Alcatel-Lucent 5620

SERVICE AWARE MANAGER | RELEASE 9.0 R2

XML OSS INTERFACE DEVELOPER GUIDE

3HE 06498 AAAB TQZZA Edition 01

Alcatel-Lucent Proprietary
This document contains proprietary information of Alcatel-Lucent and is not to be disclosed or used except in accordance with applicable agreements.
Copyright 2011 © Alcatel-Lucent. All rights reserved.

Alcatel-Lucent assumes no responsibility for the accuracy of the information presented, which is subject to change without notice.

Alcatel, Lucent, Alcatel-Lucent, the Alcatel-Lucent logo, and TiMetra are registered trademarks of Alcatel-Lucent. All other trademarks are the property of their respective owners.

Copyright 2011 Alcatel-Lucent.
All rights reserved.

Disclaimers

Alcatel-Lucent products are intended for commercial uses. Without the appropriate network design engineering, they must not be sold, licensed or otherwise distributed for use in any hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life-support machines, or weapons systems, in which the failure of products could lead directly to death, personal injury, or severe physical or environmental damage. The customer hereby agrees that the use, sale, license or other distribution of the products for any such application without the prior written consent of Alcatel-Lucent, shall be at the customer's sole risk. The customer hereby agrees to defend and hold Alcatel-Lucent harmless from any claims for loss, cost, damage, expense or liability that may arise out of or in connection with the use, sale, license or other distribution of the products in such applications.

This document may contain information regarding the use and installation of non-Alcatel-Lucent products. Please note that this information is provided as a courtesy to assist you. While Alcatel-Lucent tries to ensure that this information accurately reflects information provided by the supplier, please refer to the materials provided with any non-Alcatel-Lucent product and contact the supplier for confirmation. Alcatel-Lucent assumes no responsibility or liability for incorrect or incomplete information provided about non-Alcatel-Lucent products.

However, this does not constitute a representation or warranty. The warranties provided for Alcatel-Lucent products, if any, are set forth in contractual documentation entered into by Alcatel-Lucent and its customers.

This document was originally written in English. If there is any conflict or inconsistency between the English version and any other version of a document, the English version shall prevail.

When printed by Alcatel-Lucent, this document is printed on recycled paper.

Alcatel-Lucent License Agreement

SAMPLE END USER LICENSE AGREEMENT

1. LICENSE

1.1 Subject to the terms and conditions of this Agreement, Alcatel-Lucent grants to Customer and Customer accepts a nonexclusive, nontransferable license to use any software and related documentation provided by Alcatel-Lucent pursuant to this Agreement ("Licensed Program") for Customer's own internal use, solely in conjunction with hardware supplied or approved by Alcatel-Lucent. In case of equipment failure, Customer may use the Licensed Program on a backup system, but only for such limited time as is required to rectify the failure.

1.2 Customer acknowledges that Alcatel-Lucent may have encoded within the Licensed Program optional functionality and capacity (including, but not limited to, the number of equivalent nodes, delegate workstations, paths and partitions), which may be increased upon the purchase of the applicable license extensions.

1.3 Use of the Licensed Program may be subject to the issuance of an application key, which shall be conveyed to the Customer in the form of a Supplement to this End User License Agreement. The purchase of a license extension may require the issuance of a new application key.

2. PROTECTION AND SECURITY OF LICENSED PROGRAMS

2.1 Customer acknowledges and agrees that the Licensed Program contains proprietary and confidential information of Alcatel-Lucent and its third party suppliers, and agrees to keep such information confidential. Customer shall not disclose the Licensed Program except to its employees having a need to know, and only after they have been advised of its confidential and proprietary nature and have agreed to protect same.

2.2 All rights, title and interest in and to the Licensed Program, other than those expressly granted to Customer herein, shall remain vested in Alcatel-Lucent or its third party suppliers. Customer shall not, and shall prevent others from copying, translating, modifying, creating derivative works, reverse engineering, decompiling, encumbering or otherwise using the Licensed Program except as specifically authorized under this Agreement. Notwithstanding the foregoing, Customer is authorized to make one copy for its archival purposes only. All appropriate copyright and other proprietary notices and legends shall be placed on all Licensed Programs supplied by Alcatel-Lucent, and Customer shall maintain and reproduce such notices on any full or partial copies made by it.

