# set restricted
```

Configuring Access to Attributes for Dynamic Service Activator

You can control the configured attributes to which a gateway client has access. If you do not configure the client's access to attributes, all configured attributes are allowed.

To control the attributes to which a gateway client has access:

1. From configuration mode, access the statement that configures the gateway client's access to attributes. You must use the same name for the gateway client that is configured on the Web application server.

If you disable the access control mechanism, then the client has no restrictions on access to the configured attributes.

```
[edit]  
user@host# edit shared dsa group name configuration client name permissions  
attributes
```

2. (Optional) Specify the service attributes to which the gateway client has access.

```
[edit shared dsa group trial configuration client name permissions attributes]  
user@host# set service [service...]
```

Set this value only if you want gateway clients to have restricted access to configured attributes. By default, all configured attributes are allowed. If you do not want to allow access to any of these attributes, set this value to **none**.

3. (Optional) Specify the subscription attributes to which the gateway client has access.

```
[edit shared dsa group trial configuration client name permissions attributes]
user@host# set subscription [subscription...]
```

Set this value only if you want gateway clients to have restricted access to configured attributes. By default, all configured attributes are allowed. If you do not want to allow access to any of these attributes, set this value to **none**.

4. (Optional) Specify the subscriber attributes to which the gateway client has access.

```
[edit shared dsa group trial configuration client name permissions attributes]
user@host# set subscriber [subscriber...]
```

Set this value only if you want gateway clients to have restricted access to configured attributes. By default, all configured attributes are allowed. If you do not want to allow access to any of these attributes, set this value to **none**.

Configuring the SAE to Send Tracking Events to Dynamic Service Activator

The SAE communicates with Dynamic Service Activator through the Java Message Service (JMS) adapter plug-in. This SAE plug-in sends SAE tracking events to Dynamic Service Activator.

To configure the JMS adapter plug-in:

1. From configuration mode, access the JMS adapter plug-in configuration. In this sample procedure, the JMS adapter plug-in called soapapps is configured in the nw-area SAE group.

```
[edit]
user@host# edit shared sae group nw-area configuration plug-ins name soapapps
jms-adaptor
```

2. Configure the grouped configuration used by all Dynamic Service Activator instances to which this plug-in forwards SAE events.

```
[edit shared sae group nw-area configuration plug-ins name soapapps jms-adaptor]
user@host# set shared-dsa-configuration shared-dsa-configuration
```

3. Specify the Dynamic Server Activator application servers to which the SAE events are published. The URLs reference the JNDI name servers on the application servers.

```
[edit shared dsa group nw-area configuration plug-ins name soapapps jms-adaptor]
user@host# set dsa-application-server-urls [dsa-application-server-urls...]
```

4. (Optional) Specify the SAE plug-in event attributes. The attribute values are the event's subject ID, and they specify a subscriber or interface. The values can be set by the SAE's subscriber classification script. If any of the event attributes contain a value that matches the subject ID in a Dynamic Service Activator event subscription, then the plug-in forwards the event to a Dynamic Service Activator instance.

```
[edit shared sae group nw-area configuration plug-ins name soapapps jms-adaptor]
user@host# set subject-id-attribute-name [subject-id-attribute-name...]
```

If you want to configure JMS adapter plug-in 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.

Configuring Dynamic Service Activator to Publish Events to SOAP Applications

When the SAE sends tracking events to Dynamic Service Activator, Dynamic Service Activator can publish events to external SOAP applications used by content service providers. Events are published according to the configured event subscription. Tasks to configure event subscriptions are:

- Configuring External SOAP Applications on page 30
- Configuring Event Subscriptions on page 31

Configuring External SOAP Applications

Dynamic Service Activator can publish subscriber, service session, and interface events to external SOAP applications.

To configure the external SOAP application to which Dynamic Service Activator can publish events:

1. From configuration mode, access the statement that configures the application to which events are published. You must use the same name for the gateway client that is configured on the Web application server.

```
[edit]
user@host# edit shared dsa group trial configuration client name application
application-id
```

where *application-id* identifies the external SOAP application.

2. (Optional) Specify that sending events to this external SOAP application is disabled.

```
[edit shared dsa group trial configuration client name application application-id]
user@host# set disabled
```

3. Specify the URL of the external SOAP application.

```
[edit shared dsa group trial configuration client name application application-id]
user@host# set listener-url listener-url
```

4. (Optional) If HTTP authentication is required, specify the username Dynamic Service Activator provides to the external SOAP application.

```
[edit shared dsa group trial configuration client name application application-id]
user@host# set http-id http-id
```

5. (Optional) If HTTP authentication is required, specify the password Dynamic Service Activator provides to the external SOAP application.

```
[edit shared dsa group trial configuration client name application application-id]
user@host# set http-password http-password
```

6. (Optional) Specify the size of the queue that holds received SAE events that have not been published yet.

```
[edit shared dsa group trial configuration client name application application-id]
user@host# set jms-queue-size jms-queue-size
```

Configuring Event Subscriptions

You can configure the event subscriptions owned by external SOAP applications. The event subscription defines a set of events, and the attributes in those events, that are published to the external SOAP application.

To configure event subscriptions:

1. From configuration mode, access the statement that configures the event subscription owned by an application.

```
[edit]
user@host# edit shared dsa group trial configuration client name application
application-id event-subscription event-subscription-name
```

where *event-subscription-name* is the arbitrary identifier of the event subscription.

2. (Optional) Specify that this event subscription is disabled.

```
[edit shared dsa group trial configuration client name application application-id
event-subscription event-subscription-name]
user@host# set disabled
```

3. Configure the persistent identifier that specifies the subscriber or interface for which events are published. Only those events associated with the specified subscriber or interface are forwarded to the external SOAP application.

```
[edit shared dsa group trial configuration client name application application-id
event-subscription event-subscription-name]
user@host# set subject-id subject-id
```

4. (Optional) For interface events, configure the identifier for the interface that is published to the external SOAP application instead of the persistent identifier.

```
[edit shared dsa group trial configuration client name application application-id
event-subscription event-subscription-name]
user@host# set public-interface-id public-interface-id
```

5. (Optional) Specify the types of events that Dynamic Service Activator forwards to the external SOAP application. If no event types are specified, all event types are allowed. Only subscriptions for subscriber and service session events can be created by calls to the Dynamic Service Activator's SOAP interface.

```
[edit shared dsa group trial configuration client name application application-id
event-subscription event-subscription-name]
user@host# set event-type-filter [(user-start | user-interim | user-stop | service-start
| service-interim | service-stop | interface-start | interface-interim | interface-stop)...]
```

6. (Optional) Specify the names of services for which Dynamic Service Activator can send service session events to the external SOAP application. If no service names are specified, events for all services are allowed.

```
[edit shared dsa group trial configuration client name application application-id
event-subscription event-subscription-name]
user@host# set service-name-filter [service-name-filter...]
```

7. (Optional) Specify the SAE plug-in events that Dynamic Service Activator can forward to the external SOAP application. This filter allows constraints to be placed on the event attributes. If event attributes do not satisfy the specified constraints, Dynamic Service Activator cannot forward the event to the external SOAP application. If no events are specified, no constraints are applied.

```
[edit shared dsa group trial configuration client name application application-id
event-subscription event-subscription-name]
user@host# set event-filter event-filter
```

8. (Optional) Specify the names of SAE plug-in event attributes that Dynamic Service Activator can forward to the external SOAP application. If no attribute names are specified, all attributes are forwarded.

```
[edit shared dsa group trial configuration client name application application-id
event-subscription event-subscription-name]
user@host# set attribute-names [attribute-names...]
```

Configuring the Logging Destinations for Dynamic Service Activator

Use the following configuration statements to configure logging destinations for Dynamic Service Activator:

```
shared dsa group name configuration logger name
```

```
shared dsa group name configuration logger name file {
  filter filter;
  filename filename;
  rollover-filename rollover-filename;
  maximum-file-size maximum-file-size;
}
```

```
shared dsa group name configuration logger name syslog {
  filter filter;
  host host;
  facility facility;
  format format;
}
```

Tasks to configure the logging destinations are:

- Configuring Logging Destinations to Store Messages in a File on page 32
- Configuring Logging Destinations to Send Messages to the System Logging Facility on page 33

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 statement that configures the name and type of logging destination. In this sample procedure, the logging destination called file-1 is configured in the trial group.

```
[edit]
user@host# edit shared dsa group trial configuration logger file-1 file
```

2. Specify the properties for the logging destination.

```
[edit shared dsa group trial configuration logger file-1 file]
user@host# set ?
```

For more information about configuring properties for the logging destination, see [Configuring a Component to Store Log Messages in a File \(SRC CLI\)](#).

Configuring Logging Destinations to Send Messages to the System Logging Facility

To configure logging destinations to send log messages to the system logging facility:

1. From configuration mode, access the statement that configures the name and type of logging destination. In this sample procedure, the logging destination called syslog-1 is configured in the trial group.

```
[edit]
user@host# edit shared dsa group trial configuration logger syslog-1 syslog
```

2. Specify the properties for the logging destination.

```
[edit shared dsa group trial configuration logger syslog-1 syslog]
user@host# set ?
```

For more information about configuring properties for the logging destination, see [Configuring System Logging \(SRC CLI\)](#).

- Related Topics**
- Overview of Dynamic Service Activator on page 7
 - Configuring Local Properties for Dynamic Service Activator (SRC CLI) on page 16
 - Configuring Dynamic Service Activator Properties (C-Web Interface) on page 40
 - Viewing Information About NIC Proxies (SRC CLI) on page 49
 - Monitoring NIC Proxies (SRC CLI) on page 49

Deploying Dynamic Service Activator on the Virtual Host (SRC CLI)

Use the following configuration statements to deploy Dynamic Service Activator on the virtual host:

```
slot number dsa deploy {
  virtual-host virtual-host;
}
```

To deploy the application on the virtual host:

1. From configuration mode, access the statement that configures the virtual host.

```
[edit]
user@host# edit slot 0 dsa deploy
```

2. Specify the virtual host on which to deploy Dynamic Service Activator. Virtual hosts for Web applications are configured in the application server.

```
[edit slot 0 dsa initial deploy]
user@host# set virtual-host virtual-host
```

- Related Topics**
- Overview of the Web Application Server on C Series Controllers
 - Deploying Dynamic Service Activator on the Virtual Host (C-Web Interface) on page 44
 - Configuring Local Properties for Dynamic Service Activator (SRC CLI) on page 16
 - Configuring Dynamic Service Activator Properties (SRC CLI) on page 19
 - Configuring Virtual Hosts for the Web Applications (SRC CLI)

Starting Dynamic Service Activator (SRC CLI)

You must configure a group before you enable Dynamic Service Activator. To deploy Dynamic Service Activator, you must specify the virtual host, and the Web application server must be running.

To start Dynamic Service Activator:

- From operational mode, enable Dynamic Service Activator.