3. TERM

3.1 This Agreement shall become effective for each Licensed Program upon delivery of the Licensed Program to Customer.

3.2 Alcatel-Lucent may terminate this Agreement: (a) upon notice to Customer if any amount payable to Alcatel-Lucent is not paid within thirty (30) days of the date on which payment is due; (b) if Customer becomes bankrupt, makes an assignment for the benefit of its creditors, or if its assets vest or become subject to the rights of any trustee, receiver or other administrator; (c) if bankruptcy, reorganization or insolvency proceedings are instituted against Customer and not dismissed within 15 days; or (d) if Customer breaches a material provision of this Agreement and such breach is not rectified within 15 days of receipt of notice of the breach from Alcatel-Lucent.

3.3 Upon termination of this Agreement, Customer shall return or destroy all copies of the Licensed Program. All obligations of Customer arising prior to termination, and those obligations relating to confidentiality and nonuse, shall survive termination.

4. CHARGES

4.1 Upon shipment of the Licensed Program, Alcatel-Lucent will invoice Customer for all fees, and any taxes, duties and other charges. Customer will be invoiced for any license extensions upon delivery of the new software application key or, if a new application key is not required, upon delivery of the extension. All amounts shall be due and payable within thirty (30) days of receipt of invoice, and interest will be charged on any overdue amounts at the rate of 1 1/2% per month (19.6% per annum).

5. SUPPORT AND UPGRADES

5.1 Customer shall receive software support and upgrades for the Licensed Program only to the extent provided for in the applicable Alcatel-Lucent software support policy in effect from time to time, and upon payment of any applicable fees. Unless expressly excluded, this Agreement shall be deemed to apply to all updates, upgrades, revisions, enhancements and other software which may be supplied by Alcatel-Lucent to Customer from time to time.

6. WARRANTIES AND INDEMNIFICATION

6.1 Alcatel-Lucent warrants that the Licensed Program as originally delivered to Customer will function substantially in accordance with the functional description set out in the associated user documentation for a period of 90 days from the date of shipment, when used in accordance with the user documentation. Alcatel-Lucent's sole liability and Customer's sole remedy for a breach of this warranty shall be Alcatel-Lucent's good faith efforts to rectify the nonconformity or, if after repeated efforts Alcatel-Lucent is unable to rectify the nonconformity, Alcatel-Lucent shall accept return of the Licensed Program and shall refund to Customer all amounts paid in respect thereof. This warranty is available only once in respect of each Licensed Program, and is not renewed by the payment of an extension charge or upgrade fee.

6.2 ALCATEL-LUCENT EXPRESSLY DISCLAIMS ALL OTHER WARRANTIES, REPRESENTATIONS, COVENANTS OR CONDITIONS OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, WARRANTIES OR REPRESENTATIONS OF WORKMANSHIP, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, DURABILITY, OR THAT THE OPERATION OF THE LICENSED PROGRAM WILL BE ERROR FREE OR THAT THE LICENSED PROGRAMS WILL NOT INFRINGE UPON ANY THIRD PARTY RIGHTS.

6.3 Alcatel-Lucent shall defend and indemnify Customer in any action to the extent that it is based on a claim that the Licensed Program furnished by Alcatel-Lucent infringes any patent, copyright, trade secret or other intellectual property right, provided that Customer notifies Alcatel-Lucent within ten (10) days of the existence of the claim, gives Alcatel-Lucent sole control of the litigation or settlement of the claim, and provides all such assistance as Alcatel-Lucent may reasonably require. Notwithstanding the foregoing, Alcatel-Lucent shall have no liability if the claim results from any modification or unauthorized use of the Licensed Program by Customer, and Customer shall defend and indemnify Alcatel-Lucent against any such claim.

6.4 Alcatel-Lucent Products are intended for standard commercial uses. Without the appropriate network design engineering, they must not be sold, licensed or otherwise distributed for use in any hazardous environments requiring fail safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life-support machines, or weapons systems, in which the failure of products could lead directly to death, personal injury, or severe physical or environmental damage. The Customer hereby agrees that the use, sale, license or other distribution of the Products for any such application without the prior written consent of Alcatel-Lucent, shall be at the Customer's sole risk. The Customer also agrees to defend and hold Alcatel-Lucent harmless from any claims for loss, cost, damage, expense or liability that may arise out of or in connection with the use, sale, license or other distribution of the Products in such applications.