```
user@host> enable component dsa
```

- Related Topics**
- Overview of Dynamic Service Activator on page 7
 - Deploying Dynamic Service Activator on the Virtual Host (SRC CLI) on page 33
 - Starting Dynamic Service Activator (C-Web Interface) on page 45
 - Creating Grouped Configurations for Dynamic Service Activator (SRC CLI) on page 15
 - Configuring Dynamic Service Activator Properties (SRC CLI) on page 19

Sample Data for Dynamic Service Activator

The SRC software includes sample data that you load with the **request system ldap load dsa-configuration** command. Loading the sample data does not update roles of existing clients. If clients with the names Bob, Joe, or Fred already exist in your configuration, you should use the **request system ldap load dsa-configuration replace** command to replace the existing client configurations with the sample data configuration. The sample Dynamic Service Activator is configured as follows:

- NIC proxy properties—For each subscriber type.
- Permissions—Specifically for each method and script.

Methods, Scripts, and Clients

The sample data shows several methods and one script. The methods are configured as follows:

- Clients Fred and Joe have access with no constraints to the `Subscriber_readSubscription` method.
- Client Joe has access to the `Subscriber_activateService` and the `Subscriber_deactivateService` methods with no constraints. Client Fred, however, can use these methods only to manage services with names that start with “Audio.”
- Client Fred can also use the `Subscriber_modifyService` method for services with names that start with “Audio.” Client Joe cannot use this method, but he can see the `Subscriber_modifyService` method for all services.
- Client Joe is the only client who can use the `Subscriber_login` and `Subscriber_logout` methods.
- Both clients Joe and Fred can use the `invokeGwExtension` method.

The script is configured as follows:

- Client Bob can use the `Echo` method. The script requires that the fourth argument be an IP address that starts with 10 (the IP address has the form 10.x.x.x).
- Client Joe can use the `Echo` method without restriction.

PCMM Available Services

The sample data has two global PCMM services and two PCMM clients configured:

- The two global available services are `News` and `Video-Silver`.
- Client Joe is allowed the `Video-Silver` and `PCMM-Down` service.
- Client Fred is allowed the `Video-Gold` service.
- Client Bob is not allowed any services (except for the global services).

Related Topics

- Loading Sample Data in to a Juniper Networks Database (SRC CLI)
- Overview of Dynamic Service Activator on page 7
- Starting Dynamic Service Activator (SRC CLI) on page 34
- Methods for the Dynamic Service Activator Web Service Interface on page 76
- Running Methods and Scripts for Dynamic Service Activator Services (SRC CLI) on page 57
- Running Methods for PCMM Services (SRC CLI) on page 70

CHAPTER 4

Configuring Dynamic Service Activator (C-Web Interface)

- Enabling Dynamic Service Activator on a Web Application Server (C-Web Interface) on page 37
- Creating Grouped Configurations for Dynamic Service Activator (C-Web Interface) on page 38
- Configuring Dynamic Service Activator (C-Web Interface) on page 39
- Configuring Local Properties for Dynamic Service Activator (C-Web Interface) on page 39
- Configuring Dynamic Service Activator Properties (C-Web Interface) on page 40
- Deploying Dynamic Service Activator on the Virtual Host (C-Web Interface) on page 44
- Starting Dynamic Service Activator (C-Web Interface) on page 45

Enabling Dynamic Service Activator on a Web Application Server (C-Web Interface)

Deploy one copy of Dynamic Service Activator on a Web application server to support all business partners. Before you enable Dynamic Service Activator, you must configure and start the application server.

To enable Dynamic Service Activator:

1. Configure grouped configurations.
See “Creating Grouped Configurations for Dynamic Service Activator (C-Web Interface)” on page 38.
2. Configure local properties.
See “Configuring Local Properties for Dynamic Service Activator (C-Web Interface)” on page 39.
3. Configure Dynamic Service Activator properties.
See “Configuring Dynamic Service Activator Properties (C-Web Interface)” on page 40.
4. Deploy Dynamic Service Activator.

See “Deploying Dynamic Service Activator on the Virtual Host (C-Web Interface)” on page 44.

5. Start Dynamic Service Activator.

See “Starting Dynamic Service Activator (C-Web Interface)” on page 45.

Creating Grouped Configurations for Dynamic Service Activator (C-Web Interface)

You must configure Dynamic Service Activator within a group. Dynamic Service Activator configurations are organized into a hierarchy of groups. Subordinate groups inherit configuration from superior groups. When you create a configuration group, the software creates a configuration with default values filled in.

Configuration groups allow you to share the Dynamic Service Activator configuration with different Dynamic Service Activator instances in the SRC network. You can also set up different configurations for different instances.

You can then create a grouped Dynamic Service Activator configuration that is shared with some Dynamic Service Activator instances. For example, if you create two different Dynamic Service Activator groups called config1 and config2 within the shared Dynamic Service Activator configuration, you could select the Dynamic Service Activator configuration that should be associated with a particular Dynamic Service Activator instance.

To select and configure a group:

1. Click **Configure**, expand **Slot**, expand the slot for which you want to configure the group, and then click **DSA**.

The DSA pane appears.

2. From the Shared list, select a group for a Dynamic Service Activator instance.
3. (Optional) Type a name for the new group in the box below the Shared list using the **/<path>** format, and click **Add**.
4. Click **Apply**.

The group appears in the side pane when you are configuring Dynamic Service Activator properties.

Related Topics

- Overview of Dynamic Service Activator on page 7
- Creating Grouped Configurations for Dynamic Service Activator (SRC CLI) on page 15
- Configuring Dynamic Service Activator Properties (C-Web Interface) on page 40
- Starting Dynamic Service Activator (C-Web Interface) on page 45
- Viewing Statistics for Dynamic Service Activator (C-Web Interface) on page 51

Configuring Dynamic Service Activator (C-Web Interface)

To use Dynamic Service Activator in the SRC network, you configure basic and initial properties, including directory connection and directory eventing properties. You also configure Dynamic Service Activator properties.

For information about these configuration procedures, see:

1. Configuring Local Properties for Dynamic Service Activator (C-Web Interface) on page 39
2. Configuring Dynamic Service Activator Properties (C-Web Interface) on page 40

Configuring Local Properties for Dynamic Service Activator (C-Web Interface)

Configure basic and initial properties, including directory connection and directory eventing properties. Tasks to configure the local properties for Dynamic Service Activator are:

- Configuring Basic Local Properties for Dynamic Service Activator on page 39
- Configuring Initial Properties for Dynamic Service Activator on page 39
- Configuring Initial Directory Connection Properties for Dynamic Service Activator on page 39
- Configuring Initial Directory Eventing Properties for Dynamic Service Activator on page 40

Configuring Basic Local Properties for Dynamic Service Activator

To configure basic local properties:

1. Click **Configure>Slot>Slot:0>DSA**.
2. Enter information as described in the Help text in the main pane, and click **Apply**.

Configuring Initial Properties for Dynamic Service Activator

To configure initial local properties:

1. Click **Configure>Slot>Slot:0>DSA>Initial**.
2. Enter information as described in the Help text in the main pane, and click **Apply**.

Configuring Initial Directory Connection Properties for Dynamic Service Activator

To configure directory connection properties:

1. Click **Configure>Slot>Slot:0>DSA>Initial>Directory Connection**.
The Directory Connection pane appears.
2. Enter information as described in the Help text in the main pane, and click **Apply**.

Configuring Initial Directory Eventing Properties for Dynamic Service Activator

To configure initial directory eventing properties:

1. Click **Configure>Slot>Slot:0>DSA>Initial>Directory Eventing**.
The Directory Eventing pane appears.
2. Enter information as described in the Help text in the main pane, and click **Apply**.

- Related Topics**
- Overview of Dynamic Service Activator on page 7
 - Configuring Local Properties for Dynamic Service Activator (SRC CLI) on page 16
 - Configuring Dynamic Service Activator Properties (C-Web Interface) on page 40
 - Starting Dynamic Service Activator (C-Web Interface) on page 45
 - Viewing Statistics for Dynamic Service Activator (C-Web Interface) on page 51

Configuring Dynamic Service Activator Properties (C-Web Interface)

Tasks to configure the Dynamic Service Activator are:

- Configuring General Properties for Dynamic Service Activator on page 40
- Configuring Subscriber Types for Dynamic Service Activator on page 41
- Configuring the NIC Proxies for Dynamic Service Activator on page 41
- Configuring Access to Methods and Scripts for Dynamic Service Activator on page 42
- Configuring Access to Methods for Dynamic Service Activator on page 42
- Configuring Access to Scripts for Dynamic Service Activator on page 43
- Configuring the Logging Destinations to Store Log Messages in a File on page 43
- Configuring the Logging Destinations to Send Log Messages to the System Logging Facility on page 44

Configuring General Properties for Dynamic Service Activator

The general properties for Dynamic Service Activator determine the behavior of the application rather than the relationship between a gateway client and the application.

To configure general properties for Dynamic Service Activator:

1. Click **Configure>Shared>DSA**, expand the group for which you want to configure general properties, and click **Configuration**.
2. (Optional) To allow gateway clients unrestricted access to all methods and scripts, select the **Disable Access Control Mechanism** check box.

The client still must provide a valid client name and password, and the client name must be configured to access at least one method (for Dynamic Service Activator or PCMM) to access methods of that type. By default, gateway clients have access

only to methods and scripts that you specify in the configuration. Access control should be disabled only for troubleshooting purposes.

Configuring Subscriber Types for Dynamic Service Activator

You configure which types of information identify subscribers to the SAE. The subscriber types that you can configure are the same subscriber types that you can use in applications created with the SAE CORBA remote API.

To configure subscriber types:

1. Click **Configure>Shared>DSA**, expand the group for which you want to configure properties, and expand **Configuration>Subscriber Types**.
2. (Optional) In the Create new list, select **Subscriber Types**.
Type a name for the new configuration in the dialog box, and click **OK**.
The configuration appears in the side pane.
3. Select the Subscriber Types configuration, enter information as described in the Help text in the main pane, and click **Apply**.

Configuring the NIC Proxies for Dynamic Service Activator

You create a NIC proxy for each subscriber type to be configured. The name of the NIC proxy must match the name configured for the NIC proxy namespace.

Subscriber types that have different subscriber ID types can use the same NIC proxy. For example, a subscriber type configured as SubscriberType1 that has a subscriber ID type of interface-name, and a subscriber type configured as subscriberType2 that has a subscriber ID type of interface-index can both use the same NIC proxy. Likewise, a subscriber type configured as SubscriberType1 and a subscriber type configured as subscriberType2 that both have a subscriber ID type of address can use the same NIC proxy.

To configure NIC proxies:

1. Click **Configure>Shared>DSA**, expand the group for which you want to configure properties, and expand **Configuration>NIC Proxy Configuration**.
2. (Optional) In the Create new list, select **NIC Proxies**.
Type a name for the new configuration in the dialog box, and click **OK**.
The configuration appears in the side pane.
3. Expand the NIC Proxies configuration, select the NIC proxy properties you want to configure, enter information as described in the Help text in the main pane, and click **Apply**.

Configuring Access to Methods and Scripts for Dynamic Service Activator

Configuring access to methods and scripts involves adding methods, scripts, and clients to the configuration and configuring access properties between each client and each method or script.



NOTE: Client profiles are cached by Dynamic Service Activator 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.

When permissions are configured, roles are assigned to application server user objects automatically. The first time you add a method or script for a client, the DSA role is added to the corresponding application server user; and when the last method or script is deleted, the DSA role is removed from the corresponding user. Only role and password changes take up to 30 minutes to take effect.

If you do not want to wait 30 minutes for the changes to take effect, restart the Web application server.

Dynamic Service Activator interacts with the Web application server to determine whether a gateway client has access to a method or script. The name and credentials, such as a password, that are used to authenticate the gateway client are configured on the Web application server as user accounts.

Access constraints are regular expressions that the arguments for the method or script in the SOAP request must match. If the arguments for the method or script in a particular SOAP request do not match these regular expressions, then Dynamic Service Activator rejects the request.

Configuring Access to Methods for Dynamic Service Activator

To configure methods for Dynamic Service Activator:

1. Click **Configure>Shared>DSA**, expand the group for which you want to configure properties, and expand **Configuration>Method**.
2. Expand the Method configuration, select the properties you want to configure, enter information as described in the Help text in the main pane, and click **Apply**.

To configure access to methods for Dynamic Service Activator:

1. Click **Configure>Shared>DSA**, expand the group for which you want to configure properties, and expand **Configuration**.
2. (Optional) In the Create new list, select **Client**.

Type a name for the new configuration in the dialog box, and click **OK**. You must use the same name for the gateway client that is configured on the Web application server.

The configuration appears in the side pane.

3. Expand **Client>Permissions**.
4. In the Create new list, select the method you want to configure. Expand the Method configuration, enter information as described in the Help text in the main pane, and click **Apply**.

Configuring Access to Scripts for Dynamic Service Activator

To configure scripts for Dynamic Service Activator:

1. Click **Configure>Shared>DSA**, expand the group for which you want to configure properties, and expand **Configuration**.
2. (Optional) In the Create new list, select **Script**.
Type a name for the new configuration in the dialog box, and click **OK**. Use the text string that exactly matches the name of the script.
The configuration appears in the side pane.
3. Expand the Script configuration, select the properties you want to configure, enter information as described in the Help text in the main pane, and click **Apply**.

To configure access to scripts:

1. Click **Configure>Shared>DSA**, expand the group for which you want to configure properties, and expand **Configuration**.
2. (Optional) In the Create new list, select **Client**.
Type a name for the new configuration in the dialog box, and click **OK**. You must use the same name for the gateway client that is configured on the Web application server.
The configuration appears in the side pane.
3. Expand **Client>Permissions**.
4. In the Create new list, select **Script**.
Type a name for the new configuration in the dialog box, and click **OK**. Use the text string that exactly matches the name of the script.
The configuration appears in the side pane.
5. Expand the Script configuration, enter information as described in the Help text in the main pane, and click **Apply**.

Configuring the Logging Destinations to Store Log Messages in a File

To modify logging destinations to store log messages in a file:

1. Click **Configure>Shared>DSA**, expand the group for which you want to configure logging, and expand **Configuration**.
2. Expand the Logger:file configuration, click **File**, enter information as described in the Help text in the main pane, and click **Apply**.

To create logging destinations to store log messages in a file:

1. Click **Configure>Shared>DSA**, expand the group for which you want to configure logging, and expand **Configuration**.
2. In the Create new list, select **Logger**.
3. Type a name for the new logger in the dialog box, and click **OK**.
The logger appears in the side pane.
4. Expand the Logger configuration, click **File**, enter information as described in the Help text in the main pane, and click **Create**.

Configuring the Logging Destinations to Send Log Messages to the System Logging Facility

To modify logging destinations to send log messages to the system logging facility:

1. Click **Configure>Shared>DSA**, expand the group for which you want to configure logging, and expand **Configuration**.
2. Expand the Logger configuration, click **Syslog**, enter information as described in the Help text in the main pane, and click **Apply**.

To create logging destinations to send log messages to the system logging facility:

1. Click **Configure>Shared>DSA**, expand the group for which you want to configure logging, and expand **Configuration**.
2. In the Create new list, select **Logger**.
3. Type a name for the new logger in the dialog box, and click **OK**.
The logger appears in the side pane.
4. Expand the Logger configuration, click **Syslog**, enter information as described in the Help text in the main pane, and click **Create**.

- Related Topics**
- Overview of Dynamic Service Activator on page 7
 - Configuring Dynamic Service Activator Properties (SRC CLI) on page 19
 - Configuring Local Properties for Dynamic Service Activator (C-Web Interface) on page 39
 - Viewing Information About NIC Proxies (C-Web Interface) on page 52

Deploying Dynamic Service Activator on the Virtual Host (C-Web Interface)

Virtual hosts for Web applications are configured in the application server.

To deploy the application on the virtual host:

1. Click **Configure>Slot>Slot:0>DSA>Deploy**.
2. Enter information as described in the Help text in the main pane, and click **Apply**.

- Related Topics**
- Deploying Dynamic Service Activator on the Virtual Host (SRC CLI) on page 33
 - Starting Dynamic Service Activator (C-Web Interface) on page 45
 - Creating Grouped Configurations for Dynamic Service Activator (C-Web Interface) on page 38
 - Configuring Local Properties for Dynamic Service Activator (C-Web Interface) on page 39
 - Configuring Dynamic Service Activator Properties (C-Web Interface) on page 40
 - Overview of the Web Application Server on C Series Controllers
 - Configuring Virtual Hosts for the Web Applications (SRC CLI)

Starting Dynamic Service Activator (C-Web Interface)

You must configure a group before you enable Dynamic Service Activator. To deploy Dynamic Service Activator, you must specify the virtual host, and the Web application server must be running.

To start Dynamic Service Activator:

1. Click **Manage>Enable**.
The Enable pane appears.
2. From the Component list, select **dsa**, and click **OK**.

- Related Topics**
- Before You Use Dynamic Service Activator on page 14
 - Deploying Dynamic Service Activator on the Virtual Host (C-Web Interface) on page 44
 - Starting Dynamic Service Activator (SRC CLI) on page 34
 - Creating Grouped Configurations for Dynamic Service Activator (C-Web Interface) on page 38
 - Configuring Dynamic Service Activator Properties (C-Web Interface) on page 40

CHAPTER 5

Monitoring Dynamic Service Activator (SRC CLI)

- SRC CLI Commands to Monitor Dynamic Service Activator on page 47
- Output Control Keys for monitor Command on page 48
- Viewing Statistics for Dynamic Service Activator (SRC CLI) on page 48
- Viewing Information About SOAP Operations (SRC CLI) on page 48
- Monitoring SOAP Operations (SRC CLI) on page 49
- Viewing Information About NIC Proxies (SRC CLI) on page 49
- Monitoring NIC Proxies (SRC CLI) on page 49

SRC CLI Commands to Monitor Dynamic Service Activator

You can view statistics for Dynamic Service Activator and information about SOAP operations and NIC proxies. Table 4 on page 47 lists the commands you use to monitor Dynamic Service Activator.

Table 4: Commands to Monitor Dynamic Service Activator

Command	Output Displayed
show dsa statistics general	General statistics for Dynamic Service Activator.
show dsa statistics soap-operation	Information about SOAP operations for Dynamic Service Activator.
monitor dsa soap-operation operation-name	Real-time statistics about SOAP operations for Dynamic Service Activator.
show dsa statistics nic-proxy	Information about NIC proxies for Dynamic Service Activator.
monitor dsa nic-proxy proxy-name	Real-time statistics about NIC proxies for Dynamic Service Activator.

- Related Topics**
- Overview of Dynamic Service Activator on page 7
 - Starting Dynamic Service Activator (SRC CLI) on page 34

- Output Control Keys for monitor Command on page 48
- Viewing Statistics for Dynamic Service Activator (SRC CLI) on page 48

Output Control Keys for monitor Command

The output of the monitor command shows how much each field has changed since you started the command or since you cleared the counters by using the c key. For a description of the statistical information provided in the output of this command, see the show command output.

To control the output of the monitor command while it is running, use the keys listed in Table 5 on page 48. The keys are not case-sensitive.

Table 5: Output Control Keys for the monitor Command

Key	Action
c	Clears (returns to zero) the delta counters since the command was started.
f	Freezes the display, halting the display of updated statistics and delta counters.
q or Esc	Quits the command and returns to the command prompt.
t	Thaws the display, resuming the update of the statistics and delta counters.

- Related Topics**
- SRC CLI Commands to Monitor Dynamic Service Activator on page 47

Viewing Statistics for Dynamic Service Activator (SRC CLI)

Purpose View general statistics for Dynamic Service Activator with the SRC CLI.

Action user@host> **show dsa statistics general**

- Related Topics**
- Overview of Dynamic Service Activator on page 7
 - Starting Dynamic Service Activator (SRC CLI) on page 34

Viewing Information About SOAP Operations (SRC CLI)

Purpose View information about SOAP operations for Dynamic Service Activator.

Action To display information about a specific SOAP operation:

1. Click **Monitor>DSA>Statistics>SOAP Operation**.
The Statistics/SOAP Operation pane appears.

2. From the Operation Name list, select the operation for which you want to display information.
3. Click **OK**.

The Statistics/SOAP Operation pane displays information about the SOAP operation.

- Related Topics**
- Overview of Dynamic Service Activator on page 7
 - Overview of the Web Services Gateway on page 3
 - SOAP Fault Codes for Dynamic Service Activator on page 86
 - Monitoring SOAP Operations (SRC CLI) on page 49

Monitoring SOAP Operations (SRC CLI)

Purpose Display real-time statistics about SOAP operations for Dynamic Service Activator. You can monitor SOAP operations.

Action To display real-time statistics about SOAP operations for Dynamic Service Activator:
user@host> **monitor dsa soap-operation operation-name operation-name**

- Related Topics**
- Overview of Dynamic Service Activator on page 7
 - Overview of the Web Services Gateway on page 3
 - Viewing Information About SOAP Operations (SRC CLI) on page 48
 - Output Control Keys for monitor Command on page 48

Viewing Information About NIC Proxies (SRC CLI)

Purpose View information about NIC proxies for Dynamic Service Activator.

Action To display information about a specific NIC proxy:
user@host> **show dsa statistics nic-proxy proxy-name proxy-name**

- Related Topics**
- For more information about NIC proxies, see the *SRC PE Network Guide*
 - Monitoring NIC Proxies (SRC CLI) on page 49

Monitoring NIC Proxies (SRC CLI)

Purpose Display real-time statistics about NIC proxies for Dynamic Service Activator.

Action To display real-time statistics about NIC proxies for Dynamic Service Activator:
user@host> **monitor dsa nic-proxy proxy-name proxy-name**

- Related Topics**
- For more information about NIC proxies, see the *SRC PE Network Guide*
 - Viewing Information About NIC Proxies (SRC CLI) on page 49

- Output Control Keys for monitor Command on page 48

CHAPTER 6

Monitoring Dynamic Service Activator (C-Web Interface)

- Viewing Statistics for Dynamic Service Activator (C-Web Interface) on page 51
- Viewing Information About SOAP Operations (C-Web Interface) on page 51
- Viewing Information About NIC Proxies (C-Web Interface) on page 52

Viewing Statistics for Dynamic Service Activator (C-Web Interface)

Purpose View general statistics for Dynamic Service Activator.

Action Click **Monitor>DSA>Statistics>General**.

The Statistics/General pane appears.

- Related Topics**
- Overview of Dynamic Service Activator on page 7
 - Viewing Statistics for Dynamic Service Activator (SRC CLI) on page 48
 - Starting Dynamic Service Activator (C-Web Interface) on page 45

Viewing Information About SOAP Operations (C-Web Interface)

Purpose View information about SOAP operations for Dynamic Service Activator.

Action To display information about a specific SOAP operation:

1. Click **Monitor>DSA>Statistics>SOAP Operation**.

The Statistics/SOAP Operation pane appears.

2. From the Operation Name list, select the operation for which you want to display information.
3. Click **OK**.

The Statistics/SOAP Operation pane displays information about the SOAP operation.

- Related Topics**
- Overview of Dynamic Service Activator on page 7

- Overview of the Web Services Gateway on page 3
- Viewing Information About SOAP Operations (SRC CLI) on page 48

Viewing Information About NIC Proxies (C-Web Interface)

Purpose View information about NIC proxies for Dynamic Service Activator.

Action To display information about a specific NIC proxy:

1. Click **Monitor>DSA>Statistics>NIC Proxy**.
The Statistics/NIC Proxy pane appears.
2. From the Proxy Name list, select the NIC proxy for which you want to display information.
3. Click **OK**.
The Statistics/NIC Proxy pane displays information about the NIC proxy.

- Related Topics**
- Viewing Information About NIC Proxies (SRC CLI) on page 49
 - Monitoring NIC Proxies (SRC CLI) on page 49

CHAPTER 7

Testing Dynamic Service Activator (SRC CLI)

- Testing the Web Application Gateway Client on page 53
- Configuring the Test Environment for Dynamic Service Activator Services on page 54
- Running Methods and Scripts for Dynamic Service Activator Services (SRC CLI) on page 57
- Configuring the Test Environment for On-Demand Video Services on page 61
- Running Methods for On-Demand Video Services (SRC CLI) on page 62
- Configuring the Test Environment for Personalized Video Services on page 65
- Running Methods for Personalized Video Services (SRC CLI) on page 66
- Configuring the Test Environment for PCMM Services on page 69
- Running Methods for PCMM Services (SRC CLI) on page 70

Testing the Web Application Gateway Client

The SRC software includes a Web application gateway client that you can use to test Dynamic Service Activator's ability to invoke methods and scripts for:

- Dynamic Service Activator services

For more information about the methods for these services, see “Methods for the Dynamic Service Activator Web Service Interface” on page 76.

- On-demand video services

For more information about the methods for these services, see “Web Service Interface Methods for On-Demand Video Services” on page 92.

- Personalized video services

For more information about the methods for these services, see “Web Service Interface Methods for Personalized Video Services” on page 93.

- PCMM services

For more information about the methods for these services, see “SRC PCMM Web Service Interface Methods” on page 101.

- Related Topics**
- API for Dynamic Service Activator on page 75
 - Configuring the Test Environment for Dynamic Service Activator Services on page 54
 - Running Methods and Scripts for Dynamic Service Activator Services (SRC CLI) on page 57
 - Configuring the Test Environment for On-Demand Video Services on page 61
 - Running Methods for On-Demand Video Services (SRC CLI) on page 62
 - Configuring the Test Environment for Personalized Video Services on page 65
 - Running Methods for Personalized Video Services (SRC CLI) on page 66
 - Configuring the Test Environment for PCMM Services on page 69
 - Running Methods for PCMM Services (SRC CLI) on page 70

Configuring the Test Environment for Dynamic Service Activator Services

Configuring the settings for your test environment is optional. You can choose to enter the settings each time you test the Web client or to configure the test settings. If you choose the latter option, you can avoid repeatedly providing the same information each time you use the client.

- Configuring Settings for Dynamic Service Activator Services (SRC CLI) on page 54
- Configuring the Subscriber URI for Dynamic Service Activator Services (SRC CLI) on page 55
- Verifying Settings for Dynamic Service Activator Services (SRC CLI) on page 56
- Deleting Settings for Dynamic Service Activator Services (SRC CLI) on page 56

Configuring Settings for Dynamic Service Activator Services (SRC CLI)

Use the following command to configure the testing environment for the Dynamic Service Activator Web service interface:

```
test dsa dsa-service environment set <client-id client-id> <client-password  
client-password> <subscriber-id subscriber-id> <subscriber-password  
subscriber-password> <host host> <port port>
```

To configure the settings for the test environment:

1. Issue the **test dsa dsa-service environment set** command.

To configure the settings for the document literal implementation of Dynamic Service Activator (DSA2), issue the **test dsa dsa2-service environment set** command.
2. (Optional) To specify the username that Dynamic Service Activator uses to authenticate this client, use the **client-id** option.
3. (Optional) To specify the password that Dynamic Service Activator uses to authenticate this client, use the **client-password** option.
4. (Optional) To specify the username that the SAE uses to authenticate this subscriber, use the **subscriber-id** option.

5. (Optional) To specify the password that the SAE uses to authenticate this subscriber, use the **subscriber-password** option.
6. (Optional) To specify the hostname of Dynamic Service Activator, use the **host** option. The default value is localhost.
7. (Optional) To specify the port for Dynamic Service Activator, use the **port** option. The default port is 8080.

Configuring the Subscriber URI for Dynamic Service Activator Services (SRC CLI)

Use the following command to configure the subscriber Uniform Resource Identifier (URI) in the testing environment for the Dynamic Service Activator Web service interface:

```
test dsa dsa-service environment set subscriber-uri <subscriber-uri> <subscriber-type
subscriber-type> <subscriber-address subscriber-address> <login-name login-name>
<dn dn> <virtual-router virtual-router> <interface-name interface-name>
<interface-index interface-index> <primary-user-name primary-user-name>
<session-handle session-handle> <subscriber-constraints subscriber-constraints>
```

To configure the settings for the test environment:

1. Issue the **test dsa dsa-service environment set subscriber-uri** command.
To configure the settings for the document literal implementation of Dynamic Service Activator (DSA2), issue the **test dsa dsa2-service environment set subscriber-uri** command.
2. (Optional) To specify the subscriber URI value in the same format as the syntax for the subURI argument, use **subscriber-uri**. See “Format of the Subscriber’s URI” on page 83.
For example: **test dsa dsa-service environment set subscriber-uri ip:ipAddress=1.2.3.4**
3. (Optional) To specify the name of the configured subscriber type used to construct the subscriber URI, use the **subscriber-type** option.
4. (Optional) To specify the IP address of the subscriber, use the **subscriber-address** option. This value is mandatory when the subscriber ID type is **address** or **address-interface-name**.
5. (Optional) To specify the name with which the subscriber logs in, use the **login-name** option. This value is mandatory when the subscriber ID type is **login-name**.
6. (Optional) To specify the DN of the subscriber profile, use the **dn** option. This value is mandatory when the subscriber ID type is **dn**.
7. (Optional) To specify the name of the virtual router associated with the subscriber, use the **virtual-router** option. This value is mandatory when the subscriber ID type is **interface-name** or **interface-index**, optional when the subscriber ID type is **address-interface-name**.
8. (Optional) To specify the name of the interface on which the subscriber logs in, use the **interface-name** option. This value is mandatory when the subscriber ID type is **interface-name** and is optional when the subscriber ID type is **address-interface-name**.

9. (Optional) To specify the SNMP index of the interface on which the subscriber logs in, use the **interface-index** option. This value is mandatory when the subscriber ID type is **interface-index**.
10. (Optional) To specify the primary username, use the **primary-user-name** option. This value is mandatory when the subscriber ID type is **primary-user-name**.
11. (Optional) To specify the subscriber session handle, use the **session-handle** option. This value is mandatory when the subscriber ID type is **session-handle**.
12. (Optional) To specify the constraints for the NIC key, use the **subscriber-constraints** option.

Verifying Settings for Dynamic Service Activator Services (SRC CLI)

Use the following command to verify the settings for the Dynamic Service Activator Web service interface:

```
test dsa dsa-service environment show
```

Use the following commands to verify specific settings:

```
test dsa dsa-service environment show client-id
test dsa dsa-service environment show client-password
test dsa dsa-service environment show host
test dsa dsa-service environment show port
test dsa dsa-service environment show subscriber-id
test dsa dsa-service environment show subscriber-password
test dsa dsa-service environment show subscriber-uri
```

For the document literal implementation of Dynamic Service Activator (DSA2), issue the **test dsa dsa2-service environment show** commands to verify the settings.

Deleting Settings for Dynamic Service Activator Services (SRC CLI)

Use the following command to delete the settings for the Dynamic Service Activator Web service interface:

```
test dsa dsa-service environment clear
```

Use the following commands to delete specific settings:

```
test dsa dsa-service environment clear client-id
test dsa dsa-service environment clear client-password
test dsa dsa-service environment clear host
test dsa dsa-service environment clear port
test dsa dsa-service environment clear subscriber-id
test dsa dsa-service environment clear subscriber-password
test dsa dsa-service environment clear subscriber-uri
```

For the document literal implementation of Dynamic Service Activator (DSA2), issue the **test dsa dsa2-service environment clear** commands to delete the settings.

- Related Topics**
- Testing the Web Application Gateway Client on page 53
 - Sample Data for Dynamic Service Activator on page 34

- Running Methods and Scripts for Dynamic Service Activator Services (SRC CLI) on page 57

Running Methods and Scripts for Dynamic Service Activator Services (SRC CLI)

To run a method or script for the Dynamic Service Activator Web service interface:

- Run the **test dsa dsa-service** command.

```
user@host> test dsa dsa-service (invoke-gateway-extension | invoke-script |
subscriber-activate-service | subscriber-deactivate-service | subscriber-login |
subscriber-logout | subscriber-modify-service | subscriber-read-subscription |
subscribers-activate-service | subscribers-deactivate-service | subscribers-login |
subscribers-logout | subscribers-modify-service | subscribers-read |
subscribers-read-subscriber | subscribers-read-subscription)
```

For the document literal implementation of Dynamic Service Activator (DSA2), issue the **test dsa dsa2-service** command.

```
user@host> test dsa dsa2-service (invoke-gateway-extension | invoke-script |
subscriber-activate-service | subscriber-deactivate-service | subscriber-login |
subscriber-logout | subscriber-modify-service | subscriber-read-subscription |
subscribers-activate-service | subscribers-deactivate-service | subscribers-login |
subscribers-logout | subscribers-modify-service | subscribers-read |
subscribers-read-subscriber | subscribers-read-subscription)
```

where:

- **invoke-gateway-extension**—Invokes a servlet that has been created and deployed in the Web Services Gateway Web application server.
- **invoke-script**—Manages all operations involved with invoking scripts.
- **subscriber-activate-service**—Activates subscribers' subscriptions to services.
- **subscriber-deactivate-service**—Deactivates subscribers' subscriptions to services.
- **subscriber-login**—Logs in subscribers.
- **subscriber-logout**—Logs out subscribers.
- **subscriber-modify-service**—Modifies subscriptions.
- **subscriber-read-subscription**—Determines whether a subscriber accesses services through the SRC owner's network and obtains all of that subscriber's subscriptions.
- **subscribers-activate-service**—Activates subscribers' subscriptions to services for all subscriber sessions.
- **subscribers-deactivate-service**—Deactivates subscribers' subscriptions to services for all subscriber sessions.
- **subscribers-login**—Logs in subscribers.
- **subscribers-logout**—Logs out subscribers.

- **subscribers-modify-service**—Modifies subscriptions for all subscriber sessions.
 - **subscribers-read**—Reads attributes for the services, subscribers, and subscriptions of all subscriber sessions.
 - **subscribers-read-subscriber**—Reads attributes for the subscriber session.
 - **subscribers-read-subscription**—Reads attributes for the subscriber's subscriptions of all subscriber sessions.
-
- Testing Subscriber Logins and Logouts (SRC CLI) on page 58
 - Testing Subscriber Access to Subscriptions (SRC CLI) on page 58
 - Testing Subscription Activations and Deactivations (SRC CLI) on page 59
 - Testing Subscription Modifications (SRC CLI) on page 59
 - Testing Script Invocations (SRC CLI) on page 60
 - Testing Gateway Extension Invocations (SRC CLI) on page 60
 - Testing Access to Attributes for Subscriber Sessions (SRC CLI) on page 60
 - Example: Testing Subscriber Access to Subscriptions on page 61

Testing Subscriber Logins and Logouts (SRC CLI)

These methods support only subscribers who are identified by their IP addresses. These methods do not support subscribers who are identified by the names they use to log in or by their DNs.

Use the following commands to test the methods that log in and log out subscribers:

```
test dsa dsa-service subscriber-login <subscriber-uri subscriber-uri> <subscriber-id  
  subscriber-id> <subscriber-password subscriber-password> <client-id client-id>  
  <client-password client-password> <host host> <port port>  
  
test dsa dsa-service subscriber-logout <subscriber-uri subscriber-uri> <client-id client-id>  
  <client-password client-password> <host host> <port port>  
  
test dsa dsa-service subscribers-login <subscriber-uri subscriber-uri> <subscriber-id  
  subscriber-id> <subscriber-password subscriber-password> <rollback rollback>  
  <client-id client-id> <client-password client-password> <host host> <port port>  
  
test dsa dsa-service subscribers-logout <subscriber-uri subscriber-uri> <client-id client-id>  
  <client-password client-password> <host host> <port port>
```

For the document literal implementation of Dynamic Service Activator (DSA2), issue the **test dsa dsa2-service** commands.

Testing Subscriber Access to Subscriptions (SRC CLI)

Use the following commands to test the method that determines whether a subscriber accesses services through the SRC owner's network and obtains all of that subscriber's subscriptions:

```
test dsa dsa-service subscriber-read-subscription <subscriber-uri subscriber-uri>  
  <attributes attributes> <filter filter> <client-id client-id> <client-password  
  client-password> <host host> <port port>
```

```
test dsa dsa-service subscribers-read-subscription <subscriber-uri subscriber-uri>
<attributes attributes> <filter filter> <client-id client-id> <client-password
client-password> <host host> <port port>
```

For the document literal implementation of Dynamic Service Activator (DSA2), issue the **test dsa dsa2-service** commands.

Testing Subscription Activations and Deactivations (SRC CLI)

Use the following commands to test the methods that activate and deactivate subscribers' subscriptions to services:

```
test dsa dsa-service subscriber-activate-service <subscriber-uri subscriber-uri>
service-name service-name <service-session service-session> <accounting-tag
accounting-tag> <downstream-bandwidth downstream-bandwidth>
<upstream-bandwidth upstream-bandwidth> <session-timeout session-timeout>
<subscription-user subscription-user> <subscription-password subscription-password>
<substitutions substitutions> <client-id client-id> <client-password client-password>
<host host> <port port>

test dsa dsa-service subscriber-deactivate-service <subscriber-uri subscriber-uri>
service-name service-name <service-session service-session> <client-id client-id>
<client-password client-password> <host host> <port port>

test dsa dsa-service subscribers-activate-service <subscriber-uri subscriber-uri>
service-name service-name <service-session service-session> <accounting-tag
accounting-tag> <downstream-bandwidth downstream-bandwidth>
<upstream-bandwidth upstream-bandwidth> <session-timeout session-timeout>
<subscription-user subscription-user> <subscription-password subscription-password>
<substitutions substitutions> <client-id client-id> <client-password client-password>
<host host> <port port>

test dsa dsa-service subscribers-deactivate-service <subscriber-uri subscriber-uri>
service-name service-name <service-session service-session> <client-id client-id>
<client-password client-password> <host host> <port port>
```

For the document literal implementation of Dynamic Service Activator (DSA2), issue the **test dsa dsa2-service** commands.

Testing Subscription Modifications (SRC CLI)

Use the following commands to test the method that modifies subscribers' subscriptions to services:

```
test dsa dsa-service subscriber-modify-service <subscriber-uri subscriber-uri>
service-name service-name <service-session service-session> <accounting-tag
accounting-tag> <downstream-bandwidth downstream-bandwidth>
<upstream-bandwidth upstream-bandwidth> <session-timeout session-timeout>
<subscription-user subscription-user> <subscription-password subscription-password>
<substitutions substitutions> <client-id client-id> <client-password client-password>
<host host> <port port>

test dsa dsa-service subscribers-modify-service <subscriber-uri subscriber-uri>
service-name service-name <service-session service-session> <accounting-tag
accounting-tag> <downstream-bandwidth downstream-bandwidth>
<upstream-bandwidth upstream-bandwidth> <session-timeout session-timeout>
<subscription-user subscription-user> <subscription-password subscription-password>
```

```
<substitutions substitutions> <client-id client-id> <client-password client-password>  
<host host> <port port>
```

For the document literal implementation of Dynamic Service Activator (DSA2), issue the **test dsa dsa2-service** commands.

Testing Script Invocations (SRC CLI)

Use the following command to test the method that manages all operations involved with invoking scripts:

```
test dsa dsa-service invoke-script sae-script-name sae-script-name sae-script-arguments  
sae-script-arguments <client-id client-id> <client-password client-password> <host  
host> <port port>
```

This method retrieves requests to invoke scripts from the gateway client, authenticates the gateway client, verifies the arguments supplied by the gateway client, communicates with other SRC components, and returns values to the gateway client.

For the document literal implementation of Dynamic Service Activator (DSA2), issue the **test dsa dsa2-service invoke-script** command.

Testing Gateway Extension Invocations (SRC CLI)

Use the following command to test the method that invokes a servlet that has been created and deployed in the Web Services Gateway Web application server:

```
test dsa dsa-service invoke-gateway-extension gateway-extension-name  
gateway-extension-name gateway-extension-arguments gateway-extension-arguments  
<client-id client-id> <client-password client-password> <host host> <port port>
```

The servlet can be a standalone application, or it can be part of a WAR or EAR file. When deployed, servlets invoked with this method should be accessible only from the local host.

For the document literal implementation of Dynamic Service Activator (DSA2), issue the **test dsa dsa2-service invoke-gateway-extension** command.

Testing Access to Attributes for Subscriber Sessions (SRC CLI)

Use the following command to test the method that reads attributes for the services, subscribers, and subscriptions of all subscriber sessions:

```
test dsa dsa-service subscribers-read <subscriber-uri subscriber-uri  
<subscription-attributes subscription-attributes <subscription-filter subscription-filter  
<service-attributes service-attributes <service-filter service-filter  
<subscriber-attributes subscriber-attributes <client-id client-id> <client-password  
client-password> <host host> <port port>
```

Use the following command to test the method that reads attributes for a subscriber session:

```
test dsa dsa-service subscribers-read-subscriber <subscriber-uri subscriber-uri  
<subscriber-attributes subscriber-attributes <client-id client-id> <client-password  
client-password>
```


Use the following command to test the method that reads attributes for the subscriber's subscriptions:

```
test dsa dsa-service subscribers-read-subscription <subscriber-uri subscriber-uri>
<attributes attributes> <filter filter> <client-id client-id> <client-password
client-password> <host host> <port port>
```

For the document literal implementation of Dynamic Service Activator (DSA2), issue the **test dsa dsa2-service** commands.

Example: Testing Subscriber Access to Subscriptions

To view a list of the Dynamic Service Activator client's subscriptions:

1. Issue the **test dsa dsa-service subscriber-read-subscription** command.
2. Enter the required information (such as client ID, client password, and subscriber address). The entered data must match the data you configured.

For example, this information is provided for Fred:

```
user@host> test dsa dsa-service subscriber-read-subscription client-id Fred
client-password secret subscriber-address 10.19.1.6 subscriber-type ip attributes
"serviceName"
The Subscriber_readSubscription method was successfully performed for
subURL: ip:ipAddress=10.19.1.6&timestamp=1216221624715
serviceName: CoAService
```

- Related Topics**
- Testing the Web Application Gateway Client on page 53
 - Sample Data for Dynamic Service Activator on page 34
 - Configuring the Test Environment for Dynamic Service Activator Services on page 54

Configuring the Test Environment for On-Demand Video Services

Configuring the settings for your test environment is optional. You can choose to enter the settings in the DSA SOAP Client window each time you use the Web client or to configure the default settings. If you choose the latter option, you can avoid repeatedly providing the same information each time you use the client.

- Configuring Settings for On-Demand Video Services (SRC CLI) on page 61
- Verifying Settings for On-Demand Video Services (SRC CLI) on page 62
- Deleting Settings for On-Demand Video Services (SRC CLI) on page 62

Configuring Settings for On-Demand Video Services (SRC CLI)

Use the following command to configure the testing environment for the Web service interface for on-demand video services:

```
test dsa on-demand-video-service environment set <client-id client-id> <client-password
client-password> <host host> <port port>
```

To configure the settings for the test environment:

1. Issue the **test dsa on-demand-video-service environment set** command.
2. (Optional) To specify the username that Dynamic Service Activator uses to authenticate this client, use the **client-id** option.
3. (Optional) To specify the password that Dynamic Service Activator uses to authenticate this client, use the **client-password** option.
4. (Optional) To specify the hostname of Dynamic Service Activator, use the **host** option. The default value is localhost.
5. (Optional) To specify the port for Dynamic Service Activator, use the **port** option. The default port is 8080.

Verifying Settings for On-Demand Video Services (SRC CLI)

Use the following command to verify the settings for the Web service interface for on-demand video services:

```
test dsa on-demand-video-service environment show
```

Use the following commands to verify specific settings:

```
test dsa on-demand-video-service environment show client-id
test dsa on-demand-video-service environment show client-password
test dsa on-demand-video-service environment show host
test dsa on-demand-video-service environment show port
```

Deleting Settings for On-Demand Video Services (SRC CLI)

Use the following commands to delete the settings for the Web service interface for on-demand video services:

```
test dsa on-demand-video-service environment clear
```

Use the following commands to delete specific settings:

```
test dsa on-demand-video-service environment clear client-id
test dsa on-demand-video-service environment clear client-password
test dsa on-demand-video-service environment clear host
test dsa on-demand-video-service environment clear port
```

- Related Topics**
- Testing the Web Application Gateway Client on page 53
 - Running Methods for On-Demand Video Services (SRC CLI) on page 62

Running Methods for On-Demand Video Services (SRC CLI)

To run a method for the Web service interface for on-demand video services:

- Run the **test dsa on-demand-video-service** command.

```
user@host> test dsa on-demand-video-service (query-client-status |
release-network-resources | reserve-network-resource)
```

where:

- **query-client-status**—Searches for active sessions of a given video server managed by the SRC software.
- **release-network-resources**—Specifies the resources that are being requested to be released in the ReleaseNetworkResources message.
- **reserve-network-resource**—Specifies the resources that are being requested in the ReserveNetworkResource message.
- Testing Queries for Active Sessions (SRC CLI) on page 63
- Testing Resource Release Requests (SRC CLI) on page 63
- Testing Resource Reservation Requests (SRC CLI) on page 64

Testing Queries for Active Sessions (SRC CLI)

Use the following command to test the method that searches for the active sessions of a given video server:

```
test dsa on-demand-video-service query-client-status reservation-client-ipv4  
reservation-client-ipv4 <client-id client-id> <client-password client-password> <host  
host> <port port>
```

To test the method:

1. Issue the **test dsa on-demand-video-service query-client-status** command.
2. To specify the IP address of the video server, use the **reservation-client-ipv4** option.
3. (Optional) To specify the username that Dynamic Service Activator uses to authenticate this client, use the **client-id** option.
4. (Optional) To specify the password that Dynamic Service Activator uses to authenticate this client, use the **client-password** option.
5. (Optional) To specify the hostname of Dynamic Service Activator, use the **host** option. The default value is localhost.
6. (Optional) To specify the port for Dynamic Service Activator, use the **port** option. The default port is 8080.

Testing Resource Release Requests (SRC CLI)

Use the following command to test the method that specifies the resources that are being requested for release in the ReleaseNetworkResources message:

```
test dsa on-demand-video-service release-network-resources session-id  
session-id session-id source-ipv4 source-ipv4 destination-ipv4 destination-ipv4 <client-id  
client-id> <client-password client-password> <host host> <port port>
```

To test the method:

1. Issue the **test dsa on-demand-video-service release-network-resources** command.
2. To specify the globally unique identifier generated by the video server, use the **session-id** option.

3. To specify the IP address of the stream origin, use the **source-ipv4** option. The stream origin is the video server.
4. To specify the IP address of the stream destination, use the **destination-ipv4** option. The stream destination is the subscriber.
5. (Optional) To specify the username that Dynamic Service Activator uses to authenticate this client, use the **client-id** option.
6. (Optional) To specify the password that Dynamic Service Activator uses to authenticate this client, use the **client-password** option.
7. (Optional) To specify the hostname of Dynamic Service Activator, use the **host** option. The default value is localhost.
8. (Optional) To specify the port for Dynamic Service Activator, use the **port** option. The default port is 8080.

Testing Resource Reservation Requests (SRC CLI)

Use the following command to test the method that specifies the resources that are being requested in the ReserveNetworkResource message:

```
test dsa on-demand-video-service reserve-network-resource session-id session-id bit-rate  
bit-rate reservation-client-ipv4 reservation-client-ipv4 policy-manager-client-url  
policy-manager-client-url source-ipv4 source-ipv4 destination-ipv4 destination-ipv4  
<client-id client-id> <client-password client-password> <host host> <port port>
```

To test the method:

1. Issue the **test dsa on-demand-video-service reserve-network-resource** command.
2. To specify the globally unique identifier generated by the video server, use the **session-id** option.
3. To specify the requested bandwidth, use the **bit-rate** option.
4. To specify the IP address of the video server, use the **reservation-client-ipv4** option.
5. To specify the URL that the policy manager uses to initiate requests to the client, use the **policy-manager-client-url** option.
6. To specify the IP address of the stream origin, use the **source-ipv4** option. The stream origin is the video server.
7. To specify the IP address of the stream destination, use the **destination-ipv4** option. The stream destination is the subscriber.
8. (Optional) To specify the username that Dynamic Service Activator uses to authenticate this client, use the **client-id** option.
9. (Optional) To specify the password that Dynamic Service Activator uses to authenticate this client, use the **client-password** option.

10. (Optional) To specify the hostname of Dynamic Service Activator, use the **host** option. The default value is localhost.
11. (Optional) To specify the port for Dynamic Service Activator, use the **port** option. The default port is 8080.

- Related Topics**
- Testing the Web Application Gateway Client on page 53
 - Configuring the Test Environment for On-Demand Video Services on page 61

Configuring the Test Environment for Personalized Video Services

Configuring the settings for your test environment is optional. You can choose to enter the settings in the DSA SOAP Client window each time you use the Web client or to configure the default settings. If you choose the latter option, you can avoid repeatedly providing the same information each time you use the client.

- Configuring Settings for Personalized Video Services (SRC CLI) on page 65
- Verifying Settings for Personalized Video Services (SRC CLI) on page 65
- Deleting Settings for Personalized Video Services (SRC CLI) on page 66

Configuring Settings for Personalized Video Services (SRC CLI)

Use the following command to configure the testing environment for the Web service interface for personalized video services:

```
test dsa personalized-video-service environment set <client-id client-id> <client-password  
client-password> <host host> <port port>
```

To configure the settings for the test environment:

1. Issue the **test dsa personalized-video-service environment set** command.
2. (Optional) To specify the username that Dynamic Service Activator uses to authenticate this client, use the **client-id** option.
3. (Optional) To specify the password that Dynamic Service Activator uses to authenticate this client, use the **client-password** option.
4. (Optional) To specify the hostname of Dynamic Service Activator, use the **host** option. The default value is localhost.
5. (Optional) To specify the port for Dynamic Service Activator, use the **port** option. The default port is 8080.

Verifying Settings for Personalized Video Services (SRC CLI)

Use the following command to verify the settings for the Web service interface for personalized video services:

```
test dsa personalized-video-service environment show
```

Use the following commands to verify specific settings:

```
test dsa personalized-video-service environment show client-id
test dsa personalized-video-service environment show client-password
test dsa personalized-video-service environment show host
test dsa personalized-video-service environment show port
```

Deleting Settings for Personalized Video Services (SRC CLI)

Use the following commands to delete the settings for the Web service interface for personalized video services:

```
test dsa personalized-video-service environment clear
```

Use the following commands to delete specific settings:

```
test dsa personalized-video-service environment clear client-id
test dsa personalized-video-service environment clear client-password
test dsa personalized-video-service environment clear host
test dsa personalized-video-service environment clear port
```

- Related Topics**
- Testing the Web Application Gateway Client on page 53
 - Running Methods for Personalized Video Services (SRC CLI) on page 66

Running Methods for Personalized Video Services (SRC CLI)

To run a method for the Web service interface for personalized video services:

- Run the **test dsa personalized-video-service** command.

```
user@host> test dsa personalized-video-service (allocate-resource | query-status |
release-resource)
```

where:

- **allocate-resource**—Specifies the resources that are being requested in the AllocateResource message.
 - **query-status**—Searches for active sessions of a given video server managed by the SRC software.
 - **release-resource**—Specifies the resources that are being requested to be released in the ReleaseResource message.
- Testing Resource Allocation Requests (SRC CLI) on page 66
 - Testing Queries for Active Sessions (SRC CLI) on page 68
 - Testing Resource Release Requests (SRC CLI) on page 68

Testing Resource Allocation Requests (SRC CLI)

Use the following command to test the method that specifies the resources that are being requested in the AllocateResource message:

```
test dsa personalized-video-service allocate-resource application-name application-name
session-id session-id bit-rate bit-rate signaling-server signaling-server media-source-ip
media-source-ip destination-ip destination-ip message-version message-version
```

<mac-address *mac-address*> <media-type *media-type*> <content-type *content-type*>
 <priority *priority*> <duration *duration*> <trick-file-duration-delay *trick-file-duration-delay*>
 <client-id *client-id*> <client-password *client-password*> <host *host*> <port *port*>

To test the method:

1. Issue the **test dsa personalized-video-service allocate-resource** command.
2. To specify the application generating the request, use the **application-name** option. The application name maps to the service name.
3. To specify the globally unique identifier generated by the video server, use the **session-id** option.
4. To specify the requested bandwidth, use the **bit-rate** option.
5. To specify the fully qualified domain name of the signaling server, use the **signaling-server** option.
6. To specify the IP address of the video stream source, use the **media-source-ip** option.
7. To specify the IP address of the video stream destination, use the **destination-ip** option. The video stream destination is the subscriber.
8. To specify the version of the message type, use the **message-version** option.
9. (Optional) To specify the MAC address of the originating device, use the **mac-address** option.
10. (Optional) To specify the media type, use the **media-type** option. For the personalized video service, the media type is **video**.
11. (Optional) To specify the content type, use the **content-type** option. For the personalized video service, the content type is **sd** or **hd**.
12. (Optional) To specify the priority, use the **priority** option. For the personalized video service, the priority is **free** or **pay**.
13. (Optional) To specify the duration of the playout for the media type, use the **duration** option.
14. (Optional) To specify the maximum time the media type can be in the pause condition, use the **trick-file-duration-delay** option.
15. (Optional) To specify the username that Dynamic Service Activator uses to authenticate this client, use the **client-id** option.
16. (Optional) To specify the password that Dynamic Service Activator uses to authenticate this client, use the **client-password** option.
17. (Optional) To specify the hostname of Dynamic Service Activator, use the **host** option. The default value is localhost.
18. (Optional) To specify the port for Dynamic Service Activator, use the **port** option. The default port is 8080.

Testing Queries for Active Sessions (SRC CLI)

Use the following command to test the method that searches for the active sessions of a given video server:

```
test dsa personalized-video-service query-status application-name application-name  
application-server application-server message-version message-version <client-id client-id  
> <client-password client-password <host host <port port >
```

To test the method:

1. Issue the **test dsa personalized-video-service query-status** command.
2. To specify the application generating the request, use the **application-name** option. The application name maps to the service name.
3. To specify the IP address of the video server, use the **application-server** option.
4. To specify the version of the message type, use the **message-version** option.
5. (Optional) To specify the username to authenticate this client, use the **client-id** option.
6. (Optional) To specify the password to authenticate this client, use the **client-password** option.
7. (Optional) To specify the hostname, use the **host** option. The default value is localhost.
8. (Optional) To specify the port, use the **port** option. The default port is 8080.

Testing Resource Release Requests (SRC CLI)

Use the following command to test the method that specifies the resources that are being requested for release in the ReleaseResource message:

```
test dsa personalized-video-service release-resource application-name application-name  
session-id session-id message-version message-version <client-id client-id  
> <client-password client-password <host host <port port >
```

To test the method:

1. Issue the **test dsa personalized-video-service release-resource** command.
2. To specify the application generating the request, use the **application-name** option. The application name maps to the service name.
3. To specify the globally unique identifier generated by the video server, use the **session-id** option.
4. To specify the version of the message type, use the **message-version** option.
5. (Optional) To specify the username that Dynamic Service Activator uses to authenticate this client, use the **client-id** option.
6. (Optional) To specify the password that Dynamic Service Activator uses to authenticate this client, use the **client-password** option.

7. (Optional) To specify the hostname of Dynamic Service Activator, use the **host** option. The default value is localhost.
8. (Optional) To specify the port for Dynamic Service Activator, use the **port** option. The default port is 8080.

- Related Topics**
- Testing the Web Application Gateway Client on page 53
 - Configuring the Test Environment for Personalized Video Services on page 65

Configuring the Test Environment for PCMM Services

Configuring the settings for your test environment is optional. You can choose to enter the settings in the DSA SOAP Client window each time you use the Web client or to configure the default settings. If you choose the latter option, you can avoid repeatedly providing the same information each time you use the client.

- Configuring Settings for PCMM Services (SRC CLI) on page 69
- Verifying Settings for PCMM Services (SRC CLI) on page 69
- Deleting Settings for PCMM Services (SRC CLI) on page 70

Configuring Settings for PCMM Services (SRC CLI)

Use the following command to configure the testing environment for the PCMM Web service interface:

```
test dsa pcmm-service environment set <client-id client-id> <client-password
client-password> <subscriber-address subscriber-address> <subscriber-uri
subscriber-uri>
```

To configure the settings for the test environment:

1. Issue the **test dsa pcmm-service environment set** command.
2. (Optional) To specify the username that Dynamic Service Activator uses to authenticate this client, use the **client-id** option.
3. (Optional) To specify the password that Dynamic Service Activator uses to authenticate this client, use the **client-password** option.
4. (Optional) To specify the IP address of the subscriber, use the **subscriber-address** option.
5. (Optional) To specify the subscriber's Uniform Resource Identifier (URI), use the **subscriber-uri** option.

Verifying Settings for PCMM Services (SRC CLI)

Use the following command to verify the settings for the PCMM Web service interface:

```
test dsa pcmm-service environment show
```

Use the following commands to verify specific settings:

```
test dsa pcmm-service environment show client-id
test dsa pcmm-service environment show client-password
test dsa pcmm-service environment show subscriber-address
test dsa pcmm-service environment show subscriber-uri
```

Deleting Settings for PCMM Services (SRC CLI)

Use the following commands to delete the settings for the PCMM Web service interface:

```
test dsa pcmm-service environment clear
```

Use the following commands to delete specific settings:

```
test dsa pcmm-service environment clear client-id
test dsa pcmm-service environment clear client-password
test dsa pcmm-service environment clear subscriber-address
test dsa pcmm-service environment clear subscriber-uri
```

- Related Topics**
- Testing the Web Application Gateway Client on page 53
 - Sample Data for Dynamic Service Activator on page 34
 - Configuring the Test Environment for Dynamic Service Activator Services on page 54
 - Running Methods for PCMM Services (SRC CLI) on page 70

Running Methods for PCMM Services (SRC CLI)

To run a method for the PCMM Web service interface:

- Run the **test dsa pcmm-service** command.

```
user@host> test dsa pcmm-service (commit-resources | release-resources |
query-contexts | query-available-services)
```

where:

- **commit-resources**—Specifies the resources that are being requested in the CommitResource message.
 - **release-resources**—Specifies the resources that are being requested to be released in the ReleaseResources message.
 - **query-contexts**—Searches for the context ID and context status for a subscriber.
 - **query-available-services**—Searches for the services that are available to the calling application.
- Testing Resource Requests (SRC CLI) on page 71
 - Testing Resource Release Requests (SRC CLI) on page 71
 - Testing Queries for Subscriber Contexts (SRC CLI) on page 72
 - Testing Queries for Available Services (SRC CLI) on page 72

Testing Resource Requests (SRC CLI)

Use the following command to test the method that specifies the resources that are being requested in the CommitResource message:

```
test dsa pcmm-service commit-resources <subscriber-address subscriber-address>
<subscriber-uri subscriber-uri> service-name service-name <context-id context-id>
<time-usage-limit time-usage-limit> <classifier classifier> <traffic-profile traffic-profile>
<flow-spec flow-spec> <client-id client-id> <client-password client-password>
```

To test the method:

1. Issue the **test dsa pcmm-service commit-resources** command.
2. To specify the name of the SRC service, use the **service-name** option.
3. (Optional) To specify the IP address of the subscriber, use the **subscriber-address** option.
4. (Optional) To specify the subscriber's Uniform Resource Identifier (URI), use the **subscriber-uri** option.
5. (Optional) To specify the globally unique identifier that the application manager must use if it is included in the message, use the **context-id** option. The context ID is used as the SRC session name when the PCMM gateway activates a service.
6. (Optional) To specify a limit on the lifetime of a context, use the **time-usage-limit** option. An application server may specify multiple time usage limits to request different limits in the upstream and downstream directions. If the application server does not specify a time usage limit, the application manager determines the time usage limit.
7. (Optional) To specify the object that identifies the traffic flow for which the application server is requesting services, use the **classifier** option.
8. (Optional) To specify information about the bandwidth and QoS characteristics desired for a request, use the **traffic-profile** option. You express the traffic profile by configuring the SRC policies with FlowSpec, service class name, or DOCSIS actions.
9. (Optional) To specify a FlowSpec action, use the **flow-spec** option.
10. (Optional) To specify the username that Dynamic Service Activator uses to authenticate this client, use the **client-id** option.
11. (Optional) To specify the password that Dynamic Service Activator uses to authenticate this client, use the **client-password** option.

Testing Resource Release Requests (SRC CLI)

Use the following command to test the method that specifies the resources that are being requested to be released in the ReleaseResources message:

```
test dsa pcmm-service release-resources <subscriber-address subscriber-address>
<subscriber-uri subscriber-uri> <service-name service-name> <context-id context-id>
<client-id client-id> <client-password client-password>
```

To test the method:

1. Issue the **test dsa pcmm-service release-resources** command.
2. (Optional) To specify the IP address of the subscriber, use the **subscriber-address** option.
3. (Optional) To specify the subscriber's Uniform Resource Identifier (URI), use the **subscriber-uri** option.
4. (Optional) To specify the name of the SRC service, use the **service-name** option.
5. (Optional) To specify the globally unique identifier that the application manager must use if it is included in the message, use the **context-id** option. The context ID is used as the SRC session name when the PCMM gateway activates a service.
6. (Optional) To specify the username that Dynamic Service Activator uses to authenticate this client, use the **client-id** option.
7. (Optional) To specify the password that Dynamic Service Activator uses to authenticate this client, use the **client-password** option.

Testing Queries for Subscriber Contexts (SRC CLI)

Use the following command to test the method that searches for the context ID and context status for a subscriber:

```
test dsa pcmm-service query-contexts <subscriber-address subscriber-address>  
  <subscriber-uri subscriber-uri> <service-name service-name> <context-id context-id>  
  <client-id client-id> <client-password client-password>
```

To test the method:

1. Issue the **test dsa pcmm-service query-contexts** command.
2. (Optional) To specify the IP address of the subscriber, use the **subscriber-address** option.
3. (Optional) To specify the subscriber's URI, use the **subscriber-uri** option.
4. (Optional) To specify the name of the SRC service, use the **service-name** option.
5. (Optional) To specify the globally unique identifier that the application manager must use if it is included in the message, use the **context-id** option. The context ID is used as the SRC session name when the PCMM gateway activates a service.
6. (Optional) To specify the username that Dynamic Service Activator uses to authenticate this client, use the **client-id** option.
7. (Optional) To specify the password that Dynamic Service Activator uses to authenticate this client, use the **client-password** option.

Testing Queries for Available Services (SRC CLI)

Use the following command to test the method that searches for the services that are available to the calling application:

```
test dsa pcmm-service query-available-services <client-id client-id> <client-password  
  client-password>
```

To test the method:

1. Issue the **test dsa pcmm-service query-available-services** command.
2. (Optional) To specify the username that Dynamic Service Activator uses to authenticate this client, use the **client-id** option.
3. (Optional) To specify the password that Dynamic Service Activator uses to authenticate this client, use the **client-password** option.

- Related Topics**
- Testing the Web Application Gateway Client on page 53
 - Sample Data for Dynamic Service Activator on page 34
 - Running Methods and Scripts for Dynamic Service Activator Services (SRC CLI) on page 57
 - Configuring the Test Environment for PCMM Services on page 69

CHAPTER 8

Developing Gateway Clients

- API for Dynamic Service Activator on page 75
- Methods for the Dynamic Service Activator Web Service Interface on page 76
- Format of the Subscriber's URI on page 83
- Subscription Attributes on page 85
- SOAP Fault Codes for Dynamic Service Activator on page 86

API for Dynamic Service Activator

This topic contains information that developers need to create gateway clients and that administrators need to manage gateway clients and their interactions with the gateway.

Public SOAP Interfaces of Web Applications

When you have installed Dynamic Service Activator, you can access a Web Services Description Language (WSDL) file for the application. The WSDL file defines the SOAP properties that you or your customers can use to develop a gateway client. The URL for Dynamic Service Activator is:

<http://<host>:<portNumber>:/dsa/services/DynamicServiceActivation?wsdl>

- **<host>**—IP address or name of the host that supports Dynamic Service Activator
- **<portNumber>**—Number of the TCP port

There is an implementation of Dynamic Service Activator that uses the document literal encoding style (DSA2). The URL for this implementation is:

<http://<host>:<portNumber>:/dsa/services/DSA2?wsdl>

- **<host>**—IP address or name of the host that supports Dynamic Service Activator
- **<portNumber>**—Number of the TCP port

Related Topics

- Overview of Dynamic Service Activator on page 7
- Overview of the Web Services Gateway on page 3
- Sample Data for Dynamic Service Activator on page 34
- Methods for the Dynamic Service Activator Web Service Interface on page 76

- SOAP Fault Codes for Dynamic Service Activator on page 86

Methods for the Dynamic Service Activator Web Service Interface

This topic describes the methods associated with Dynamic Service Activator, and provides information additional to that in the WSDL file.

invokeGwExtension

- Invokes a servlet that has been created and deployed in the Web Services Gateway Web application server. The servlet can be a standalone application, or it can be part of a WAR or EAR file.

When deployed, servlets invoked with `invokeGwExtension` should be accessible only from the local host.

- Arguments
 - `extensionName`—String that contains the name of the servlet that the gateway client invokes
 - `extensionArguments`—String array of arguments that the gateway client passes to the servlet
- Guidelines—The names in the following components and properties must be the same as the name of the `extensionName` argument:
 - The name of the WAR file that is the gateway extension servlet
 - In the `WEB-INF/web.xml` file in the servlet section, the servlet name
 - In the `WEB-INF/web.xml` file in the servlet-mapping section, the URL pattern in the format `/servlet/<extensionName>`
- Expected output—String returned by the extension
- SOAP fault codes—See “SOAP Fault Codes for Dynamic Service Activator” on page 86

invokeScript

- Manages all operations involved with invoking scripts: retrieves requests to invoke scripts from the gateway client, authenticates the gateway client, verifies the arguments supplied by the gateway client, communicates with other SRC components, and returns values to the gateway client. For a complete description of Dynamic Service Activator's interactions with the gateway client and other components, see “Overview of Dynamic Service Activator” on page 7.
- Arguments
 - `scriptName`—String that contains the name of the script that the gateway client wants to invoke
 - `scriptArgs`—String array of arguments that the gateway client passes to the script

- Expected output—String returned by the script
- SOAP fault codes—See “SOAP Fault Codes for Dynamic Service Activator” on page 86

Subscriber_readSubscription

- Determines whether a subscriber accesses services through the SRC owner's network and obtains all of that subscriber's subscriptions; returns the result in a two-dimensional array.
- Arguments
 - subURI—String that contains the subscriber's URI (see “Format of the Subscriber's URI” on page 83)
 - select—Similar to a SQL select statement. Use a filter string for the first field of the select to indicate the subscriptions. Use a list of attribute names for the second field to indicate the subscription attributes.

For more information about how to specify the filter and attributes in a select argument, see the SAE CORBA Remote API documentation on the Juniper Networks Web site at

<http://www.juniper.net/techpubs/software/management/sdx/api-index.html> (the sae.Select structure).

- Expected output—Multidimensional array of Attr objects that contain the subscriptions for the subscriber
- SOAP fault codes—See “SOAP Fault Codes for Dynamic Service Activator” on page 86

Subscriber_readSubscription_retAttrSeq

- Determines whether a subscriber accesses services through the SRC owner's network and obtains all of that subscriber's subscriptions; returns the result in a one-dimensional array for SOAP clients that do not support two-dimensional arrays.
- Arguments
 - subURI—String that contains the subscriber's URI (see “Format of the Subscriber's URI” on page 83)
 - select—Similar to a SQL select statement. Use a filter string for the first field of the select to indicate the subscriptions. Use a list of attribute names for the second field to indicate the subscription attributes.

For more information about how to specify the filter and attributes in a select argument, see the SAE CORBA Remote API documentation on the Juniper Networks Web site at

<http://www.juniper.net/techpubs/software/management/sdx/api-index.html> (the sae.Select structure).

- Expected output—One-dimensional array of AttrSeq objects that contain the subscriptions for the subscriber
- SOAP fault codes—See “SOAP Fault Codes for Dynamic Service Activator” on page 86

Subscriber_activateService

- Activates subscribers' subscriptions to services.
 - Arguments
 - subURI—String that contains the subscriber's URI (see "Format of the Subscriber's URI" on page 83)
 - subscriptionName—String that contains the name of the subscription
 - sessionName—String that contains the name of the service session; default string is *default*
 - activationAttributes—Array of one or more of the following attributes that can be specified for the subscription:
 - sessionTimeout
 - downstreamBandwidth
 - upstreamBandwidth
 - sessionTag
 - subscriptionUsername
 - subscriptionPassword
 - substitutions
- For information about these attributes, see "Subscription Attributes" on page 85.
- Expected output—None
 - SOAP fault codes—See "SOAP Fault Codes for Dynamic Service Activator" on page 86

Subscriber_deactivateService

- Deactivates subscribers' subscriptions to services.
- Arguments
 - subURI—String that contains the subscriber's URI (see "Format of the Subscriber's URI" on page 83)
 - subscriptionName—String that contains the name of the subscription
 - sessionName—String that contains the name of the service session; default string is *default*
- Expected output—None
- SOAP fault codes—See "SOAP Fault Codes for Dynamic Service Activator" on page 86

Subscriber_modifyService

- Modifies subscriptions.
- Arguments

- subURI—String that contains the subscriber's URI (see "Format of the Subscriber's URI" on page 83)
- subscriptionName—String that contains the name of the subscription
- sessionName—String that contains the name of the service session; default string is *default*
- optionalAttributes—Array of one or more of the following attributes that can be modified for the subscription:
 - sessionTimeout
 - downstreamBandwidth
 - upstreamBandwidth
 - sessionTag
 - substitutions

For information about these attributes, see "Subscription Attributes" on page 85.

- Expected output—None
- SOAP fault codes—See "SOAP Fault Codes for Dynamic Service Activator" on page 86

Subscriber_login

- Logs in subscribers



NOTE: This method supports only subscribers who are identified by their IP addresses. This method does not support subscribers who are identified by the names they use to log in or by their DNSs.

- Arguments
 - subURI—String that contains the subscriber's URI (see "Format of the Subscriber's URI" on page 83)
- Expected output—Boolean operator that indicates success
- SOAP fault codes—See "SOAP Fault Codes for Dynamic Service Activator" on page 86

Subscriber_logout

- Logs out subscribers. This method supports only subscribers who are identified by their IP addresses or the names they use to log in. This method does not support subscribers who are identified by their DNSs.
- Arguments

- subURI—String that contains the subscriber's URI (see "Format of the Subscriber's URI" on page 83)
- Expected output—None
- SOAP fault codes—See "SOAP Fault Codes for Dynamic Service Activator" on page 86

Subscribers_activateService

- Activates subscribers' subscriptions to services. Manages service activations by inserting a session record in the Session State Registrar (SSR) that contains information from arguments supplied by the gateway client in addition to the subscriber's IP address, VPN ID, and session start time.
- Arguments
 - subURI—String that contains the subscriber's URI (see "Format of the Subscriber's URI" on page 83)
 - subscriptionName—String that contains the name of the subscription
 - sessionName—String that contains the name of the service session; default string is *default*
 - activationAttributes—Array of one or more of the following attributes that can be specified for the subscription:
 - sessionTimeout
 - downstreamBandwidth
 - upstreamBandwidth
 - sessionTag
 - subscriptionUsername
 - subscriptionPassword
 - substitutions

For information about these attributes, see "Subscription Attributes" on page 85.

- Expected output—None
- SOAP fault codes—See "SOAP Fault Codes for Dynamic Service Activator" on page 86

Subscribers_deactivateService

- Deactivates subscribers' subscriptions to services. Manages service deactivations by deleting the session record from the Session State Registrar (SSR).
- Arguments
 - subURI—String that contains the subscriber's URI (see "Format of the Subscriber's URI" on page 83)
 - subscriptionName—String that contains the name of the subscription

- `sessionName`—String that contains the name of the service session; default string is *default*
- Expected output—None
- SOAP fault codes—See “SOAP Fault Codes for Dynamic Service Activator” on page 86

Subscribers_modifyService

- Modifies subscriptions. Manages service modifications by updating the session record in the Session State Registrar (SSR).
- Arguments
 - `subURI`—String that contains the subscriber’s URI (see “Format of the Subscriber’s URI” on page 83)
 - `subscriptionName`—String that contains the name of the subscription
 - `sessionName`—String that contains the name of the service session; default string is *default*
 - `optionalAttributes`—Array of one or more of the following attributes that can be modified for the subscription:
 - `sessionTimeout`
 - `downstreamBandwidth`
 - `upstreamBandwidth`
 - `sessionTag`
 - `substitutions`

For information about these attributes, see “Subscription Attributes” on page 85.

- Expected output—None
- SOAP fault codes—See “SOAP Fault Codes for Dynamic Service Activator” on page 86

Subscribers_login

- Logs in subscribers



NOTE: This method supports only subscribers who are identified by their IP addresses. This method does not support subscribers who are identified by the names they use to log in or by their DNs.

- Arguments
 - `subURI`—String that contains the subscriber’s URI (see “Format of the Subscriber’s URI” on page 83)
 - `rollback`—When this argument is set to true and a failure occurs, the gateway logs out previous successful logins.

- Expected output—Boolean operator that indicates success
- SOAP fault codes—See “SOAP Fault Codes for Dynamic Service Activator” on page 86

Subscribers_logout

- Logs out subscribers. This method supports only subscribers who are identified by their IP addresses or the names they use to log in. This method does not support subscribers who are identified by their DNSs.
- Arguments
 - subURI—String that contains the subscriber's URI (see “Format of the Subscriber's URI” on page 83)
- Expected output—None
- SOAP fault codes—See “SOAP Fault Codes for Dynamic Service Activator” on page 86

Subscribers_read

- Obtains attributes of subscriber sessions for services, service sessions, and subscriber sessions; returns the result in a two-dimensional array.
- Arguments
 - subURI—String that contains the subscriber's URI (see “Format of the Subscriber's URI” on page 83)
 - subscription—Similar to a SQL select statement. Use a filter string for the first field of the select to indicate the subscriptions. Use a list of attribute names for the second field to indicate the subscription attributes.

For more information about how to specify the filter and attributes in a select argument, see the SAE CORBA Remote API documentation on the Juniper Networks Web site at

<http://www.juniper.net/techpubs/software/management/sdx/api-index.html> (the sae.Select structure).

- service—Similar to a SQL select statement. Use a filter string for the first field of the select to indicate the services. Use a list of attribute names for the second field to indicate the service attributes.

For more information about how to specify the filter and attributes in a select argument, see the SAE CORBA Remote API documentation on the Juniper Networks Web site at

<http://www.juniper.net/techpubs/software/management/sdx/api-index.html> (the sae.Select structure).

- subscriberAttrs—List of subscriber attributes.
- Expected output—Multidimensional array of Attr objects that contain the subscriptions for each identified subscriber
- SOAP fault codes—See “SOAP Fault Codes for Dynamic Service Activator” on page 86

Subscribers_readSubscriber

- Obtains attributes of subscriber sessions; returns the result in a one-dimensional array for SOAP clients that do not support two-dimensional arrays.
- Arguments
 - subURI—String that contains the subscriber's URI (see "Format of the Subscriber's URI" on page 83)
 - attrs—List of attributes.
- Expected output—One-dimensional array of AttrSeq objects that contain the subscriber session attributes as a list of attribute-value pairs for one subscriber
- SOAP fault codes—See "SOAP Fault Codes for Dynamic Service Activator" on page 86

Subscribers_readSubscription

- Determines whether a subscriber accesses services through the SRC owner's network and obtains all of that subscriber's subscriptions; returns the result in a two-dimensional array.
- Arguments
 - subURI—String that contains the subscriber's URI (see "Format of the Subscriber's URI" on page 83)
 - select—Similar to a SQL select statement. Use a filter string for the first field of the select to indicate the subscriptions. Use a list of attribute names for the second field to indicate the subscription attributes.

For more information about how to specify the filter and attributes in a select argument, see the SAE CORBA Remote API documentation on the Juniper Networks Web site at

<http://www.juniper.net/techpubs/software/management/sdx/api-index.html> (the sae.Select structure).

- Expected output—Multidimensional array of Attr objects that contain the subscriptions for the subscriber
- SOAP fault codes—See "SOAP Fault Codes for Dynamic Service Activator" on page 86

Format of the Subscriber's URI

Many of Dynamic Service Activator's methods require the argument subURI, the subscriber's Uniform Resource Identifier (URI). This argument comprises two portions: the type of subscriber (subscriber-type) and a list of the subscriber's attributes (subscriber-comp). The syntax for the subURI argument is:

```
<subscriber-type>:<subscriber-comp> [& <subscriber-comp>]*
```

The <subscriber-type> variable is the name of the Subscriber Type instance that is defined in the directory during the Dynamic Service Activator configuration. For example, the sample data provides these Subscriber Type instances for the <subscriber-type> value:

- ip—The sidType is SIT_ADDRESS
- dn—The sidType is SIT_DN
- login—The sidType is SIT_LOGIN_NAME
- assignedIp—The sidType is SIT_ADDR_IF_NAME
- intfName—The sidType is SIT_IF_NAME
- intfIndex—The sidType is SIT_IF_INDEX

The <subscriber-comp> variable has the format <type>=<value>.

- <type> can be one of the following:
 - ipAddress—Subscriber's IP address; mandatory when the sidType is SIT_ADDRESS or SIT_ADDR_IF_NAME
 - timestamp—Time at which the request was sent; optional when the sidType is SIT_ADDRESS
 - dn—Subscriber's DN; mandatory when the sidType is SIT_DN
 - intfName—Name of the interface on which the subscriber logs in; mandatory when the sidType is SIT_IF_NAME, optional when the sidType is SIT_ADDR_IF_NAME
 - vrfName—Name of the VRF associated with the subscriber; mandatory when the sidType is SIT_IF_NAME or SIT_IF_INDEX, optional when the sidType is SIT_ADDR_IF_NAME
 - login_name—Name with which the subscriber logs in; mandatory when the sidType is SIT_LOGIN_NAME
 - intfIndex—SNMP index of the interface on which the subscriber logs in; mandatory when the sidType is SIT_IF_INDEX
 - primary_user_name—Primary username; mandatory when the sidType is SIT_PRIMARY_USER_NAME
- <value> can be a combination of the following:
 - <operator>—One of the following:
; / | ? | : | @ | & | = | + | \$ | ,
 - <textString>—Text string that uses the following characters:
 - Lowercase letters
 - Uppercase letters
 - Arabic numerals
 - - | _ | . | ! | ~ | * | ' | (|)
 - % <hexNumber>—2-character hexadecimal number

For example, you can use the following subscriber's URI to look up a subscriber by IP address as specified by the sample data:

ip:ipAddress=192.168.1.10

The sample data defines a subscriber type named ip, whose sidType is SIT_ADDRESS and nicProxyNamespace is /nicProxies/ip. As a result, ip is the value of the <subscriber-type> variable. Because the sidType is SIT_ADDRESS, ipAddress is mandatory for the <type> component, and the subscriber's IP address is the <value> component of the <subscriber-comp> variable.

- Related Topics**
- API for Dynamic Service Activator on page 75
 - Configuring Dynamic Service Activator Properties (SRC CLI) on page 19
 - Methods for the Dynamic Service Activator Web Service Interface on page 76
 - Subscription Attributes on page 85
 - SOAP Fault Codes for Dynamic Service Activator on page 86

Subscription Attributes

Some methods take attributes for subscriptions, as described below.

sessionTimeout

- Timeout for the service.
- Value—Number of seconds in the range -1–2147483647
- Guideline— -1 indicates no timeout.
- Example—600

downstreamBandwidth

- Traffic rate between the subscriber and the network.
- Value—Number of bits per second in the range 0–2147483647
- Example—10000

upstreamBandwidth

- Traffic rate between the network and subscriber.
- Value—Number of bits per second in the range 0–2147483647
- Example—5000

sessionTag

- Tag that the software uses to track a session for accounting purposes.
- Value—Text string
- Example—News:Joe

subscriptionUsername

- Name of the subscriber to the service.
- Value—Text string
- Example—Joe

subscriptionPassword

- Password for the service.
- Value—Text string
- Example—Secret

substitutions

- Attributes and values that the method should substitute for existing settings.
- Value—Array of strings; each array has the format
<substutionType>=<substitutionValue>
- Example—Port=9999

SOAP Fault Codes for Dynamic Service Activator

When Dynamic Service Activator receives a SOAP request that it cannot handle, it returns a SOAP fault message to the gateway client. This message contains a text string that specifies a SOAP fault code and a text string that provides additional information about the fault.

Dynamic Service Activator returns the following SOAP fault codes. If the SOAP fault code has the format `client.<variables>`, you must correct the gateway client. However, if the SOAP fault code has the format `server.<variables>`, record the error code and notify the gateway administrator.

Client.InvalidArguments

- Specifies that one or more of the arguments for the script or method is an empty string or has a null value.
- Classification of call—Denied
- Action—Check that all arguments are correct.

Client.InvalidSubscriberFormat

- Specifies that the subURI argument, which the client passes to all methods except `invokeScript`, is invalid.
- Classification of call—Denied
- Action—Check that subURI arguments are correct.

Server.AccessDenied

- Specifies that the client does not have permission to make the request because of one or more of the following errors:
 - Web Services Gateway does not recognize the client.
 - Supplied name of the method or script is incorrect.
 - Number or value of the arguments supplied for the script or method is incorrect.
- Classification of call—Denied
- Action—Check that the values are correct. If they are, contact the gateway administrator.

Server.AccessControlMisconfiguration

- Specifies that the gateway administrator did not configure Dynamic Service Activator correctly, in one of the following ways:
 - The `saeLocatorArg` is not an integer.
 - The number of arguments is less than the value of the `saeLocatorArg` parameter.
 - The regular expressions are not correctly defined.
- Classification of call—Failed
- Action—Contact the gateway administrator.

Server.Misconfiguration

- Specifies that the gateway administrator did not configure the Web application to authenticate SOAP clients, and an unauthenticated client has passed a request to Dynamic Service Activator, which refuses unauthenticated requests.
- Classification of call—Failed
- Action—Contact the gateway administrator.

Server.SAEUnreachable

- Specifies that the gateway client either cannot identify the SAE server or cannot reach it through CORBA. This SAE can be one on which the subscriber has logged in or one on which the client wants to invoke a script.
- Classification of call—Failed
- Action—Contact the gateway administrator.

Server.SAE.UnkownSAEUser

- Specifies that the SAE on which Dynamic Service Activator tried to run a method or script cannot identify the subscriber. Dynamic Service Activator may have retained in its cache an SAE that is no longer current for the subscriber.
- Classification of call—Failed
- Action—Try this operation again. If it fails again, contact the gateway administrator.

Server.SAE.UserNotUniqueToSAE

- Specifies that the subscriber is already active on the SAE on which Dynamic Service Activator tried to run a method or script. Dynamic Service Activator may have retained in its cache an SAE that is no longer current.
- Classification of call—Failed
- Action—Try this operation again. If it fails again, contact the gateway administrator.

Server.SAE.UnknownService

- Specifies that the SAE cannot identify the service that a subscription specifies.
- Classification of call—Failed
- Action—Check the service parameter for the following methods:
 - Subscriber_activateService
 - Subscriber_deactivateService
 - Subscriber_modifyService

Server.SAE.UnknownSubscription

- Specifies that the SAE cannot identify the subscription that the subscriber wants to activate.
- Classification of call—Failed
- Action—Check the service parameter for the following methods:
 - Subscriber_activateService
 - Subscriber_deactivateService

- Subscriber_modifyService

Server.SAE.ServiceAuthenticationError

- Specifies that the service the subscriber wants to activate requires authentication.
- Classification of call—Failed
- Action—Check the activation attributes. If they are correct, contact the gateway administrator.

Server.SAE.UnknownServiceSession

- Specifies that the SAE cannot identify the service session that the subscriber wants to modify.
- Classification of call—Failed
- Action—Check the serviceId and sessionName parameters.

Server.SAE.LoginError

- Specifies that the subscriber could not log in successfully.
- Classification of call—Failed
- Action—Check the subURI, username, and userPassword arguments.

Server.SAE.Exception

- Specifies that the SAE raised an exception after Dynamic Service Activator tried to run a method or script on the SAE.
- Classification of call—Failed
- Action—Contact the gateway administrator.

Server.SAE.Overload

- Specifies that the SAE raised an overload exception after Dynamic Service Activator tried to run a method or script on the SAE.
- Classification of call—Failed
- Action—Contact the gateway administrator.

Server.SAE.ScriptProcessorError

- Specifies that the SAE's script processor failed to invoke the script. This error could occur for many reasons, such as:
 - Gateway administrator did not deploy the script on the SAE host.

- Gateway administrator did not configure the script processor correctly.
- Classification of call—Failed
- Action—Contact the gateway administrator.

Server.SAE.ScriptExecutionError

- Specifies that the script processor invoked the script, but the script was not completed successfully.
- Classification of call—Failed
- Action—Contact the gateway administrator.

CHAPTER 9

Activating Video Services with SOAP

- Overview of Web Service Interface for Video Services on page 91
- Web Service Interface Methods for On-Demand Video Services on page 92
- Web Service Interface Methods for Personalized Video Services on page 93
- Configuring Dynamic Service Activator for Video Services on page 95
- Configuring Dynamic Service Activator as an External Plug-In (SRC CLI) on page 96
- Configuring the Video Policy Manager (SRC CLI) on page 96
- Configuring the Session Database (SRC CLI) on page 99
- Monitoring the Session Database (SRC CLI) on page 100

Overview of Web Service Interface for Video Services

In certain network topologies, a Web service SOAP/XML interface between a video server and the video server policy manager allows a video server to dynamically request resources on the operator's access network. It provides operations for requesting, releasing, and querying network resources.

The Dynamic Service Activator Web service interface supports two types of video services based on the set-top box—on-demand video services and personalized video services.

The Web service interface supports the following operations for on-demand video services:

- ReleaseNetworkResources
- ReserveNetworkResource
- ClientStatus

The Web service interface supports the following operations for personalized video services:

- AllocateResource
- ReleaseResources
- GetStatus

Each of these operations is implemented as a method in the Dynamic Service Activator Web service interface.

- Related Topics**
- Configuring Dynamic Service Activator for Video Services on page 95
 - Configuring Dynamic Service Activator as an External Plug-In (SRC CLI) on page 96
 - Configuring the Video Policy Manager (SRC CLI) on page 96
 - Configuring the Session Database (SRC CLI) on page 99
 - Web Service Interface Methods for On-Demand Video Services on page 92
 - Web Service Interface Methods for Personalized Video Services on page 93

Web Service Interface Methods for On-Demand Video Services

This topic describes the methods in the Web service interface for on-demand video services.

ClientStatus

- Requests the list of sessions managed by the policy manager. For each session that is active in policy manager but not in the video server, the video server issues a `ReleaseNetworkResourceRequest` to the policy manager to release the resources allocated for the session.
- Argument—`ClientStatusRequest`
 - `ReservationClientIPV4`—IPv4 address of the video server.
- Expected output—`ClientStatusResponse` with the number of added sessions and a list of the session IDs for active sessions

ReleaseNetworkResource

- Specifies the release of resources when a subscriber stops video streaming in the `ReleaseNetworkResource` message.
- Argument—`ReleaseNetworkResourceRequest`
 - `SessionId`—Unique identifier for the session.
 - `SourceIPV4`—(Optional) IPv4 address of the video server from which the video stream originates.
 - `DestinationIPV4`—(Optional) IPv4 address of the subscriber requesting the video stream.
 - `Status`—Termination status of the video stream. OK means normal termination; FAIL means the video stream was terminated because of a problem.
- Expected output—`ReleaseNetworkResourceResponse` indicating success (0) or failure (1)

ReserveNetworkResource

- Specifies the resources that are being requested for video streaming in the ReserveNetworkResource message.
- Argument—ReserveNetworkResourceRequest
 - SessionId—Unique identifier for the session.
 - BitRate—Provides information about the required bandwidth for the session.
 - ReservationClientIPv4—IPv4 address of the video server.
 - PolicyMgrClientURL—Video server's URL that the policy manager uses to initiate requests to the video server.
 - SourceIPv4—IPv4 address of the video server from which the video stream originates.
 - DestinationIPv4—IPv4 address of the subscriber requesting the video stream.
 - PolicyInformation—Provides policy information that the policy manager uses to process the session.
 - Reserved for future use.
- Expected output—ReserveNetworkResourceResponse indicating success (0) or failure (1)

Web Service Interface Methods for Personalized Video Services

This topic describes the methods in the Web service interface for personalized video services.

AllocateResource

- Specifies the resources that are being requested for video streaming.
- Argument—AllocateResourceRequest
 - MessageVersion—Message version to check compatibility.
 - ApplicationName—Name of the application generating the request. The policy manager can use this attribute to select different policies for different applications.
 - SessionId—Unique identifier for the session.
 - BitRate—Provides downstream bandwidth for the session.
 - SignalingServer—Address of signaling server that identifies the source of the request.
 - MediaSourceIP—IPv4 address of the interface from which the video stream originates.
 - DestinationIP—IPv4 address of the subscriber requesting the video stream.
 - MACAddress—(Optional) MAC address of the originating device.
 - MediaType—(Optional) Media type: VIDEO, DATA, VOICE, or APPLICATION.

- **ContentType**—(Optional) Content type: SD or HD.
- **Priority**—(Optional) Media streaming priority: FREE, PAY, or AD.
- **Duration**—(Optional) Expected duration of media stream.
- **TrickFileDurationDelay**—(Optional) Maximum time that the media stream can be paused.
- **Expected output**—AllocateResourceResponse indicating success (200) or failure (any other value)

ReleaseResource

- Specifies the release of resources when a video session is terminated or when the policy manager detects a problem with service delivery.
- **Argument**—ReleaseResourceRequest
 - **MessageVersion**—Message version to check compatibility.
 - **ApplicationName**—Name of the application generating the request.
 - **SessionId**—Unique identifier for the session.
 - **Status**—Code and text indicating reason for session termination.
- **Expected output**—ReleaseNetworkResourceResponse indicating success (200) or failure (any other value)

GetStatus

- Requests the status or synchronizes sessions between the policy manager and the video server.
- **Argument**—StatusRequest
 - **MessageVersion**—Message version to check compatibility.
 - **ApplicationName**—Name of the application generating the request.
 - **ApplicationServer**—Address of the server from which the request originates.
 - **StatusType**—Type of status request; can be SessionList, SessionID, Version, or Keepalive.
The type of status request results in different responses.
 - **MediaType**—Media type for status request; used only when StatusType is SessionList.
 - **SessionId**—Unique identifier for the session.
 - **Status**—Code and text indicating reason for session termination.
- **Expected output**—StatusResponse depends on the StatusType with a list of the session IDs for active sessions.

- **SessionList—Response** indicates the number of added sessions and contains a list of the session IDs for active sessions. If **MediaType** is included in the request, the response includes only sessions of the specified media type.
- **SessionID—Response** contains information about a single session, including **BitRate**, **SourceIP**, and **DestinationIP**.
- **Version—Response** contains information about the software version of the video server or the policy manager, including **MessageVersion**, **SoftwareVersion** (software version of the server), or **InterfaceVersion** (version of the SOAP interface for the server being queried).
- **KeepAlive—Response** indicates availability of the server being queried.

Configuring Dynamic Service Activator for Video Services

To configure Dynamic Service Activator to support video services, you configure Dynamic Service Activator with access to these methods. See “Configuring Dynamic Service Activator Properties (SRC CLI)” on page 19.

To support on-demand video services, configure access to these methods:

- **query-client-status**—Searches for active sessions of a given video server managed by the SRC software.
- **release-network-resource**—Specifies the resources that are being requested to be released in the **ReleaseNetworkResources** message.
- **reserve-network-resource**—Specifies the resources that are being requested in the **ReserveNetworkResource** message.

To support personalized video services, configure access to these methods:

- **allocate-resource**—Specifies the resources that are being requested in the **AllocateResource** message.
- **query-status**—Searches for active sessions of a given video server managed by the SRC software.
- **release-resource**—Specifies the resources that are being requested to be released in the **ReleaseResource** message.

In addition to the general Dynamic Service Activator configuration tasks, perform these tasks:

- Configuring Dynamic Service Activator as an External Plug-In (SRC CLI) on page 96
- Configuring the Video Policy Manager (SRC CLI) on page 96
- Configuring the Session Database (SRC CLI) on page 99
- Monitoring the Session Database (SRC CLI) on page 100

Related Topics • Enabling Dynamic Service Activator on a Web Application Server (SRC CLI) on page 15

Configuring Dynamic Service Activator as an External Plug-In (SRC CLI)

To support video services, you must configure the SAE to recognize Dynamic Service Activator by configuring an external plug-in for the SAE.

To configure an external plug-in for the SAE:

1. From configuration mode, access the statement that configures the external plug-ins.

```
user@host# edit shared sae configuration plug-ins name name external
```

2. Configure the object reference of the external plug-in that is exported to the SAE.

```
[edit shared sae configuration plug-ins name name external]  
user@host# set corba-object-reference corba-object-reference
```

where *corba-object-reference* is the corbaloc URL in the format `corbaloc::host:9802/hostname/dsaPOA/ASGVIDEO`, where:

- *host* is the IP address of the C Series Controller
- *hostname* is the hostname of the C Series Controller

3. Specify the service-session-name plug-in attribute.

```
[edit shared sae configuration plug-ins name name external]  
user@host# set attributes service-session-name
```

Configuring the Video Policy Manager (SRC CLI)

To support video services, configure the video policy manager to specify the video server and to map information in a request to an SRC service. To configure the video policy manager, perform the following tasks:

- Configuring the Video Server on page 96
- Configuring the Video Service Mappings (SRC CLI) on page 97

Configuring the Video Server

You can manage video servers that provide on-demand or personalized video services.

To configure the video server that is making SOAP requests to provide video services:

1. From configuration mode, access the statement that configures the video server, and specify the IP address as the name of the video server.

```
user@host# edit shared dsa configuration video-policy-manager video-server name
```

2. Specify the type of video server.

```
[edit shared dsa configuration video-policy-manager video-server name]  
user@host# set type (seachange | tandberg)
```

where **seachange** provides on-demand video services and **tandberg** provides personalized video services.

3. Specify the URL of the video server used for synchronization requests.

```
[edit shared dsa configuration video-policy-manager video-server name]
user@host# set url url
```

4. Specify the subscriber type requesting resources, which can be assigned IP (**address-subscriber**) or interface name (**static-interface-subscriber**).

```
[edit shared dsa configuration video-policy-manager video-server name]
user@host# set subscriber-type (address-subscriber | static-interface-subscriber)
```

5. (Optional) Specify the software version of the video server; used only when type is **tandberg**.

```
[edit shared dsa configuration video-policy-manager video-server name]
user@host# set software-version software-version
```

6. (Optional) Specify the interface version of the video server; used only when type is **tandberg**.

```
[edit shared dsa configuration video-policy-manager video-server name]
user@host# set interface-version interface-version
```

7. (Optional) Specify the interval for synchronization requests.

```
[edit shared dsa configuration video-policy-manager video-server name]
user@host# set interval interval
```

8. (Optional) Specify the client ID to access the policy manager; used for basic HTTP authentication.

```
[edit shared dsa configuration video-policy-manager video-server name]
user@host# set client-id client-id
```

9. (Optional) Specify the client password to access the policy manager; used for basic HTTP authentication.

```
[edit shared dsa configuration video-policy-manager video-server name]
user@host# set client-password client-password
```

Configuring the Video Service Mappings (SRC CLI)

You can configure how SOAP requests are mapped to SRC service activation requests for on-demand or personalized video services.

To configure the mapping for on-demand video services:

1. From configuration mode, access the statement that configures the service mapping for SOAP requests.

```
user@host# edit shared dsa configuration video-policy-manager service-map
```

2. (Optional) Specify the default service name to use if no mapping is defined.

```
[edit shared dsa configuration video-policy-manager service-map]
user@host# set default-service default-service
```

SOAP requests for personalized video services contain an application name and can include media type and content type attributes, which can be used to define the service.

To configure the mapping for personalized video services:

1. From configuration mode, access the statement that configures the service mapping.
user@host# edit shared dsa configuration video-policy-manager service-map
2. (Optional) Specify the default service name to use if a mapping is not defined.
[edit shared dsa configuration video-policy-manager service-map]
user@host# set default-service *default-service*
3. (Optional) Configure the service mapping for a specific application in the SOAP request to the SRC service name.
[edit shared dsa configuration video-policy-manager service-map]
user@host# edit application *name*
4. (Optional) Specify the default service name that maps to the application. If the media type attribute is not provided in the request, this service name is used.
[edit shared dsa configuration video-policy-manager service-map application *name*]
user@host# set default-service *default-service*
5. (Optional) Specify that the value in the SOAP request be used for the service name.
[edit shared dsa configuration video-policy-manager service-map application *name*]
user@host# set use-request-value
6. (Optional) Configure the service mapping for a specific media type in the SOAP request to the SRC service name.
[edit shared dsa configuration video-policy-manager service-map application *name*]
user@host# edit media-type (application | data | video | voice)

For example:

- ```
[edit shared dsa configuration video-policy-manager service-map application name]
user@host# edit media-type video
```
7. (Optional) Specify the part of the service name that maps to the media type. If the content type attribute is not provided in the request, this service name is used.  
**[edit shared dsa configuration video-policy-manager service-map application *name***  
**media-type video]**  
**user@host# set default-service *default-service***
  8. (Optional) Specify that the value in the SOAP request be used for the service name.  
**[edit shared dsa configuration video-policy-manager service-map application *name***  
**media-type video]**  
**user@host# set use-request-value**
  9. (Optional) Configure the service mapping for a specific content type in the SOAP request to the SRC service name.  
**[edit shared dsa configuration video-policy-manager service-map application *name***  
**media-type video]**  
**user@host# edit content-type (hd | sd)**

For example:

```
[edit shared dsa configuration video-policy-manager service-map application name]
user@host# edit content-type hd
```

10. (Optional) Specify the part of the service name that maps to the content type.

```
[edit shared dsa configuration video-policy-manager service-map application name
media-type video content-type hd]
user@host# set default-service default-service
```

11. (Optional) Specify that the value in the SOAP request be used for the service name.

```
[edit shared dsa configuration video-policy-manager service-map application name
media-type video content-type hd]
user@host# set use-request-value
```

## Configuring the Session Database (SRC CLI)

To support video services, Dynamic Service Activator uses a session database to maintain information about active video sessions. Each session record includes such information as session ID, SRC service name, and subscriber.

To configure the session database, perform the following tasks:

- Configuring the Session Database Cluster Server (SRC CLI) on page 99
- Configuring the Session Database Topology (SRC CLI) on page 99
- Deleting Session Database Configurations on page 100

### Configuring the Session Database Cluster Server (SRC CLI)

To configure the server for the session database cluster:

1. From configuration mode, access the statement that configures the session database cluster. Only one cluster (primary) is supported.

```
user@host# edit shared session-database cluster primary servers
```

2. Specify the hostname of the session database cluster server.

```
[edit shared session-database cluster primary servers]
user@host# set server hostname
```

3. Specify the IP address of the server.

```
[edit shared session-database cluster primary servers]
user@host# set address address
```

### Configuring the Session Database Topology (SRC CLI)

To configure the topology for the session database cluster:

1. From configuration mode, access the statement that configures the session database cluster. Only one cluster (primary) is supported.

```
user@host# edit shared session-database cluster primary servers
```

2. (Optional) Specify the maximum amount of memory available to the session database. Do not change this value unless instructed to do so by Juniper Networks.

**[edit shared session-database cluster primary servers]**  
**user@host# set java-heap-size *java-heap-size***

3. (Optional) Specify the garbage collection options.

**[edit shared session-database cluster primary servers]**  
**user@host# set java-garbage-collection-options *java-garbage-collection-options***

## Deleting Session Database Configurations

When you create a session database configuration, the Web application server becomes a client of the session database so that they can share service session information. If you delete the session database configuration, you must disable access to service session information in the session database from the Web application server.

To disable access from the Web application server:

- From configuration mode, delete the statement that enables the Web application server to share service session information with the session database.

**[edit]**  
**user@host# delete application-server share-service-session-store**

**Related Topics** • Monitoring the Session Database (SRC CLI) on page 100

---

## Monitoring the Session Database (SRC CLI)

- Viewing the Session Database Status (SRC CLI) on page 100
- Viewing Active Service Sessions (SRC CLI) on page 100

### Viewing the Session Database Status (SRC CLI)

**Purpose** View the session database cluster status.

**Action** user@host> **show session-database status**

### Viewing Active Service Sessions (SRC CLI)

**Purpose** View the number of active service sessions. This command displays the number of sessions that were added since cluster startup and includes any sessions that have been released.

**Action** user@host> **show session-database service-session-count**

**Related Topics** • Configuring the Session Database (SRC CLI) on page 99



## CHAPTER 10

# Activating PCMM Services with SOAP

- Overview of Web Service Interface for PCMM on page 101
- SRC PCMM Web Service Interface Methods on page 101
- Configuring PCMM Policies and Parameter Substitutions (SRC CLI) on page 104
- Configuring Services That Are Available for PCMM Clients (SRC CLI) on page 106

## Overview of Web Service Interface for PCMM

---

PCMM Web Service Interface Specification (PKT-SP-MM-WS-101-051221) defines a common Web service SOAP/XML interface between a generic application server and a PCMM application manager. The interface allows an application server to dynamically request resources on the cable operator's access network. It provides operations for requesting, releasing, and querying network resources on the cable network. The SRC PCMM Web service interface supports the following operations, which are specified in the PCMM Web service interface:

- CommitResources
- ReleaseResources
- QueryAvailableServices
- QueryContexts

Each of these operations is implemented as a method in the SRC PCMM Web service interface.

- Related Topics**
- Configuring Services That Are Available for PCMM Clients (SRC CLI) on page 106
  - Configuring PCMM Policies and Parameter Substitutions (SRC CLI) on page 104
  - SRC PCMM Web Service Interface Methods on page 101

## SRC PCMM Web Service Interface Methods

---

This section describes the methods in the SRC PCMM Web service interface.

### *CommitResources*

- Specifies the resources that are being requested in the CommitResource message.
- Argument—CommitResourcesReq
  - classifier—Object that identifies the traffic flow that the application server is requesting services for. The object contains:

- protocol
- sourceIpAddress
- sourceIpMask
- sourcePortStart
- sourcePortEnd
- destinationIpAddress
- destinationIpMask
- destinationPortStart
- destinationPortEnd

For more information, see “Configuring Classify-Traffic Conditions for Dynamic Service Activator” on page 104.

- trafficProfile—Provides information about the bandwidth and QoS characteristics desired for a request. You express the traffic profile through SRC policies. You can configure the policies one of the following ways:
  - FlowSpec action—See “Configuring FlowSpec Actions for Dynamic Service Activator” on page 105.
  - Specifying a traffic class in the service class name action—See “Configuring Service Class Name Actions for Dynamic Service Activator” on page 105.
  - Bandwidth parameter—See “Configuring DOCSIS Actions for Dynamic Service Activator” on page 105.
- TimeUsageLimit—Specifies, in seconds, a limit on the lifetime of a context. An application server may specify multiple TimeUsageLimit elements to request different limits in the upstream and downstream directions. If the application server does not specify a TimeUsageLimit, the application manager determines the TimeUsageLimit.
- SubscriberID—IPv4 address or SubscriberURI
  - If a subscriber URI and an IPv4 address are provided, the IPv4 address is ignored.
  - If the IPv4 address is supplied and the subscriber URI is not supplied, then a subscriber URI is constructed with the given IPv4 address as an assignedIp subscriber type (sidType is SIT\_ADDR\_IF\_NAME).

- **ServiceName**—Name of SRC service
- **contextID**—Globally unique identifier that the application manager must use if it is included in the message

The context ID is used as the SRC **sessionName** when the PCMM gateway activates a service.

- **Expected output**—CommitResourcesRsp. The CommitResourcesRsp object contains the ContextID of the activated service.

### ***ReleaseResources***

- Specifies the resources that are being requested to be released in the ReleaseResources message.
- **Argument**—ReleaseResourcesReq
  - **Subscriber URI**—IP address of the subscriber
  - **ServiceName**—Name of SRC service
  - **contextID**—Globally unique identifier that the application manager must use if it is included in the message

The context ID is used as the SRC **sessionName** when the PCMM gateway activates a service.

- **Expected output**—ReleaseResourcesRsp

### ***QueryAvailableServices***

- Searches for the services that are available to the calling application.
- **Expected output**—QueryAvailableServicesRsp

### ***QueryContexts***

- Searches for the context ID and context status for a subscriber.
- **Argument**—QueryContextsReq
  - **SubscriberID**—IPv4 address or SubscriberURI
    - If a subscriber URI and an IPv4 address is provided, the IPv4 address is ignored.
    - If the IPv4 address is supplied and the subscriber URI is not supplied, then a subscriber URI is constructed with the given IPv4 address as an assignedIp subscriber type (sidType is SIT\_ADDR\_IF\_NAME).
  - **ServiceName**—Name of SRC service
  - **contextID**—Globally unique identifier that the application manager must use if it is included in the message

The context ID is used as the SRC sessionName when the PCMM gateway activates a service.

- Expected output—QueryContextsRsp

## Configuring PCMM Policies and Parameter Substitutions (SRC CLI)

This topic describes how to define your SRC policies and parameter substitutions so that they comply with the PCMM Web service interface. The sample data shows the types of policies and parameters that you can configure. View the sample data with the following command:

```
user@host> show configuration policies folder sample folder pcmm folder pcmm-ws
```

If you use parameter substitutions, the PCMM Web service interface provides the values for the substitutions. You must name your parameters as specified in this topic.

- Configuring Classify-Traffic Conditions for Dynamic Service Activator on page 104
- Configuring FlowSpec Actions for Dynamic Service Activator on page 105
- Configuring Service Class Name Actions for Dynamic Service Activator on page 105
- Configuring DOCSIS Actions for Dynamic Service Activator on page 105

### Configuring Classify-Traffic Conditions for Dynamic Service Activator

Table 6 on page 104 lists the classify-traffic condition fields that you can configure for PCMM classifiers that will be used with PCMM Web service interface. It also provides the parameter names that you must use if you configure parameter substitutions for the classifier.

**Table 6: Parameter Names for Classify-Traffic Conditions**

| Field                  | Parameter Name                                                                   |
|------------------------|----------------------------------------------------------------------------------|
| Protocol               | protocol                                                                         |
| Source IP Address      | sourceIpAddress                                                                  |
| Source IP Mask         | sourceIpMask                                                                     |
| Port                   | sourcePortStart and sourcePortEnd<br>destinationPortStart and destinationPortEnd |
| Destination IP Address | destinationIpAddress                                                             |
| Destination IP Mask    | destinationIpMask                                                                |

## Configuring FlowSpec Actions for Dynamic Service Activator

You can use a FlowSpec action to specify the traffic profile in CommitResource messages. Table 7 on page 105 lists the fields that you can use for PCMM policies that will be used with PCMM Web service interface. It also provides the parameter names that you must use if you configure parameter substitutions for the FlowSpec action.

**Table 7: Parameter Names for FlowSpec Actions**

| Field                | Parameter Name  |
|----------------------|-----------------|
| Service Number       | serviceNumber   |
| Token Bucket Rate    | bucketRate      |
| Token Bucket Size    | bucketDepth     |
| Peak Data Rate       | peakRate        |
| Minimum Policed Unit | minPolicedUnit  |
| Maximum Packet Size  | maxDatagramSize |
| Rate                 | reservedRate    |
| Slack Term           | slackTerm       |

## Configuring Service Class Name Actions for Dynamic Service Activator

You can use a service class name action to specify the traffic profile in CommitResource messages. Table 8 on page 105 lists the field that you can use for PCMM policies that will be used with PCMM Web service interface. It also provides the parameter names that you must use if you configure parameter substitutions for a service class name action.

**Table 8: Parameter Names for Service Class Actions**

| Field         | Parameter Name |
|---------------|----------------|
| Service Class | TrafficClass   |

## Configuring DOCSIS Actions for Dynamic Service Activator

You can use a DOCSIS action to specify the priority and bandwidth in the traffic profile in CommitResource messages. Table 9 on page 106 lists the fields that you can use for PCMM policies that will be used with PCMM Web service interface. It also provides the parameter names that you must use if you configure parameter substitutions for the DOCSIS action.

Table 9: Parameter Names for DOCSIS Actions

| Field                          | Parameter Name |
|--------------------------------|----------------|
| Traffic Priority               | priority       |
| Maximum Sustained Traffic Rate | bandwidth      |
| Maximum Traffic Burst          | bandwidth      |
| Minimum Reserved Traffic Rate  | bandwidth      |

- Related Topics**
- Overview of Web Service Interface for PCMM on page 101
  - Configuring Services That Are Available for PCMM Clients (SRC CLI) on page 106
  - Configuring Classify-Traffic Conditions (SRC CLI)
  - Configuring FlowSpec Actions (SRC CLI)
  - Configuring Service Class Name Actions (SRC CLI)
  - Configuring DOCSIS Actions (SRC CLI)

## Configuring Services That Are Available for PCMM Clients (SRC CLI)

Application managers require that usernames be specified from the Web Services Security: Username Token Profile 1.0, OASIS Standard.

Use the following configuration statements to configure available services for PCMM clients:

```
shared dsa group name configuration client name
shared dsa group name configuration client name permissions {
 pcmm-service [pcmm-service...];
}
```

To configure PCMM clients and the services that are available to the clients:

1. From configuration mode, access the statement that configures the PCMM client. You must use the same name for the PCMM client that is configured on the Web application server.

If you disable the access control mechanism and you configure the Web application server to authenticate clients with any username and password, Dynamic Service Activator sends the text string “anonymous client” as the first argument to the SAE’s Java scripts interface module.