7. LIMITATION OF LIABILITY

7.1 IN NO EVENT SHALL THE TOTAL COLLECTIVE LIABILITY OF ALCATEL-LUCENT, ITS EMPLOYEES, DIRECTORS, OFFICERS OR AGENTS FOR ANY CLAIM, REGARDLESS OF VALUE OR NATURE, EXCEED THE AMOUNT PAID UNDER THIS AGREEMENT FOR THE LICENSED PROGRAM THAT IS THE SUBJECT MATTER OF THE CLAIM. IN NO EVENT SHALL THE TOTAL COLLECTIVE LIABILITY OF ALCATEL-LUCENT, ITS EMPLOYEES, DIRECTORS, OFFICERS OR AGENTS FOR ALL CLAIMS EXCEED THE TOTAL AMOUNT PAID BY CUSTOMER TO ALCATEL-LUCENT HEREUNDER. NO PARTY SHALL BE LIABLE FOR ANY INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES, WHETHER OR NOT SUCH DAMAGES ARE FORESEEABLE, AND/OR THE PARTY HAD BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7.2 The foregoing provision limiting the liability of Alcatel-Lucent's employees, agents, officers and directors shall be deemed to be a trust provision, and shall be enforceable by such employees, agents, officers and directors as trust beneficiaries.

8. GENERAL

8.1 Under no circumstances shall either party be liable to the other for any failure to perform its obligations (other than the payment of any monies owing) where such failure results from causes beyond that party's reasonable control.

8.2 This Agreement constitutes the entire agreement between Alcatel-Lucent and Customer and supersedes all prior oral and written communications. All amendments shall be in writing and signed by authorized representatives of both parties.

8.3 If any provision of this Agreement is held to be invalid, illegal or unenforceable, it shall be severed and the remaining provisions shall continue in full force and effect.

8.4 The Licensed Program may contain freeware or shareware obtained by Alcatel-Lucent from a third party source. No license fee has been paid by Alcatel-Lucent for the inclusion of any such freeware or shareware, and no license fee is charged to Customer for its use. The Customer agrees to be bound by any license agreement for such freeware or shareware. CUSTOMER ACKNOWLEDGES AND AGREES THAT THE THIRD PARTY SOURCE PROVIDES NO WARRANTIES AND SHALL HAVE NO LIABILITY WHATSOEVER IN RESPECT OF CUSTOMER'S POSSESSION AND/OR USE OF THE FREEWARE OR SHAREWARE.

8.5 Alcatel-Lucent shall have the right, at its own expense and upon reasonable written notice to Customer, to periodically inspect Customer's premises and such documents as it may reasonably require, for the exclusive purpose of verifying Customer's compliance with its obligations under this Agreement.

8.6 All notices shall be sent to the parties at the addresses listed above, or to any such address as may be specified from time to time. Notices shall be deemed to have been received five days after deposit with a post office when sent by registered or certified mail, postage prepaid and receipt requested.

8.7 If the Licensed Program is being acquired by or on behalf of any unit or agency of the United States Government, the following provision shall apply: If the Licensed Program is supplied to the Department of Defense, it shall be classified as "Commercial Computer Software" and the United States Government is acquiring only "restricted rights" in the Licensed Program as defined in DFARS 227-7202-1(a) and 227.7202-3(a), or equivalent. If the Licensed Program is supplied to any other unit or agency of the United States Government, rights will be defined in Clause 52.227-19 or 52.227-14 of the FAR, or if acquired by NASA, Clause 18-52.227-86(d) of the NASA Supplement to the FAR, or equivalent. If the software was acquired under a contract subject to the October 1988 Rights in Technical Data and Computer Software regulations, use, duplication and disclosure by the Government is subject to the restrictions set forth in DFARS 252-227.7013(c)(1)(ii) 1988, or equivalent.

8.8 Customer shall comply with all export regulations pertaining to the Licensed Program in effect from time to time. Without limiting the generality of the foregoing, Customer expressly warrants that it will not directly or indirectly export, reexport, or transship the Licensed Program in violation of any export laws, rules or regulations of Canada, the United States or the United Kingdom.