```
[edit]
user@host# edit shared dsa group name configuration client name permissions
```

2. Specify the services available to the PCMM client.

```
[edit shared dsa group name configuration client name permissions]
user@host# set pcmm-service [pcmm-service...]
```

- Related Topics**
- Overview of Web Service Interface for PCMM on page 101
  - Sample Data for Dynamic Service Activator on page 34
  - Configuring PCMM Policies and Parameter Substitutions (SRC CLI) on page 104
  - Configuring User Accounts for Web Applications (SRC CLI)
  - SRC PCMM Web Service Interface Methods on page 101





## PART 2

# Providing Services in IMS Networks

- Providing Services in IMS Networks on page 111
- Providing Services in IMS Networks (SRC CLI) on page 121
- Testing IMS Service Sessions (SRC CLI) on page 145



## CHAPTER 11

# Providing Services in IMS Networks

- Overview of an IMS Environment on page 111
- IMS and ETSI References on page 112
- IMS Layers on page 114
- ETSI-TISPAN Architecture on page 115
- SRC Software in the ETSI-TISPAN Architecture on page 117
- SRC Software in the IMS Environment on page 117

### Overview of an IMS Environment

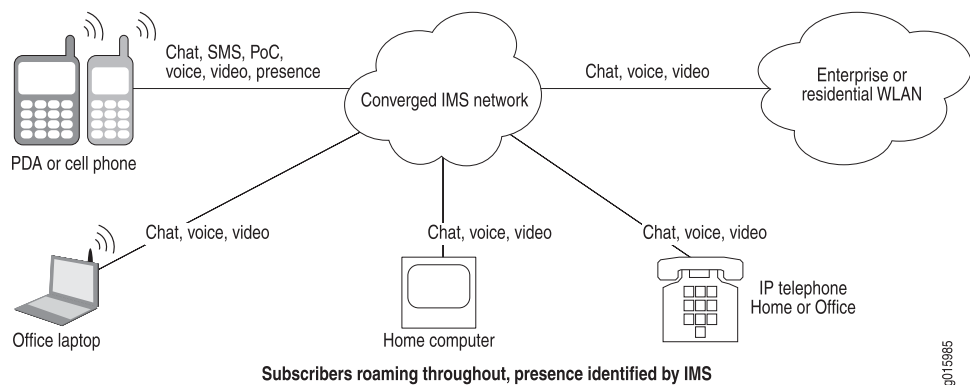
---

IP multimedia subsystem (IMS) is a flexible network architecture that allows providers to introduce rich multimedia services across both next-generation packet-switched and traditional circuit-switched networks. It uses open interfaces and functional components that can be assembled flexibly to support real-time interactive services and applications.

Third Generation Partnership Project (3GPP) developed IMS to provide a standards-based architecture for mobile carriers to migrate to their next-generation networks that will support applications that combine voice, video, and data functionality. The European Telecommunications Standards Institute (ETSI) created Telecommunications and Internet Converged Services and Protocols for Advanced Networks (TISPAN) to extend IMS support to fixed-line carriers. This extension is commonly called fixed mobile convergence (FMC). IMS/FMC allows subscribers to access any network (wireless or fixed) from any device (computer, PDA, or cell phone) and be able to move seamlessly from one network to another.

Figure 3 on page 112 shows, at a high level, a converged IMS network that manages and controls the movement of subscribers between fixed and wireless networks.

Figure 3: A Simplified IMS Converged Network (Service Focus)



By itself, IMS does not specify new services; rather, it provides a framework for network operators to build and launch their services regardless of access method. The IMS architecture simplifies network operations and allows providers to focus on service introduction and business opportunities. For example, an IMS architecture could allow fixed and mobile users to communicate using voice, video, chat, and online gaming, and to take advantage of functionality such as Push-to-Talk over Cellular (PoC; the ability to quickly arrange meetings through a walkie-talkie mechanism), instant messaging, and presence (whether and how a subscriber is available, and how the subscriber wants to be contacted).

- Related Topics**
- IMS and ETSI References on page 112
  - IMS Layers on page 114
  - ETSI-TISPAN Architecture on page 115
  - SRC Software in the ETSI-TISPAN Architecture on page 117
  - SRC Software in the IMS Environment on page 117
  - Configuring the IMS Software (SRC CLI) on page 123

## IMS and ETSI References

For more information about IMS and TISPAN, consult the following specifications:

- ETSI ES 283 026 V0.0.7 (2005-10) *Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Resource and Admission Control; Protocol for QoS reservation information exchange between the Service Policy Decision Function (SPDF) and the Access-Resource and Admission Control Function (A-RACF) in the Resource and Protocol specification.*
- ETSI TS 183 017 V.0.0.8 (2005-10) *Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Resource and Admission Control: DIAMETER protocol for session based policy set-up information exchange between the Application Function (AF) and the Service Policy Decision Function (SPDF); Protocol specification.*

- ETSI ES 283 034 V0.0.5 (2005-10) *Telecommunications and Internet converged Services and Protocols for Advanced Networks (TISPAN); Network Attachment Sub-System (NASS); e4 interface based on the DIAMETER protocol.*

## Abbreviations

Table 10 on page 113 identifies abbreviations used in the IMS and ETSI-TISPAN environments.

**Table 10: Abbreviations in the IMS and ETSI-TISPAN Environments**

| Abbreviation | Description                                                                                                                               |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| 3GPP         | 3rd Generation Partnership Project, which developed the IMS specifications.                                                               |
| A-RACF       | Access-resource and admission control function. Provides admission control and network policy assembly.                                   |
| AVP          | Attribute value pair                                                                                                                      |
| BGF          | Border gateway function                                                                                                                   |
| ETSI         | European Telecommunications Standards Institute                                                                                           |
| FMC          | Fixed mobile convergence                                                                                                                  |
| IMS          | IP multimedia subsystem                                                                                                                   |
| NGN          | Next-generation network                                                                                                                   |
| RACS         | Resource and admission control subsystem. Consists of the A-RACF and the SPDF.                                                            |
| RCEF         | Resource control enforcement function                                                                                                     |
| SPDF         | Service policy decision function. The SPDF coordinates the resource reservations requests that it receives from the application function. |
| TISPAN       | Telecommunications and Internet Converged Services and Protocols for Advanced Networks                                                    |

### Related Topics

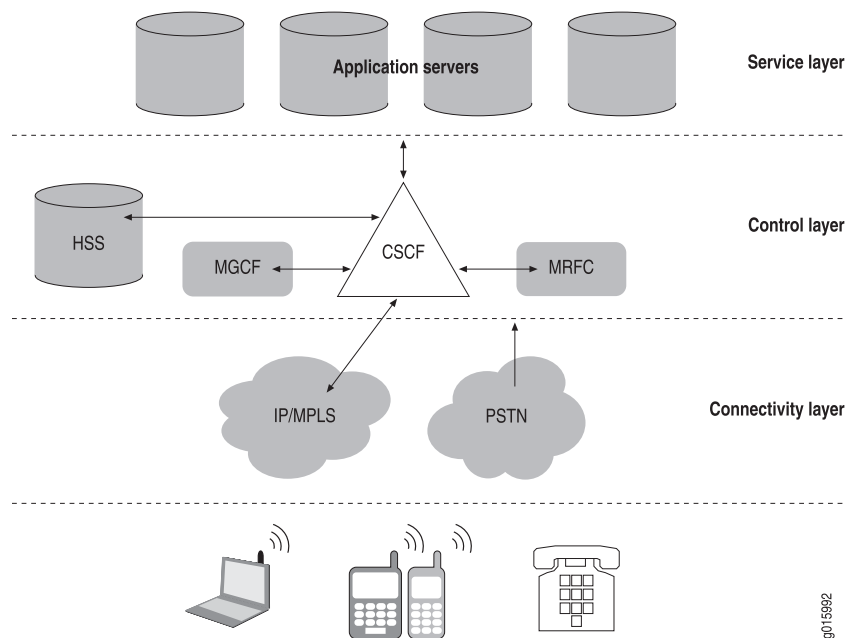
- Overview of an IMS Environment on page 111
- IMS Layers on page 114
- ETSI-TISPAN Architecture on page 115
- SRC Software in the ETSI-TISPAN Architecture on page 117
- Configuring the IMS Software (SRC CLI) on page 123

## IMS Layers

The IMS specifications define functions to handle the signaling and subscriber traffic for multimedia applications. The functions are separated into logical layers, and many of the specified functions often reside in a single platform. Vendors have the flexibility to implement IMS functions in consolidated ways, and it is natural that platforms such as softswitches will combine many logically separate IMS call-processing functions, and that routers will take on some of the session-enforcement and gateway functionality in IMS.

The three layers are the service layer, the control layer, and the connectivity layer. Figure 4 on page 114 shows a high-level view of the IMS architecture.

**Figure 4: High-Level View of the IMS Architecture**



- **Service layer**—Hosts application and content services, including application servers and Web servers. It also includes generic service enablers that manage service elements such as user groups and presence. These service elements connect to subscribers through the control plane. The application layer supports most of the multimedia applications or application enablers, such as presence and location of the subscriber.
- **Control layer**—Makes the policy decisions that are enforced in the connectivity layer. This layer provides session control and management, and is responsible for setting up and taking down packet sessions. It also contains information about subscriber authentication, service authorization, and location.
- **Connectivity layer**—Supports the core network architecture of the General Packet Radio Service (GPRS), which consists of support nodes for data services. This layer is where routers, switches, firewalls, and optical transport reside, along with gateways that translate protocols between packet- and circuit-based traffic.

## Signaling Protocol

Session Initiation Protocol (SIP) is the main signaling protocol in IMS. SIP is the proposed standard for multimedia communication between subscribers interacting with voice, video, and instant messaging. In IMS, the use of SIP facilitates interconnectivity between fixed and mobile networks.

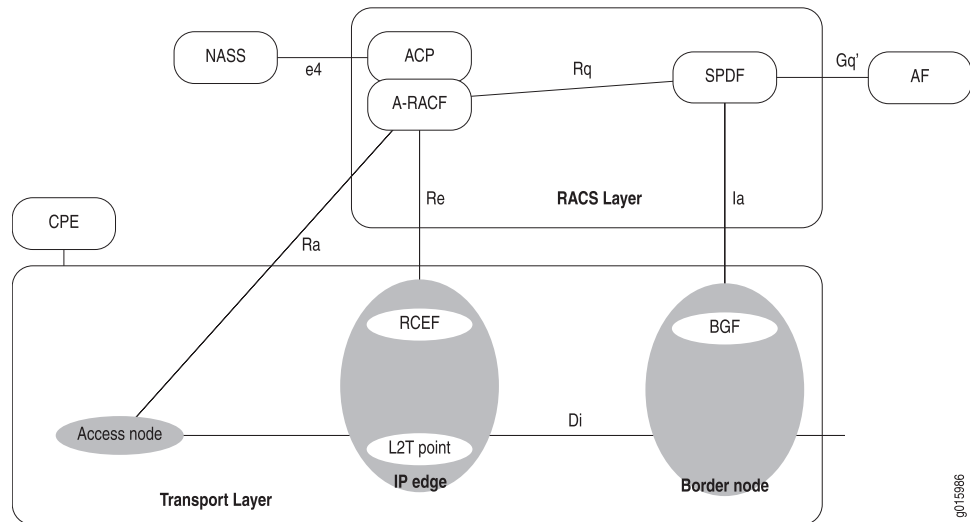
- Related Topics**
- Overview of an IMS Environment on page 111
  - IMS and ETSI References on page 112
  - SRC Software in the IMS Environment on page 117
  - Configuring the IMS Software (SRC CLI) on page 123

## ETSI-TISPA Architecture

TISPA is an extension to the IMS architecture developed by ETSI to fit the specific requirements of fixed-line providers.

Figure 5 on page 115 shows a high-level view of the TISPA architecture.

**Figure 5: High-Level View of the ETSI-TISPA Architecture**



## RACS Layer

The RACS layer is the TISPA next-generation network subsystem that is responsible for elements of policing control, including resource reservation and admission control in the access and aggregation networks. The RACS layer also includes support for NAT in the access, aggregation, and core networks required to support end-to-end application-initiated sessions.

The RACS provides policy-based transport control services to applications. These services enable applications to request and reserve transport resources from the transport networks within the scope of the RACS.

## Rq Interface

The Rq interface is the interface between the SPDF and the A-RACF. The SPDF issues requests for resources in the access network through the Rq interface. These requests indicate IP QoS characteristics. The A-RACF uses the IP QoS information to perform admission control and indicates to the SPDF through the Rq interface its admission control decisions.

## SPDF

The SPDF is a functional element that coordinates the resource reservation requests that it receives from the application function (the application-level controller, such as a SIP server). The SPDF performs the following functions:

- Determines whether the request information received from the application function is consistent with the policy rules defined in the SPDF.
- Authorizes the requested resources for the application function session. The SPDF uses the request information received from the application function to calculate the proper authorization (that is, to authorize certain media components).
- Provides the location of the BGF and/or the A-RACF device, in accordance with the required transport capabilities.
- Requests resources of the A-RACF.
- Requests services from the BGF.
- Hides the details of the RACS and the core transport layer from the control architecture.
- Provides resource mediation by mapping requests from application functions toward an appropriate A-RACF and/or BGF.

## A-RACF

The A-RACF is a functional element that provides admission control and network policy assembly.

For admission control, the A-RACF receives requests for QoS resources from the SPDF and uses the QoS information received to perform admission control. It then indicates to the SPDF whether or not a request for resources is granted.

Access network policies are a set of rules that specify the policies that should be applied to an access line. For network policy assembly, the A-RACF:

- Ensures that requests from the SPDF match the access policies because multiple SPDFs can request resources from the A-RACF.
- Combines the requests from the SPDFs that have requested resources and ensures that the total of the requests match the capabilities of the access line.

- Related Topics**
- Overview of an IMS Environment on page 111
  - IMS and ETSI References on page 112

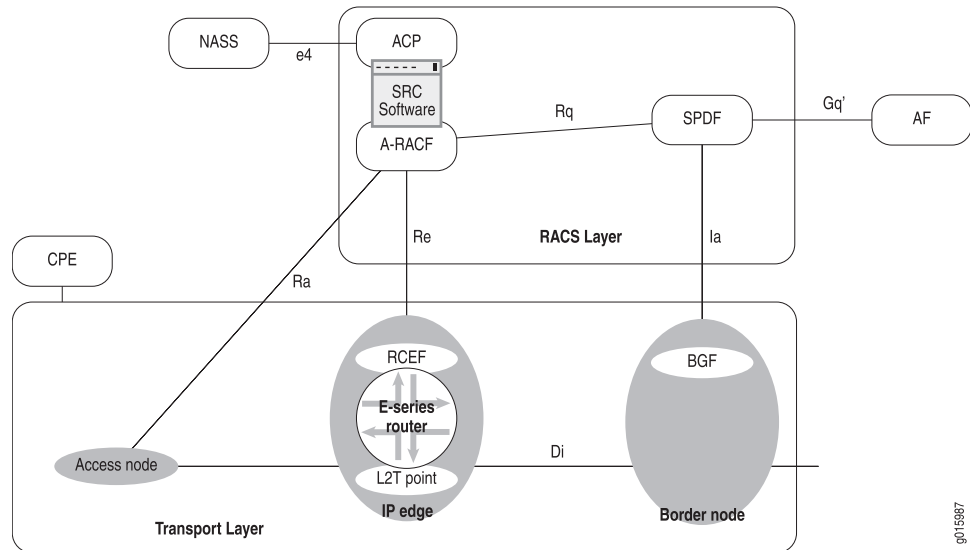


- SRC Software in the ETSI-TISPAN Architecture on page 117

## SRC Software in the ETSI-TISPAN Architecture

Figure 6 on page 117 shows the SRC software in the ETSI-TISPAN architecture.

**Figure 6: SRC Software in the ETSI-TISPAN Architecture**



The SAE provides the A-RACF functionality, and the SRC software provides a northbound Rq interface from the A-RACF to the SPDF. This interface is equivalent to the Rq interface defined in the ETSI-TISPAN release 1 architecture. It is a DIAMETER protocol-based interface that allows the SRC software to integrate with services found on the application layer of IMS.

The SRC software uses its COPS and BEEP interfaces as the Re interface to Juniper Networks routers.

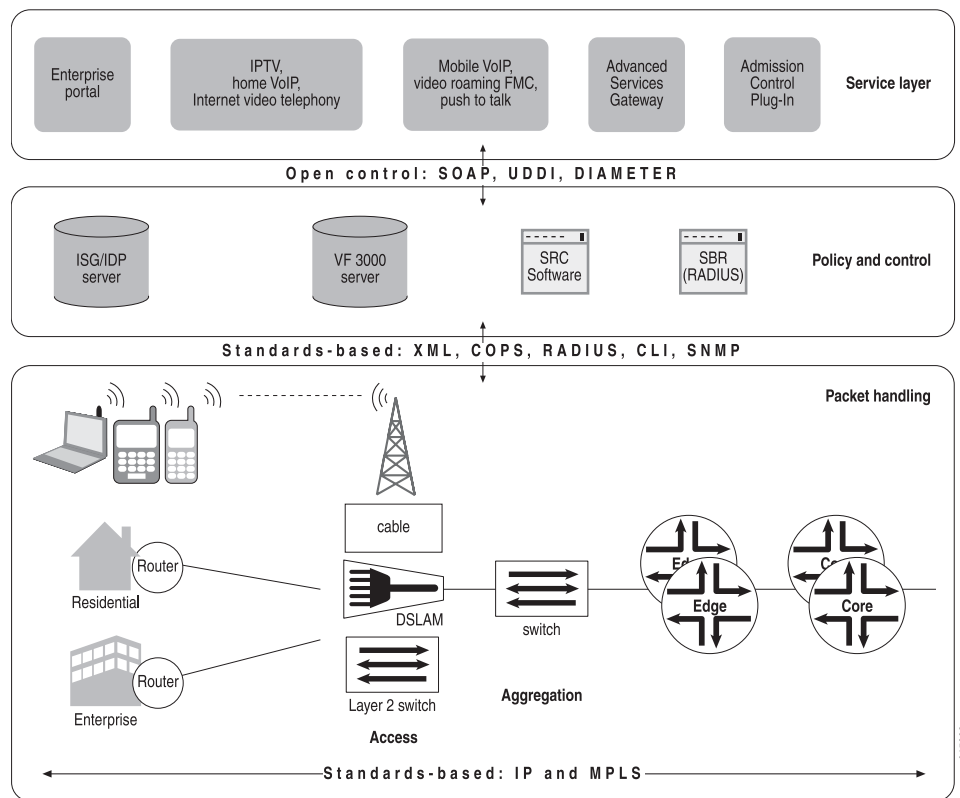
- Related Topics**
- Overview of an IMS Environment on page 111
  - IMS and ETSI References on page 112
  - ETSI-TISPAN Architecture on page 115
  - Configuring the IMS Software (SRC CLI) on page 123

## SRC Software in the IMS Environment

Figure 7 on page 118 shows the Juniper Networks layered IMS architecture.

The northbound Rq interface of the policy and control layer allows integration with SRC applications, such as the portals, the Application Services Gateway, and the Admission Control Plug-In (ACP).

Figure 7: Juniper Networks IMS Architecture



## State Synchronization

When the SRC IMS Gateway fails over or is restarted, it needs to use the session ID that is acquired during service activation to deactivate the service. This information can be passed to the SAE if the IMS Gateway synchronizes states with the SAE. If state synchronization is enabled, the current session information can be transferred so that the IMS Gateway does not have to keep a local and persistent copy of the data. The IMS Gateway registers its interoperable object reference (IOR) with the SAE so that the SAE can communicate with the IMS Gateway.

## Redundancy

If two SRC IMS Gateways synchronize states with the SAE, one IMS Gateway can provide redundancy for the other IMS Gateway. The first IMS Gateway connected to the SPDF is the primary IMS Gateway, and the other IMS Gateway is the redundant (standby) IMS Gateway. After the IMS Gateway asks the SAE to activate services for the session, the session information is passed to the SAE as activation attributes that are stored in the SAE's session store. The SAE synchronizes states with the standby IMS Gateway, so that the standby IMS Gateway can become the primary IMS Gateway if the primary IMS Gateway becomes unavailable.

- Related Topics**
- Overview of an IMS Environment on page 111
  - IMS and ETSI References on page 112

- [IMS Layers](#) on page 114
- [Configuring the IMS Software \(SRC CLI\)](#) on page 123



## CHAPTER 12

# Providing Services in IMS Networks (SRC CLI)

- Configuration Statements for IMS Support on page 121
- Configuring the IMS Software (SRC CLI) on page 123
- Configuring Initial Properties for IMS (SRC CLI) on page 123
- Configuring Directory Connection Properties for IMS (SRC CLI) on page 124
- Configuring Initial Directory Eventing Properties for IMS (SRC CLI) on page 125
- Configuring the Local Diameter Peer (SRC CLI) on page 126
- Configuring the Remote Diameter Peer (SRC CLI) on page 127
- Configuring Logging Destinations to Store Messages in a File (SRC CLI) on page 128
- Configuring Logging Destinations to Send Messages to the System Logging Facility (SRC CLI) on page 129
- Creating Grouped Configurations for IMS (SRC CLI) on page 130
- Configuring the Subscriber Type (SRC CLI) on page 131
- Configuring a NIC Proxy for IMS (SRC CLI) on page 132
- Configuring IMS for Failover (SRC CLI) on page 137
- Configuring the SAE for IMS on page 138
- Managing IMS (SRC CLI) on page 139
- Monitoring IMS (SRC CLI) on page 140
- Monitoring IMS (C-Web Interface) on page 141
- Example: Configuring JUNOS Policies for IMS (SRC CLI) on page 143

## Configuration Statements for IMS Support

---

Use the following configuration statements to configure IMS support at the **[edit]** hierarchy level.