8.9 No term or provision of this Agreement shall be deemed waived and no breach excused unless such waiver or consent is in writing and signed by the party claimed to have waived or consented. The waiver by either party of any right hereunder, or of the failure to perform or of a breach by the other party, shall not be deemed to be a waiver of any other right hereunder or of any other breach or failure by such other party, whether of a similar nature or otherwise.

8.10 This Agreement shall be governed by and construed in accordance with the laws of the Province of Ontario. The application of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded.

Preface

The Preface provides general information about the 5620 Service Aware Manager documentation suite.

Note — You can use the Search function of Acrobat Reader (File→Search) to find a term in a PDF of this document. To refine your search, use appropriate search options (for example, search for whole words only or enable case-sensitive searching). You can also search for a term in multiple PDFs at once. For more information, see the Help for Acrobat Reader.

5620 SAM documentation suite

The 5620 SAM documentation suite describes the 5620 SAM and the associated network management of its supported devices. Contact your Alcatel-Lucent support representative for information about specific network or facility considerations.

Table 1 lists the documents in the 5620 SAM documentation suite.

Table 1 5620 SAM customer documentation suite

Guide	Description
5620 SAM core documentation	
5620 SAM Planning Guide	The *5620 SAM Planning Guide* provides information about 5620 SAM scalability and recommended hardware configurations.

(1 of 4)

Guide	Description
5620 SAM | 5650 CPAM Installation and Upgrade Guide	The *5620 SAM | 5650 CPAM Installation and Upgrade Guide* provides OS considerations, configuration information, and procedures for the following: • installing, upgrading, and uninstalling 5620 SAM and 5650 CPAM software in standalone and redundant deployments • 5620 SAM system migration to a different system • conversion from a standalone to a redundant 5620 SAM system
5620 SAM User Guide	The *5620 SAM User Guide* provides information about using the 5620 SAM to manage the service-aware IP/MPLS network, including GUI basics, commissioning, service configuration, and policy management. The *5620 SAM User Guide* uses a task-based format. Each chapter contains: • a workflow that describes the steps for configuring and using the functionality • detailed procedures that list the configurable parameters on the associated forms 5620 SAM management information specific to LTE network elements is covered in the *5620 SAM LTE ePC User Guide* and *5620 SAM LTE RAN User Guide*. 5620 SAM management information specific to 1830 PSS network elements is covered in the *5620 SAM Optical User Guide*.
5620 SAM Parameter Guide	The *5620 SAM Parameter Guide* provides: • parameter descriptions that include value ranges and default values • parameter options and option descriptions • parameter and option dependencies • parameter mappings to the 5620 SAM-O XML equivalent property names There are dynamic links between the procedures in the *5620 SAM User Guide* and the parameter descriptions in the *5620 SAM Parameter Guide*. See Procedure 2 for more information. Parameters specific to LTE network elements are covered in the *5620 SAM LTE Parameter Reference*. Parameters specific to 1830 PSS network elements are covered in the *5620 SAM Optical Parameter Reference*.
5620 SAM Statistics Management Guide	The *5620 SAM Statistics Management Guide* provides information about how to configure performance and accounting statistics collection and how to view counters using the 5620 SAM. Network examples are included.
5620 SAM Scripts and Templates Developer Guide	The *5620 SAM Scripts and Templates Developer Guide* provides information that allows you to develop, manage, and execute CLI-based or XML-based scripts or templates. The guide is intended for developers, skilled administrators, and operators who are expected to be familiar with the following: • CLI scripting, XML, and the Velocity engine • basic scripting or programming • 5620 SAM functions

5620 SAM Troubleshooting Guide	The *5620 SAM Troubleshooting Guide* provides task-based procedures and user documentation to:
	• help resolve issues in the managed and management networks
	• identify the root cause and plan corrective action for:
	  • alarm conditions on a network object or customer service
	  • problems on customer services with no associated alarms
	• list problem scenarios, possible solutions, and tools to help check:
	  • network management LANs
	  • network management platforms and operating systems
	  • 5620 SAM client GUIs and client OSS applications
	  • 5620 SAM servers
	  • 5620 SAM databases

(2 of 4)

Guide	Description
5620 SAM Maintenance Guide	The *5620 SAM Maintenance Guide* provides procedures for: • generating baseline information for 5620 SAM applications • performing daily, weekly, monthly, and as-required maintenance activities for 5620 SAM-managed networks
5620 SAM Integration Guide	The *5620 SAM Integration Guide* provides procedures to allow the 5620 SAM to integrate with additional components.
5620 SAM System Architecture Guide	The *5620 SAM System Architecture Guide* is intended for technology officers and network planners to increase their knowledge of the 5620 SAM software structure and components. It describes the system structure, software components, and interfaces of the 5620 SAM. In addition, 5620 SAM fault tolerance, security, and network management capabilities are discussed from an architectural perspective.
5620 SAM Supervision Module User Guide	The *5620 SAM Supervision Module User Guide* provides information about how to configure and use the web-based 5620 SAM Supervision Module for fault management and at-a-glance network element monitoring.
5620 SAM Network Element Compatibility Guide	The *5620 SAM Network Element Compatibility Guide* provides release-specific information about the compatibility of managed device features in 5620 SAM releases.
5620 SAM Release Description	The *5620 SAM Release Description* provides information about the new features associated with a 5620 SAM software release.
5620 SAM Glossary	The *5620 SAM Glossary* defines terms and acronyms used in all of the 5620 SAM documentation, including 5620 SAM LTE documentation.
5620 SAM XML OSS Interface Developer Guide	The *5620 SAM XML OSS Interface Developer Guide* provides information that allows you to: • use the 5620 SAM XML OSS interface to access network management information • learn about the information model associated with the managed network • develop OSS applications using the packaged methods, classes, data types, and objects necessary to manage 5620 SAM functions
5620 SAM LTE documentation	
5620 SAM LTE ePC User Guide	The *5620 SAM LTE ePC User Guide* describes how to discover, configure, and manage LTE ePC devices using the 5620 SAM. The guide is intended for LTE ePC network planners, administrators, and operators. Alcatel-Lucent recommends that you review the entire *5620 SAM LTE ePC User Guide* before you attempt to use the 5620 SAM in your LTE network.
5620 SAM LTE RAN User Guide	The *5620 SAM LTE RAN User Guide* describes how to discover, configure, and manage the eNodeB using the 5620 SAM. The guide is intended for LTE RAN network planners, administrators, and operators. Alcatel-Lucent recommends that you review the entire *5620 SAM LTE RAN User Guide* before you attempt to use the 5620 SAM in your LTE network.
5620 SAM LTE Parameter Reference	The *5620 SAM LTE Parameter Reference* provides a list of all LTE ePC and LTE RAN parameters supported in the 5620 SAM.
5620 SAM LTE Alarm Reference	The *5620 SAM LTE Alarm Reference* provides a list of LTE ePC and LTE RAN alarms that can be reported in the 5620 SAM GUI.
5620 SAM 3GPP OSS Interface Developer Guide	The *5620 SAM 3GPP OSS Interface Developer Guide* describes the components and architecture of the 3GPP OSS interface to the 5620 SAM. It includes procedures and samples to assist OSS application developers to use the 3GPP interface to manage LTE devices.
5620 SAM 3GPP OSS Interface Compliance Statements	The *5620 SAM 3GPP OSS Interface Compliance Statements* document describes the compliance of the 5620 SAM 3GPP OSS interface with the 3GPP standard.
5620 SAM LTE RAN Release Description	The *5620 SAM LTE RAN Release Description* provides information about the LTE RAN features associated with the release.

(3 of 4)

Guide	Description
5620 SAM optical documentation	
5620 SAM Optical User Guide	The *5620 SAM Optical User Guide* describes how to discover, configure, and manage optical devices using the 5620 SAM. The guide is intended for optical network planners, administrators, and operators. Alcatel-Lucent recommends that you review the entire *5620 SAM Optical User Guide* before you attempt to use the 5620 SAM in your network.
5620 SAM Optical Parameter Reference	The *5620 SAM Optical Parameter Reference* provides a list of all optical device parameters supported in the 5620 SAM.
5620 SAM Optical Alarm Reference	The *5620 SAM Optical Alarm Reference* provides a list of optical device alarms that can be reported in the 5620 SAM GUI.