```
slot number ims {
 shared shared;
}
slot number ims aracf-rq {
 protocol protocol;
```

```
 port port;
 address address;
 origin-host origin-host;
 origin-realm origin-realm;
 }
slot number ims aracf-rq peer name {
 address address;
 port port;
 origin-host origin-host;
 watchdog-timeout watchdog-timeout;
 incoming-queue-limit incoming-queue-limit;
}
slot number ims initial {
 static-dn static-dn;
 dynamic-dn dynamic-dn;
}
slot number ims 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 ims initial directory-eventing {
 eventing;
 signature-dn signature-dn;
 polling-interval polling-interval;
 event-base-dn event-base-dn;
 dispatcher-pool-size dispatcher-pool-size;
}
slot number ims logger name ...
slot number ims logger name file {
 filter filter;
 filename filename;
 rollover-filename rollover-filename;
 maximum-file-size maximum-file-size;
}
slot number ims logger name syslog {
 filter filter;
 host host;
 facility facility;
 format format;
}
```

- Related Topics**
- For more information about the configuration statements, see the *SRC PE CLI Command Reference*
  - Configuring the IMS Software (SRC CLI) on page 123
  - Overview of an IMS Environment on page 111
  - Example: Configuring JUNOS Policies for IMS (SRC CLI) on page 143

## Configuring the IMS Software (SRC CLI)

To configure the IMS software:

1. Configure initial properties, including the connection to the directory and directory monitoring properties.  
 See “Configuring Initial Properties for IMS (SRC CLI)” on page 123.  
 See “Configuring Directory Connection Properties for IMS (SRC CLI)” on page 124.  
 See “Configuring Initial Directory Eventing Properties for IMS (SRC CLI)” on page 125.
2. Configure the local and remote Diameter peers.  
 See “Configuring the Local Diameter Peer (SRC CLI)” on page 126.  
 See “Configuring the Remote Diameter Peer (SRC CLI)” on page 127.
3. Configure logging destinations.  
 See “Configuring Logging Destinations to Store Messages in a File (SRC CLI)” on page 128.  
 See “Configuring Logging Destinations to Send Messages to the System Logging Facility (SRC CLI)” on page 129.
4. Configure subscriber types.  
 See “Configuring the Subscriber Type (SRC CLI)” on page 131.
5. Configure the NIC proxies.  
 See “Configuring a NIC Proxy for IMS (SRC CLI)” on page 132.
6. Start the IMS process to provide the A-RACF Rq interface.  
 See “Starting the IMS Process (SRC CLI)” on page 139.

You must restart the IMS process after you commit a configuration change. To restart IMS, see “Restarting the IMS Process (SRC CLI)” on page 139.

- Related Topics**
- Overview of an IMS Environment on page 111
  - Monitoring IMS (SRC CLI) on page 140
  - Monitoring IMS (C-Web Interface) on page 141
  - Configuration Statements for IMS Support on page 121
  - SRC Software in the IMS Environment on page 117

## Configuring Initial Properties for IMS (SRC CLI)

Use the following configuration statements to configure initial properties for IMS:

```
slot number ims initial {
 static-dn static-dn;
```

```
dynamic-dn dynamic-dn;
}
```

To configure initial local properties:

1. From configuration mode, access the statement that configures the initial properties.

```
user@host# edit slot 0 ims initial
```

2. Specify the properties for IMS.

```
[edit slot 0 ims initial]
user@host# set ?
```

For more information about configuring local properties for SRC components, see [Configuring Basic Local Properties](#).

3. (Optional) Verify your configuration.

```
[edit slot 0 ims initial]
user@host# show
```

- Related Topics**
- Overview of an IMS Environment on page 111
  - Configuring the IMS Software (SRC CLI) on page 123
  - Configuring Initial Directory Eventing Properties for IMS (SRC CLI) on page 125
  - Configuration Statements for IMS Support on page 121

---

## Configuring Directory Connection Properties for IMS (SRC CLI)

---

Use the following configuration statements to configure directory connection properties for IMS:

```
slot number ims 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 statement that configures the directory connection properties.

```
user@host# edit slot 0 ims initial directory-connection
```

2. Specify the properties for IMS.



```
[edit slot 0 ims 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 ims initial directory-connection]
user@host# show
url ldap://127.0.0.1:389/;
principal cn=conf,o=Operators,<base>;
credentials *****;
```

- Related Topics**
- Overview of an IMS Environment on page 111
  - Configuring the IMS Software (SRC CLI) on page 123
  - Configuring Initial Properties for IMS (SRC CLI) on page 123
  - Configuring Initial Directory Eventing Properties for IMS (SRC CLI) on page 125
  - Configuration Statements for IMS Support on page 121

## Configuring Initial Directory Eventing Properties for IMS (SRC CLI)

Use the following configuration statements to configure directory eventing properties for IMS:

```
slot number ims 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 statement that configures the local properties.

```
user@host# edit slot 0 ims initial eventing
```

2. Specify the initial directory eventing properties for IMS.

```
[edit slot 0 ims 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 ims initial directory-eventing]
user@host# show
```

```
eventing;
polling-interval 30;
```

- Related Topics**
- Overview of an IMS Environment on page 111
  - Configuring the IMS Software (SRC CLI) on page 123
  - Configuring Initial Properties for IMS (SRC CLI) on page 123
  - Configuring Directory Connection Properties for IMS (SRC CLI) on page 124
  - Configuration Statements for IMS Support on page 121

---

## Configuring the Local Diameter Peer (SRC CLI)

---

Use the following configuration statements to configure the local Diameter peer:

```
slot number ims aracf-rq {
 protocol protocol;
 port port;
 address address;
 origin-host origin-host;
 origin-realm origin-realm;
}
```

To configure the local Diameter peer:

1. From configuration mode, access the statement that configures the Diameter peer.  

```
user@host# edit slot 0 ims aracf-rq
```
2. (Optional) Specify the protocol used for the transport layer.  

```
[edit slot 0 ims aracf-rq]
user@host# set protocol protocol
```
3. (Optional) Specify the port used for incoming connections.  

```
[edit slot 0 ims aracf-rq]
user@host# set port port
```
4. (Optional) Specify the IP address of the local peer.  

```
[edit slot 0 ims aracf-rq]
user@host# set address address
```
5. (Optional) Specify the Diameter identifier for the local endpoint that is the originator of the Diameter message.  

```
[edit slot 0 ims aracf-rq]
user@host# set origin-host origin-host
```
6. (Optional) Specify the Diameter identifier for the realm of the local endpoint that is the originator of the Diameter message.

```
[edit slot 0 ims aracf-rq]
user@host# set origin-realm origin-realm
```

7. (Optional) Verify your configuration.

```
[edit slot 0 ims aracf-rq]
user@host# show
protocol tcp;
port 3868;
address 127.0.0.1;
origin-host testserver;
origin-realm testrealm;
peer 1 {
 address 127.0.0.1;
 origin-host testclient;
}
```

- Related Topics**
- Overview of an IMS Environment on page 111
  - Configuring the IMS Software (SRC CLI) on page 123
  - Configuring the Remote Diameter Peer (SRC CLI) on page 127
  - Monitoring IMS (SRC CLI) on page 140

## Configuring the Remote Diameter Peer (SRC CLI)

Use the following configuration statements to configure the remote Diameter peer:

```
slot number ims aracf-rq peer name {
 address address;
 port port;
 origin-host origin-host;
 watchdog-timeout watchdog-timeout;
 incoming-queue-limit incoming-queue-limit;
}
```

To configure the remote Diameter peer:

1. From configuration mode, access the statement that configures the Diameter peer. In this sample procedure, the remote SPDF peer called primary-spdf is configured.

```
user@host# edit slot 0 ims aracf-rq peer primary-spdf
```

2. (Optional) Specify the IP address of the remote peer.

```
[edit slot 0 ims aracf-rq peer primary-spdf]
user@host# set address address
```

3. (Optional) Specify the port of the remote peer.

```
[edit slot 0 ims aracf-rq peer primary-spdf]
user@host# set port port
```

4. (Optional) Specify the Diameter identifier for the remote endpoint that is the originator of the Diameter message.

```
[edit slot 0 ims aracf-rq peer primary-spdf]
user@host# set origin-host origin-host
```

5. (Optional) Specify the watchdog timeout of the connection to the remote peer.

```
[edit slot 0 ims aracf-rq peer primary-spdf]
user@host# set watchdog-timeout watchdog-timeout
```

6. (Optional) Specify the size of the incoming message queue before the system rejects messages.

```
[edit slot 0 ims aracf-rq peer primary-spdf]
user@host# set incoming-queue-limit incoming-queue-limit
```

7. (Optional) Verify your configuration.

```
[edit slot 0 ims aracf-rq peer primary-spdf]
user@host# show
address 127.0.0.1;
origin-host testclient;
```

- Related Topics**
- Overview of an IMS Environment on page 111
  - Configuring the IMS Software (SRC CLI) on page 123
  - Configuring the Local Diameter Peer (SRC CLI) on page 126
  - Monitoring IMS (SRC CLI) on page 140

---

## Configuring Logging Destinations to Store Messages in a File (SRC CLI)

Use the following configuration statements to configure file logging for IMS:

```
slot number ims logger name ...
slot number ims logger name file {
 filter filter;
 filename filename;
 rollover-filename rollover-filename;
 maximum-file-size maximum-file-size;
}
```

To configure logging destinations to store log messages in a file:

1. From configuration mode, access the statement that configures the name and type of logging destination. In this sample procedure, the logging destination called log1 is configured.

```
user@host# edit slot 0 ims logger log1 file
```

2. Specify the properties for the logging destination.

```
[edit slot 0 ims logger log1 file]
```

```
user@host# set ?
```

For more information about configuring properties for the logging destination, see [Configuring a Component to Store Log Messages in a File \(SRC CLI\)](#).

3. (Optional) Verify your configuration.

```
[edit slot 0 ims logger log1 file]
user@host# show
filter /info-;
filename var/log/ims-a-racf-rq-info.log;
rollover-filename var/log/ims-a-racf-rq-info.alt;
maximum-file-size 2000000000;
```

- Related Topics**
- [Configuring Initial Properties for IMS \(SRC CLI\)](#) on page 123
  - [Configuring Logging Destinations to Send Messages to the System Logging Facility \(SRC CLI\)](#) on page 129

## Configuring Logging Destinations to Send Messages to the System Logging Facility (SRC CLI)

Use the following configuration statements to configure system logging for IMS:

```
slot number ims logger name ...
slot number ims logger name syslog {
 filter filter;
 host host;
 facility facility;
 format format;
}
```

To configure logging destinations to send log messages to the system logging facility:

1. From configuration mode, access the statement that configures the name and type of logging destination. In this sample procedure, the logging destination called log2 is configured.

```
user@host# edit slot 0 ims logger log2 syslog
```

2. Specify the properties for the logging destination.

```
[edit slot 0 ims logger log2 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 ims logger log2 syslog]
user@host# show
```

- Related Topics**
- [Configuring Initial Properties for IMS \(SRC CLI\)](#) on page 123

- [Configuring Logging Destinations to Store Messages in a File \(SRC CLI\)](#) on page 128

## Creating Grouped Configurations for IMS (SRC CLI)

---

Configuration groups allow you to share the IMS configuration with different IMS instances in the SRC network. You can also set up different configurations for different instances.

You can then create a grouped IMS configuration that is shared with some IMS instances. For example, if you create two different IMS groups called `config1` and `config2` within the shared IMS configuration, you could select the IMS configuration that should be associated with a particular IMS instance.

Use the **shared** option of the **slot *number* ims** statement to select the group for an IMS instance as part of the local configuration. Use the **shared ims group *name* configuration** statements to configure the group.

To select and configure a group:

1. From configuration mode, select a group for an IMS instance. For example, to select a group called `config1` in the root group:

```
[edit]
user@host# set slot 0 ims shared /config1
```

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 IMS configuration.

```
[edit]
user@host# edit shared ims group config1 ?
Possible completions:
<[Enter]> Execute this command
> configuration
| Pipe through a command
```

For more information, see “Configuring the IMS Software (SRC CLI)” on page 123.

### Related Topics

- [Configuring the IMS Software \(SRC CLI\)](#) on page 123
- [Configuring the Subscriber Type \(SRC CLI\)](#) on page 131
- [Configuring a NIC Proxy for IMS \(SRC CLI\)](#) on page 132
- [Configuring IMS for Failover \(SRC CLI\)](#) on page 137
- [Overview of an IMS Environment](#) on page 111

## Configuring the Subscriber Type (SRC CLI)

Use the following configuration statements to configure the subscriber type:

```
shared ims configuration subscriber-types (ip | login-name) {
 nic-proxy nic-proxy;
 subscriber-id-type (address | login-name | primary-user-name);
}
```

To configure the subscriber type:

1. From configuration mode, access the statement that configures the subscriber type.

```
user@host# edit shared ims configuration subscriber-types (ip | login-name)
```

where:

- **ip**—Subscriber type of IP address
- **login-name**—Subscriber type of login ID

For example, the subscriber type called ip is configured in this sample procedure.

```
user@host# edit shared ims configuration subscriber-types ip
```

2. Specify the namespace that defines the properties for the NIC proxy operations for the specified subscriber ID type. Each subscriber type must use a different NIC proxy. In this sample procedure, the namespace for the NIC proxy called ip is configured.

```
[edit shared ims configuration subscriber-types ip]
user@host# set nic-proxy ip
```

3. (Optional) Specify the type of information used to identify the subscriber.

```
[edit shared ims configuration subscriber-types ip]
user@host# set subscriber-id-type (address | login-name | primary-user-name)
```

where:

- **address**—Subscriber's IP address
- **login-name**—Subscriber's login name
- **primary-user-name**—Primary username

In this sample procedure, the subscriber ID type is specified as the subscriber IP address.

```
[edit shared ims configuration subscriber-types ip]
user@host# set subscriber-id-type address
```

4. (Optional) Verify your configuration.

```
[edit shared configuration subscriber-types ip]
user@host# show
```

- Related Topics**
- Configuring the IMS Software (SRC CLI) on page 123
  - Configuring Initial Properties for IMS (SRC CLI) on page 123
  - Configuring a NIC Proxy for IMS (SRC CLI) on page 132

## Configuring a NIC Proxy for IMS (SRC CLI)

---

Tasks to configure the NIC proxy are:

- Configuring Resolution Information for a NIC Proxy on page 132
- Changing the Configuration for the NIC Proxy Cache on page 133
- Configuring a NIC Proxy for NIC Replication on page 134
- Configuring NIC Test Data on page 136

### Configuring Resolution Information for a NIC Proxy

You create a NIC proxy for each subscriber type to be configured. Subscriber types that have different subscriber ID types can use the same NIC proxy.

Use the following configuration statements to configure the NIC proxy:

```
shared ims configuration nic-proxy-configuration name
shared ims configuration 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 statement that configures the NIC proxy configuration. In this sample procedure, the NIC proxy called *ip* is configured.  

```
user@host# edit shared ims configuration nic-proxy-configuration ip resolution
```
2. Specify the NIC resolver that this NIC proxy uses. This resolver must be the same as one that is configured on the NIC host.  

```
[edit shared ims configuration nic-proxy-configuration ip resolution]
user@host# set resolver-name resolver-name
```
3. Specify the NIC data type that the key provides for the NIC resolution.  

```
[edit shared ims configuration nic-proxy-configuration ip resolution]
user@host# set key-type key-type
```

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 ims 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 ims configuration nic-proxy-configuration ip resolution]
user@host# set value-type value-type
```

5. (Optional) If the key can have more than one value, specify that the key can have multiple corresponding values.

```
[edit shared ims 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 ims configuration nic-proxy-configuration ip resolution]
user@host# set constraints constraints
```

7. (Optional) Verify your configuration.

```
[edit shared ims configuration nic-proxy-configuration ip resolution]
user@host# show
resolver-name /realms/ip/A1;
key-type Ip;
value-type SaeId;
```

## 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 change values for the NIC proxy cache:

```
shared ims configuration 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 statement that specifies the NIC proxy configuration. In this sample procedure, the NIC proxy called ip is configured.

```
user@host# edit shared ims configuration nic-proxy-configuration ip cache
```

2. (Optional) Specify the maximum number of keys for which the NIC proxy retains data.

```
[edit shared ims 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 ims configuration nic-proxy-configuration ip cache]
user@host# set cache-cleanup-interval cache-cleanup-interval
```

4. (Optional) Specify how long an entry remains in the cache.

```
[edit shared ims configuration nic-proxy-configuration ip cache]
user@host# set cache-entry-age cache-entry-age
```

5. (Optional) Verify your configuration.

```
[edit shared configuration nic-proxy-configuration ip cache]
user@host# show
cache-size 10000;
cache-cleanup-interval 15;
```

## 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 Advanced editing level.

Use the following configuration statements to configure NIC host selection for a NIC proxy:

```
shared ims configuration nic-proxy-configuration name nic-host-selection {
 groups groups ;
 selection-criteria (roundRobin | randomPick | priorityList);
}
shared ims configuration 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 statement that specifies the NIC proxy configuration. In this sample procedure, the NIC proxy called ip is configured.

```
user@host# edit shared ims configuration nic-proxy-configuration ip
nic-host-selection
```

2. (Optional) Specify the list of groups of NIC hosts that the NIC proxy can contact for resolution requests.

```
[edit shared ims configuration nic-proxy-configuration ip nic-host-selection]
user@host# set groups groups
```

3. (Optional) 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 ims 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. (Optional) Verify your configuration.

```
[edit shared ims configuration nic-proxy-configuration ip
nic-host-selection]
user@host# show
groups ;
selection-criteria round-;
```

5. Access the statement that specifies the NIC proxy configuration for blacklisting—the process of handling nonresponsive NIC hosts.

```
[edit shared ims configuration nic-proxy-configuration ip nic-host-selection]
user@host# edit blacklisting
[edit shared ims configuration nic-proxy-configuration ip nic-host-selection
blacklisting]
```

6. (Optional) 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 ims configuration nic-proxy-configuration ip nic-host-selection
blacklisting]
user@host# set try-next-system-on-error
```

7. (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 ims configuration nic-proxy-configuration ip nic-host-selection
blacklisting]
user@host# set number-of-retries-before-blacklisting
number-of-retries-before-blacklisting
```

8. (Optional) Change the interval at which the NIC proxy attempts to connect to an unavailable NIC host. The default is 15 seconds.

```
[edit shared ims configuration nic-proxy-configuration ip nic-host-selection
blacklisting]
```

```
user@host# set blacklist-retry-interval blacklist-retry-interval
```

9. (Optional) Verify your configuration.

```
[edit shared ims configuration nic-proxy-configuration ip
nic-host-selection blacklisting]
user@host# show
try-next-system-on-error;
number-of-retries-before-blacklisting 3;
blacklist-retry-interval 15;
```

## Configuring NIC Test Data

To test a resolution without the 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 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.

Use the following configuration statements to configure a NIC proxy stub from the **[edit]** hierarchy level.

```
shared ims configuration nic-proxy-configuration name test-nic-bindings {
 use-test-bindings;
}
shared ims configuration nic-proxy-configuration name test-nic-bindings key-values
 name {
 value ;
 }
```

To use the NIC proxy stub for IMS:

1. In configuration mode, navigate to the NIC proxy configuration and specify the data type of the key you want to map to a value. In this sample procedure, the key `ip` is specified for the NIC proxy called `ip`.

```
[edit shared ims configuration nic-proxy-configuration ip]
user@host# set resolution key-type ip
```

2. Enable a NIC proxy stub for a resolution.

```
[edit shared ims configuration nic-proxy-configuration ip]
user@host# set test-nic-bindings use-test-bindings
```

3. Specify the values of the keys for testing. These statements are available at the Advanced CLI editing level.

```
[edit shared ims 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 ims configuration nic-proxy-configuration ip]
user@host# set test-nic-bindings key-values jane@virneo.com 192.0.2.30
```

- Related Topics**
- Configuring the IMS Software (SRC CLI) on page 123
  - Configuring NIC Test Data (SRC CLI)
  - Configuring a NIC Proxy for NIC Replication (SRC CLI)
  - Configuration Statements for IMS Support on page 121

## Configuring IMS for Failover (SRC CLI)

Use the following configuration statements to configure IMS for failover:

```
shared ims configuration redundancy {
 state-synchronization;
 state-sync-bulk-size state-sync-bulk-size;
 state-synchronization-timeout state-synchronization-timeout ;
}
```

To configure state synchronization with the SAE:

1. From configuration mode, access the statement that configures redundancy for IMS.

```
user@host# edit shared ims configuration redundancy
```

2. (Optional) Enable state synchronization from the SAE.

```
[edit shared ims configuration redundancy]
user@host# set state-synchronization
```

3. (Optional) Specify the number of events the SAE sends to IMS at one time during state synchronization.

```
[edit shared ims configuration redundancy]
user@host# set state-sync-bulk-size state-sync-bulk-size
```

4. (Optional) Specify the time to wait for the first full synchronization request from the SAE after starting or restarting IMS.