(4 of 4)

Procedure 1 To find the 5620 SAM user documentation

The user documentation is available from the following sources:

- the User_Documentation directory on the product DVD-ROM
- Help→5620 SAM User Documentation in the 5620 SAM client GUI main menu

Note — Users of Mozilla browsers may receive an error message when using the User Documentation Index page (index.html) to open the PDF files in the 5620 SAM documentation suite. The offline storage and default cache values used by the browsers are the cause of the error message.

Alcatel-Lucent recommends changing the offline storage (Mozilla Firefox) or cache (Mozilla 1.7) values to 100 Mbytes to eliminate the error message.

Procedure 2 To view parameter descriptions from the *5620 SAM User Guide*

You can click on a parameter name in a *5620 SAM User Guide* procedure to open the matching parameter description in the *5620 SAM Parameter Guide*. Ensure the following conditions are true beforehand:

- the *5620 SAM Parameter Guide* and *5620 SAM User Guide* are located in the same directory
- Adobe Reader Release 5.0 or later is installed

1. To view a parameter description when both the *5620 SAM User Guide* and the *5620 SAM Parameter Guide* are open in Adobe Acrobat, click on the parameter name in the *5620 SAM User Guide*.

   The parameter description is displayed in the *5620 SAM Parameter Guide*.

2. To view a parameter description when only the *5620 SAM User Guide* is open in Adobe Acrobat:

   i. Click on a parameter name in a procedure in the *5620 SAM User Guide*. The *5620 SAM User Guide* closes and the *5620 SAM Parameter Guide* opens to display the parameter description.

   ii. Double-click on the Previous View button in Adobe Acrobat (or press Alt + ←) to re-open the *5620 SAM User Guide*. The *5620 SAM User Guide* opens and displays the parameter from step i.

Prerequisites

Readers of the 5620 SAM documentation suite are assumed to be familiar with the following:

- 5620 SAM software structure and components
- 5620 SAM GUI operations and tools
- typical 5620 SAM management tasks and procedures
- device and network management concepts

Conventions

Table 2 lists the conventions that are used throughout the documentation.

Table 2 Documentation conventions

Convention	Description	Example
Key name	Press a keyboard key	Delete
Italics	Identifies a variable	*hostname*

(1 of 2)

Convention	Description	Example
Key+Key	Type the appropriate consecutive keystroke sequence	CTRL+G
Key–Key	Type the appropriate simultaneous keystroke sequence	CTRL–G
*	An asterisk is a wildcard character, which means “any character” in a search argument.	log_file*.txt
↵	Press the Return key	↵
—	An em dash indicates there is no information.	—
→	Indicates that a cascading submenu results from selecting a menu item	Policies→Alarm Policies

(2 of 2)

Procedures with options or substeps

When there are options in a procedure, they are identified by letters. When there are substeps in a procedure, they are identified by Roman numerals.

Example of options in a procedure

At step 1, you can choose option a or b. At step 2, you must do what the step indicates.

1 This step offers two options. You must choose one of the following.

- **a** This is one option.
- **b** This is another option.

2 You must perform this step.

Example of substeps in a procedure

At step 1, you must perform a series of substeps within a step. At step 2, you must do what the step indicates.

1 This step has a series of substeps that you must perform to complete the step. You must perform the following substeps.

- **i** This is the first substep.
- **ii** This is the second substep.
- **iii** This is the third substep.

2 You must perform this step.

Measurement conventions

Measurements in this document are expressed in metric units and follow the Système international d’unités (SI) standard for abbreviation of metric units. If imperial measurements are included, they appear in brackets following the metric unit.

Table 3 lists the measurement symbols used in this document.

Table 3 Bits and bytes conventions

Measurement	Symbol
bit	b
byte	byte
kilobits per second	kb/s

Important information

The following conventions are used to indicate important information:

Warning — Warning indicates that the described activity or situation may, or will, cause equipment damage or serious performance problems.

Caution — Caution indicates that the described activity or situation may, or will, cause service interruption.

Note — Notes provide information that is, or may be, of special interest.

Contents

Preface **ix**

5620 SAM documentation suite ix
Procedure 1 To find the 5620 SAM user documentation xii
Procedure 2 To view parameter descriptions from the 5620 SAM User Guide xiii
Prerequisites xiii
Conventions xiii
Procedures with options or substeps xiv
Measurement conventions xiv
Important information xv

Getting started

1 — Product and support overview **1-1**

1.1 5620 SAM product suite 1-2
1.2 5620 SAM-O overview 1-3
5620 SAM-O client 1-5
1.3 Supported technologies and configuration 1-5
Redundant network configurations 1-5
1.4 Supported devices 1-6
1.5 Before you begin 1-7
1.6 Open interfaces professional support 1-8

2 — 5620 SAM-O release features 2-1

2.1 5620 SAM-O Release 9.0 functionality 2-2
2.2 5620 SAM-O Release 8.0 functionality 2-2
2.3 5620 SAM-O Release 7.0 functionality 2-4
2.4 5620 SAM-O Release 6.1 functionality 2-5
2.5 5620 SAM-O Release 6.0 functionality 2-5
2.6 5620 SAM-O Release 5.0 functionality 2-7
2.7 5620 SAM-O Release 4.0 functionality 2-11
2.8 Understanding changes, modifications, and deprecations to the 5620 SAM object model 2-13
2.9 Upgrade implications 2-15

5620 SAM-O communications

3 — Communication with the 5620 SAM server 3-1

3.1 Overview 3-2
3.2 5620 SAM system components 3-2
3.3 Client interfaces 3-3
Event monitoring using JMS 3-4
Request and response using the 5620 SAM-O 3-4
XML and SOAP 3-4
3.4 Secure communication 3-5
SSL message encryption 3-5
HTTPS communication with 5620 SAM-O 3-6
Session management 3-6
Procedure 3-1 To generate an MD5-hashed password for 5620 SAM main server access 3-7
5620 SAM-O security features controlled through the 5620 SAM client GUI 3-8
3.5 Workflow to set up and operate 5620 SAM-O 3-10

4 — Event monitoring using JMS 4-1

4.1 Overview 4-2
4.2 JMS connections 4-2
4.3 JMS subscriptions 4-3
Client ID 4-3
JMS topics 4-3
Message filters 4-4
Acknowledgement modes 4-6
Persistence modes 4-7
4.4 JMS Events 4-7
JMS XML schemas 4-7
JMS event header properties 4-8
JMS event classes 4-10
4.5 JMS and redundancy 4-23

4.6 Connection monitoring and error recovery 4-24
Procedure 4-1 To resync an OSS database with the 5620 SAM 4-25
JMS exception 4-26
Monitoring for incoming events 4-26
5620 SAM-O session termination 4-27
Missed events 4-27
4.7 Managing and monitoring JMS sessions 4-29
Monitoring sessions 4-29
JMS logging 4-30
Procedure 4-2 To manage a 5620 SAM-O JMS client session or remove a durable subscription 4-30
4.8 JMS consumer creation 4-31
Procedure 4-3 To compile and create a sample JMS client *.java file 4-31
4.9 Workflow to set up and operate a JMS connection 4-33
4.10 JMS record and playback tool 4-34
Procedure 4-4 To record, stop and playback JMS messages 4-34

5 — XML requests 5-1

5.1 XML communication with the 5620 SAM server 5-2
PostXML 5-2
Procedure 5-1 To post an XML request to the 5620 SAM server 5-4
5.2 Monitoring the status of the 5620 SAM-O connection using XML API ping 5-5
XML API ping 5-5
5.3 Workflow to set up and operate an HTTP XML request-response connection 5-7
5.4 Viewing XML requests ignored by the server 5-7
Procedure 5-2 To enable server log file logging of ignored XML requests 5-8
5.5 5620 SAM-O server-specific commands 5-8
Obtaining the local server time for a request 5-8
Identifying the server software load 5-9

5620 SAM-O information model

6 — 5620 SAM-O information model overview 6-1

6.1 5620 SAM-O information model overview 6-2
6.2 5620 SAM-O documentation 6-3

7 — 5620 SAM-O reference 7-1

7.1 5620 SAM-O XML reference overview 7-2
Procedure 7-1 To access the 5620 SAM-O XML Reference 7-4
Procedure 7-2 To view information about schema changes 7-4
Procedure 7-3 To view information about general methods and types 7-4
Procedure 7-4 To view deprecated methods and properties 7-5