```
[edit shared ims configuration redundancy]
user@host# set state-synchronization-timeout state-synchronization-timeout
```

## Configuring the SAE for IMS

---

You must configure the SAE to recognize IMS by adding information about IMS to the SAE properties. Tasks for configuring the SAE for IMS are:

- Configuring IMS as an External Plug-In on page 138
- Configuring Event Publishers on page 138

### Configuring IMS as an External Plug-In

To configure an external plug-in for the SAE:

1. From configuration mode, access the statement that configures the external plug-ins.  
**user@host# edit shared sae configuration plug-ins name *name* external**
2. Configure the object reference of the external plug-in that is exported to the SAE.

**[edit shared sae configuration plug-ins name *name* external]**  
**user@host# set corba-object-reference *corba-object-reference***

where *corba-object-reference* is one of the following references:

- Path to the interoperable object reference (IOR) file in the format  
file:///opt/UMC/ims/var/run/ims.ior
- The corbaloc URL in the format corbaloc::*host*:9801/ASGIMS where *host* is the IP address of the C Series Controller or 127.0.0.1
- Common Object Services (COS) in the format  
corbaname::*host*:2809#ASGIMS/statesync/*hostname* or  
corbaname::*host*:2809/NameService#ASGIMS/statesync/*hostname* where
  - *host* is the IP address of the C Series Controller
  - *hostname* is the hostname of the C Series Controller

3. Specify the plug-in attributes.

**[edit shared sae configuration plug-ins name *name* external]**  
**user@host# set attributes ?**

Attributes for IMS are service-name, service-session-name, router-name, login-name, terminate-cause, property.

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 global service tracking events to the IMS. Any other events are ignored.

For information about configuring event publishers, see Special Types of Event Publishers.

**Related Topics** • Configuring the SAE for External Plug-Ins (SRC CLI)

- Special Types of Event Publishers

## Managing IMS (SRC CLI)

---

After you have configured IMS, you can perform these tasks:

- Starting the IMS Process (SRC CLI) on page 139
- Restarting the IMS Process (SRC CLI) on page 139
- Stopping the IMS Process (SRC CLI) on page 139
- Displaying IMS Status (SRC CLI) on page 140

### Starting the IMS Process (SRC CLI)

To start the IMS process:

```
user@host> enable component ims
```

The system responds with a start message. If IMS is already running, the system responds with a warning message.

- Related Topics**
- Configuring the IMS Software (SRC CLI) on page 123
  - Restarting the IMS Process (SRC CLI) on page 139
  - Stopping the IMS Process (SRC CLI) on page 139
  - Displaying IMS Status (SRC CLI) on page 140
  - Overview of an IMS Environment on page 111

### Restarting the IMS Process (SRC CLI)

You must restart the IMS process after you commit a configuration change.

To restart IMS:

```
user@host> restart component ims
```

The system responds with a start message. If IMS is already running, the system responds with a shutdown message and then a start message.

- Related Topics**
- Starting the IMS Process (SRC CLI) on page 139
  - Stopping the IMS Process (SRC CLI) on page 139
  - Displaying IMS Status (SRC CLI) on page 140
  - Overview of an IMS Environment on page 111

### Stopping the IMS Process (SRC CLI)

To stop the IMS process:

```
user@host> disable component ims
```

The system responds with a shutdown message. If IMS is not running when you issue the command, the system responds with the command prompt.

- Related Topics**
- Starting the IMS Process (SRC CLI) on page 139
  - Restarting the IMS Process (SRC CLI) on page 139
  - Displaying IMS Status (SRC CLI) on page 140
  - Overview of an IMS Environment on page 111

## Displaying IMS Status (SRC CLI)

**Purpose** Display IMS status.

**Action** user@host> **show component**

The system responds with a status message.

- Related Topics**
- Configuring the IMS Software (SRC CLI) on page 123
  - Stopping the IMS Process (SRC CLI) on page 139
  - Monitoring IMS (SRC CLI) on page 140
  - Monitoring IMS (C-Web Interface) on page 141
  - Overview of an IMS Environment on page 111

## Monitoring IMS (SRC CLI)

---

Monitoring tasks are:

- Viewing Server Process Information on page 140
- Viewing Statistics for the Rq Interface on page 140
- Viewing Information About Peers on page 141

## Viewing Server Process Information

**Purpose** View information about the IMS server process.

**Action** user@host> **show ims statistics aracf rq process**  
**Rq Server Process**  
Rq server up time (seconds) 692942  
Rq server up since 2007-03-13T15:30:48EDT  
Rq server threads 93  
Heap used (bytes) 16383752 (8%)  
Heap limit (bytes) 200000000

## Viewing Statistics for the Rq Interface

**Purpose** Monitor the current state of the A-RACF Rq interface.

**Action** user@host> **show ims statistics aracf rq**  
**ims aracf rq Statistics**



```

Rq Server Process
Rq server up time (seconds) 692920
Rq server up since 2007-03-13T15:30:48EDT
Rq server threads 93
Heap used (bytes) 16332120 (8%)
Heap limit (bytes) 200000000

```

## Viewing Information About Peers

**Purpose** View information about the peers.

**Action** To view information about all configured peers:

```
user@host> show ims aracf-rq peers
```

To view information about a specific peer:

```
user@host> show ims aracf-rq peers peer-name peer-name
```

To view the name and status of configured peers:

```
user@host> show ims aracf-rq peers brief
```

```
user@host> show ims aracf-rq peers peer-name peer-name brief
```

- Related Topics**
- Monitoring IMS (C-Web Interface) on page 141
  - Configuring the IMS Software (SRC CLI) on page 123
  - Displaying IMS Status (SRC CLI) on page 140

## Monitoring IMS (C-Web Interface)

You can monitor statistics for the server process and the A-RACF Rq interface with the C-Web interface by:

- Viewing Statistics for the Server Process on page 141
- Viewing Statistics for the A-RACF Rq Interface on page 142

## Viewing Statistics for the Server Process

**Purpose** View statistics for the server process.

**Action** Click **Monitor>IMS>Statistics>A-RACF>Rq>Process**.

The Process pane displays statistics for the server process.

The screenshot shows the Juniper SRX PE 4.0.x Monitor interface. The left sidebar contains a navigation menu with items: ACP, CLI, Component, Date, Disk, IMS (highlighted), Interfaces..., JPS, NIC, NTP, Redirect Server, Route..., SAE, Security, and System. The main content area displays the IMS Rq Server Process statistics. The top bar shows the user is logged in as 'sleswayball' and provides links for About, Refresh, and Logout. The breadcrumb trail is IMS > Statistics > A-RACF > Rq > Process.

| Rq Server Process           |                        |
|-----------------------------|------------------------|
| Rq server up time (seconds) | 8664                   |
| Rq server up since          | 2007-04-12T14:40:00EDT |
| Rq server threads           | 93                     |
| Heap used (bytes)           | 5026424 (3%)           |
| Heap limit (bytes)          | 200000000              |

Copyright © 2007, Juniper Networks, Inc. All Rights Reserved. Trademark Notice. Privacy. Juniper Your Net.

## Viewing Statistics for the A-RACF Rq Interface

**Purpose** View statistics for the A-RACF Rq interface.

**Action** Click **Monitor>IMS>Statistics>A-RACF>Rq**.

The Rq pane displays statistics for the A-RACF Rq interface.

The screenshot shows the Juniper SRX PE 4.0.x Monitor interface with the Rq pane selected. The left sidebar is the same as the previous screenshot. The main content area displays the IMS Rq Statistics. The top bar shows the user is logged in as 'sleswayball' and provides links for About, Refresh, and Logout. The breadcrumb trail is IMS > Statistics > A-RACF > Rq.

| Rq Server Process           |                        |
|-----------------------------|------------------------|
| Rq server up time (seconds) | 9373                   |
| Rq server up since          | 2007-04-12T14:40:00EDT |
| Rq server threads           | 93                     |
| Heap used (bytes)           | 6013200 (3%)           |
| Heap limit (bytes)          | 200000000              |

Copyright © 2007, Juniper Networks, Inc. All Rights Reserved. Trademark Notice. Privacy. Juniper Your Net.

- Related Topics**
- Monitoring IMS (SRC CLI) on page 140
  - Configuring the IMS Software (SRC CLI) on page 123
  - Displaying IMS Status (SRC CLI) on page 140

---

## Example: Configuring JUNOS Policies for IMS (SRC CLI)

---

For IMS environments, you can configure JUNOS policies. When you configure classify-traffic conditions, you can set up the software so that the SAE expands into multiple classifiers before it installs the policy on the router. If you enter a list of values in the source and destination network (IP address, mask, and IP operation) or port fields (for port-related protocols), the software creates a classifier for each possible combination of address and port. Note that the software does not expand classifiers for values that are entered as a range.

### Enabling Expansion of JUNOS Classify-Traffic Conditions

To enable the expansion of JUNOS classify-traffic conditions:

1. From configuration mode, access the statement that configures policy management properties on the SAE.

```
user@host# edit shared sae configuration policy-management-configuration
```

2. Specify whether or not the SAE expands the JUNOS classify-traffic conditions into multiple classifiers before it installs the policy on the router.

```
[edit shared sae configuration policy-management-configuration]
user@host# set enable-junos-classifier-expansion
```

- Related Topics**
- For more information about expanded classifiers, see Policy Information Model
  - Configuring Classify-Traffic Conditions (SRC CLI)
  - Policy Management Overview



# Testing IMS Service Sessions (SRC CLI)

- Testing Service Sessions for IMS on page 145
- Configuring the Test Environment for IMS Services (SRC CLI) on page 145
- Testing Service Sessions (SRC CLI) on page 149

## Testing Service Sessions for IMS

---

You can test service sessions by sending requests for:

- Activation
- Modification
- Deactivation

You can configure the settings for your test environment to easily test service sessions.

- Related Topics**
- Configuring the Test Environment for IMS Services (SRC CLI) on page 145
  - Testing Service Sessions (SRC CLI) on page 149

## Configuring the Test Environment for IMS Services (SRC CLI)

---

Configuring the settings for your test environment is optional. You can choose to configure the test settings and specify changes to the test settings.

- Configuring Settings for AAR Messages (SRC CLI) on page 145
- Configuring the Globally Unique Address (SRC CLI) on page 146
- Configuring Service Information for Media Types (SRC CLI) on page 147
- Configuring IP Flows for Media Types (SRC CLI) on page 148

## Configuring Settings for AAR Messages (SRC CLI)

Use the following command to configure the AA-Request (AAR) test message:

```
slot number ims aracf-rq test templates aar name {
 origin-host origin-host;
 origin-realm origin-realm;
 af-charging-identifier af-charging-identifier;
 authorization-lifetime authorization-lifetime;
```

```
user-name user-name;
specific-action (indication-of-bearer-release | indication-of-subscriber-detachment);
}
```

To configure the AAR message for the test environment:

1. From configuration mode, access the statement that configures the AAR message template with your settings.

```
user@host# edit slot number ims aracf-rq test templates aar name
```

2. Specify the Diameter identifier for the endpoint that is the originator of the Diameter message.

```
[edit slot number ims aracf-rq test templates aar name]
user@host# set origin-host origin-host
```

3. Specify the Diameter identifier for the realm of the endpoint that is the originator of the Diameter message.

```
[edit slot number ims aracf-rq test templates aar name]
user@host# set origin-realm origin-realm
```

4. (Optional) Specify the charging identifier for the application function (AF).

```
[edit slot number ims aracf-rq test templates aar name]
user@host# set af-charging-identifier af-charging-identifier
```

5. (Optional) Specify the timeout for the authorization.

```
[edit slot number ims aracf-rq test templates aar name]
user@host# set authorization-lifetime authorization-lifetime
```

6. (Optional) Specify the username.

```
[edit slot number ims aracf-rq test templates aar name]
user@host# set user-name user-name
```

7. (Optional) Specify the events for which notification is requested. If you do not configure this test setting, you must specify a value when testing service activations.

```
[edit slot number ims aracf-rq test templates aar name]
user@host# set specific-action (indication-of-bearer-release |
indication-of-subscriber-detachment)
```

where

- **indication-of-bearer-release**—Provides notification of a bearer's removal
- **indication-of-subscriber-detachment**—Provides notification of the subscriber detachment

## Configuring the Globally Unique Address (SRC CLI)

Use the following command to configure the globally unique address for the AAR test message:

```
slot number ims aracf-rq test templates aar name globally-unique-address {
framed-ip-address framed-ip-address;
}
```

To configure the globally unique address for the test environment:

1. From configuration mode, access the statement that configures the AAR message template with your settings.

```
user@host# edit slot number ims aracf-rq test templates aar name
globally-unique-address
```

2. (Optional) Specify the IPv4 address or the fully qualified domain name for the endpoint that is the originator of the Diameter message. If you do not configure this test setting, you must specify a value when testing service activations.

```
[edit slot number ims aracf-rq test templates aar name globally-unique-address]
user@host# set framed-ip-address framed-ip-address
```

## Configuring Service Information for Media Types (SRC CLI)

Use the following command to configure the service information that is used to determine QoS requirements for the media type:

```
slot number ims aracf-rq test templates aar name media-component-description
media-component-number {
af-application-identifier af-application-identifier;
media-type (audio | video | data | application | control | text | message | other);
max-requested-download-bandwidth max-requested-download-bandwidth;
max-requested-upload-bandwidth max-requested-upload-bandwidth;
flow-status (enabled | removed);
}
```

To configure the media component for the test environment:

1. From configuration mode, access the statement that configures the AAR message template with your settings. Specify the appropriate media component number.

```
user@host# edit slot number ims aracf-rq test templates aar name
media-component-description media-component-number
```

2. Specify the service name.

```
[edit slot number ims aracf-rq test templates aar name media-component-description
media-component-number]
user@host# set af-application-identifier af-application-identifier
```

3. (Optional) Specify the media type.

```
[edit slot number ims aracf-rq test templates aar name media-component-description
media-component-number]
user@host# set media-type (audio | video | data | application | control | text | message
| other)
```

4. (Optional) Specify the maximum download bandwidth requested.

```
[edit slot number ims aracf-rq test templates aar name media-component-description
media-component-number]
user@host# set max-requested-download-bandwidth
max-requested-download-bandwidth
```

5. (Optional) Specify the maximum upload bandwidth requested.

```
[edit slot number ims aracf-rq test templates aar name media-component-description
 media-component-number]
user@host# set max-requested-upload-bandwidth max-requested-upload-bandwidth
```

6. (Optional) Specify the action taken for the AAR.

```
[edit slot number ims aracf-rq test templates aar name media-component-description
 media-component-number]
user@host# set flow-status (enabled | removed)
```

where

- **enabled**—Commits resource reservation in both directions
- **removed**—Releases all resources associated with the corresponding resource reservation

## Configuring IP Flows for Media Types (SRC CLI)

Use the following command to configure the QoS and filters for the IP flows:

```
slot number ims aracf-rq test templates aar name media-component-description
 media-component-number media-sub-component flow-number {
 flow-description [flow-description...];
 max-requested-download-bandwidth max-requested-download-bandwidth;
 max-requested-upload-bandwidth max-requested-upload-bandwidth;
 }
```

To configure the media subcomponent for the test environment:

1. From configuration mode, access the statement that configures the AAR message template with your settings. Specify the appropriate flow number. These configuration settings override the media type settings.

```
user@host# edit slot number ims aracf-rq test templates aar name
 media-component-description media-component-number media-sub-component
 flow-number
```

2. Define the packet filter for the flow. The flow description AVP contains the classifier (or filter) information.

```
[edit slot number ims aracf-rq test templates aar name media-component-description
 media-component-number media-sub-component flow-number]
user@host# set flow-description [flow-description...]
```

The syntax of this AVP has the following restrictions:

- Only permit action should be used as action.
- No options should be used.

A subcomponent may include up to two flow descriptions (uplink and downlink), including:

- Direction (in—uplink, or out—downlink)
- Source IP address
- Destination IP address



- Source port
  - Destination port
  - Protocol
3. (Optional) Specify the maximum download bandwidth requested.
 

```
[edit slot number ims aracf-rq test templates aar name media-component-description
 media-component-number media-sub-component flow-number]
user@host# set max-requested-download-bandwidth
 max-requested-download-bandwidth
```
  4. (Optional) Specify the maximum upload bandwidth requested.
 

```
[edit slot number ims aracf-rq test templates aar name media-component-description
 media-component-number media-sub-component flow-number]
user@host# set max-requested-upload-bandwidth max-requested-upload-bandwidth
```

- Related Topics**
- Testing Service Sessions for IMS on page 145
  - Testing Service Sessions (SRC CLI) on page 149

## Testing Service Sessions (SRC CLI)

Tasks to test service sessions are:

- Testing Session Activations (SRC CLI) on page 149
- Testing Session Modifications (SRC CLI) on page 150
- Testing Session Deactivations (SRC CLI) on page 150

### Testing Session Activations (SRC CLI)

Use the following command to test the activation of service sessions:

```
test ims aracf-rq aar session-start aar-name aar-name <framed-ip-address
 framed-ip-address> <user-name user-name> <origin-host origin-host> <origin-realm
 origin-realm>
```

To test service session activations:

1. Issue the **test ims aracf-rq aar session-start** command.
2. To specify the name of the AAR message settings, use the **aar-name** option.
3. (Optional) To specify the subscriber's IP address, use the **framed-ip-address** option. If you specify a value, it will overwrite the configured test setting. If you did not configure this test setting, you must specify a value.
4. (Optional) To specify the subscriber name, use the **user-name** option. If you specify a value, it will overwrite the configured test setting. If you did not configure this test setting, you must specify a value.

5. (Optional) To specify the origin host for the simulator that generates the message, use the **origin-host** option. If you specify a value, it will overwrite the configured test setting.
6. (Optional) To specify the origin realm for the simulator that generates the message, use the **origin-realm** option. If you specify a value, it will overwrite the configured test setting.

## Testing Session Modifications (SRC CLI)

Use the following command to test service session modifications:

```
test ims aracf-rq aar session-modify session-id session-id aar-name aar-name
```

To test service session modifications:

1. Issue the **test ims aracf-rq aar session-modify** command.
2. To specify the session ID used to uniquely identify a user session, use the **session-id** option.
3. To specify the name of the AAR message settings, use the **aar-name** option.

## Testing Session Deactivations (SRC CLI)

Use the following command to test the deactivation of service sessions:

```
test ims aracf-rq str session-id session-id
```

To deactivate a particular session, use the **session-id** option.

- Related Topics**
- Testing Service Sessions for IMS on page 145
  - Configuring the Test Environment for IMS Services (SRC CLI) on page 145

## PART 3

# Index

- Index on page 153



# Index

## A

Application Services Gateway. *See* Web Services Gateway  
 arguments for scripts and methods.....4  
 ASG (Application Services Gateway). *See* Web Services Gateway

## B

B2B environments.....3  
 business partner responsibilities  
     Dynamic Service Activator.....7  
 business partners.....4  
 business-to-business environments.....3

## C

classify-traffic condition  
     expanded classifiers  
         configuring.....143  
 clients  
     gateway  
         testing.....53  
         Web Services Gateway.....75  
     Web Services Gateway.....3  
 configuration namespace.....4  
 conventions  
     notice icons.....xvii  
     text.....xvii  
 customer support.....xix  
     contacting JTAC.....xix

## D

documentation  
     comments on.....xix  
 dynamic properties.....4  
 Dynamic Service Activator  
     access constraints  
         defining.....22, 42  
     API.....75  
     attributes  
         access to.....28

configuring.....16  
     access to attributes.....28  
     access to methods and  
         scripts.....19, 22, 23, 26  
     access to service sessions.....28  
     C-Web interface.....39  
     general properties.....19  
     logging properties.....32  
     properties for clients and scripts.....22, 26  
     session database.....99  
     session handles.....21  
     subscriber types.....20  
     testing environment.....54, 61, 65, 69  
     video policy manager.....96  
 configuring with C-Web interface  
     access to methods and  
         scripts.....40, 42, 43  
     general properties.....40  
     logging properties.....43, 44  
     properties for clients and scripts.....42, 43  
 deleting  
     session database.....100  
 gateway extension  
     description.....4  
 groups  
     configuring.....15  
     configuring with C-Web interface.....38  
 interacting with Web application  
     server.....22, 42  
 loading  
     sample data.....34  
 methods.....76  
     access to.....22, 23, 42  
 monitoring  
     session database.....100  
     SRC CLI.....47  
 NIC proxies, monitoring  
     SRC CLI.....49  
 NIC proxies, viewing  
     C-Web interface.....52  
     SRC CLI.....49

|                                        |                |                                              |            |
|----------------------------------------|----------------|----------------------------------------------|------------|
| overview.....                          | 7              | methods                                      |            |
| redundancy.....                        | 9              | Dynamic Service Activator.....               | 76, 92, 93 |
| sample data.....                       | 34             | <b>N</b>                                     |            |
| scripts                                |                | namespace, configuration.....                | 4          |
| access to.....                         | 22, 26, 42, 43 | NIC (network information collector)          |            |
| service sessions                       |                | Dynamic Service Activator and.....           | 8          |
| access to.....                         | 28             | testing                                      |            |
| SOAP operations, monitoring            |                | test data.....                               | 136        |
| SRC CLI.....                           | 49             | NIC proxies                                  |            |
| SOAP operations, viewing               |                | cache, configuring                           |            |
| C-Web interface.....                   | 48, 51         | SRC CLI.....                                 | 133        |
| starting.....                          | 34             | configuration prerequisites.....             | 132        |
| C-Web interface.....                   | 45             | Dynamic Service Activator.....               | 21, 41     |
| statistics, viewing                    |                | NIC replication, configuring                 |            |
| C-Web interface.....                   | 51             | SRC CLI.....                                 | 133        |
| SRC CLI.....                           | 48             | resolution information, configuring          |            |
| testing.....                           | 53             | SRC CLI.....                                 | 132        |
| clients.....                           | 57, 62, 66, 70 | notice icons.....                            | xvii       |
| Web application gateway                |                | <b>O</b>                                     |            |
| client.....                            | 54, 61, 65, 69 | operation                                    |            |
|                                        |                | Dynamic Service Activator.....               | 8          |
| <b>E</b>                               |                | <b>P</b>                                     |            |
| events, publishing.....                | 138            | priorityList.....                            | 135        |
| expanded classifiers                   |                | publishing events.....                       | 138        |
| configuring.....                       | 143            | <b>R</b>                                     |            |
| <b>G</b>                               |                | randomPick.....                              | 135        |
| gateway                                |                | redundancy                                   |            |
| SRC.....                               | 3, 4           | Dynamic Service Activator.....               | 9          |
| gateway extension.....                 | 4              | roundRobin.....                              | 135        |
| <b>I</b>                               |                | <b>S</b>                                     |            |
| IMS service sessions                   |                | SAE (service activation engine)              |            |
| configuring.....                       | 145            | managing via external applications.....      | 3, 8       |
| testing environment.....               | 145            | SAE (service activation engine), configuring |            |
| testing.....                           | 145            | Dynamic Service Activator.....               | 96         |
| SRC CLI.....                           | 149            | IMS.....                                     | 138        |
| <b>L</b>                               |                | scripts                                      |            |
| logging properties                     |                | running on SAE.....                          | 8          |
| Dynamic Service Activator.....         | 32             | services                                     |            |
| configuring with C-Web interface.....  | 43, 44         | managing via external application.....       | 3          |
| <b>M</b>                               |                | sessions                                     |            |
| managing                               |                | service                                      |            |
| SAE via external applications.....     | 3, 8           | testing.....                                 | 145        |
| services via external application..... | 3              |                                              |            |
| manuals                                |                |                                              |            |
| comments on.....                       | xix            |                                              |            |

|                                                   |    |
|---------------------------------------------------|----|
| SOAP                                              |    |
| interfaces, public.....                           | 75 |
| requests.....                                     | 8  |
| SRC owners.....                                   | 5  |
| Dynamic Service Activator.....                    | 7  |
| SRC SOAP Gateway. <i>See</i> Web Services Gateway |    |
| static properties.....                            | 5  |
| support, technical <i>See</i> technical support   |    |

## T

|                               |      |
|-------------------------------|------|
| technical support             |      |
| contacting JTAC.....          | xix  |
| text conventions defined..... | xvii |

## W

|                                                    |                |
|----------------------------------------------------|----------------|
| Web application gateway client                     |                |
| configuring.....                                   | 54, 61, 65, 69 |
| Web application server.....                        | 5              |
| interacting with Dynamic Service                   |                |
| Activator.....                                     | 22, 42         |
| Web applications                                   |                |
| Web Services Gateway.....                          | 3              |
| Web Services Description Language. <i>See</i> WSDL |                |
| Web Services Gateway.....                          | 3              |
| clients.....                                       | 3              |
| managing.....                                      | 75             |
| testing.....                                       | 53             |
| Web application.....                               | 4              |
| wholesaler-retailer environments.....              | 3              |
| WSDL files.....                                    | 75             |