Procedure 7-5 To view supported devices 7-5
Procedure 7-6 To view JMS changes 7-5

8 — 5620 SAM-O XML packages 8-1
8.1 5620 SAM-O XML packages overview 8-2
Procedure 8-1 To view package information 8-2

9 — 5620 SAM-O XML schemas 9-1
9.1 5620 SAM-O XML schemas 9-2
Common elements of methods 9-6
Common elements of classes (types) 9-7

10 — XML message structure 10-1
10.1 XML message structure overview 10-2
Message structure 10-2
Requests 10-4
Responses 10-10
Faults and exceptions 10-11
Log file for XML requests, responses, and exceptions 10-15
10.2 CLI commands within XML methods 10-17
10.3 Mapping XML methods to GUI operations 10-18
Procedure 10-1 To enable logging of GUI operations on the 5620 SAM server 10-18
Procedure 10-2 To view the GUI operation in the server log 10-19
Output format of a GUI operation in the server log file 10-19
10.4 XML samples 10-20

5620 SAM-O common tools and methods

11 — 5620 SAM-O core methods 11-1
11.1 Core methods overview 11-2

12 — Script management 12-1
12.1 Script management 12-2
12.2 Workflow to execute a script 12-2
12.3 Sample script management requests 12-3

OSS domains

13 — Fault management 13-1

13.1 Fault management overview 13-2
Alarm management requests 13-4
13.2 Workflow to configure alarm and event management 13-5
13.3 OAM overview 13-8
Test policies 13-8
Test suites 13-9

14 — Inventory management 14-1

14.1 Inventory retrieval 14-2
<find> method 14-4
<findToFile> method 14-4
Procedure 14-1 To configure permission for the <findToFile> directory 14-9
14.2 5620 SAM object hierarchy 14-10
Physical object hierarchy 14-10
Logical object hierarchy 14-11
14.3 Filters 14-20
Children filters 14-20
Filter element types 14-21

15 — Accounting and performance monitoring 15-1

15.1 Accounting and performance monitoring overview 15-2
15.2 Statistic objects 15-2
accounting package 15-3
file package 15-3
log package 15-3
root package 15-3
15.3 Workflow to retrieve statistics 15-3
15.4 JMS XML statistics topic 15-4
15.5 Scheduled and on-demand statistics 15-5
15.6 Statistics retrieval process 15-6
Accounting statistics retrieval using registerLogToFile 15-6
Accounting and performance statistics retrieval using findToFile 15-11
Recovery of accounting statistics where OSS is using registerLogToFile 15-14
Recovery of accounting and performance statistics where OSS is using findToFile 15-14
XML statistics output file 15-15
JMS performance statistics 15-16
15.7 Sample request to retrieve historical statistics from a router 15-17
15.8 Sample request to collect interface statistics 15-17

Configuration management

16 — Configuration management overview 16-1

16.1 Configuration management overview 16-2
16.2 Generic methods 16-2
Sample request to create a channel on a SONET port 16-3
Sample request to configure a SONET port 16-4
Sample request to create a network interface 16-4
Sample request to remove a daughter card slot 16-5
16.3 Deployers 16-5
Deployment types <deployer> 16-6
Synchronous and asynchronous requests 16-6
Deployer failures 16-8
Deployer alarms 16-11
Deployer simulation 16-12
16.4 Workflow to handle deployer failures 16-13

17 — Equipment configuration management 17-1

17.1 Equipment configuration overview 17-2
17.2 Workflow to configure equipment 17-2
17.3 Workflow to configure a router 17-3
17.4 Generic NE profiles 17-3
17.5 Workflow to create a generic NE profile 17-3
17.6 Device software upgrades 17-4
Procedure 17-1 To configure the nodeSoftware timeToKeepFile property 17-5

18 — Routing protocol configuration management 18-1

18.1 Routing protocol configuration management overview 18-2
BGP 18-2
MP-BGP 18-3
RIP 18-3
LDP 18-3
IS-IS 18-3
PIM 18-4
MSDP 18-4
IGMP 18-4
18.2 Workflow to configure a routing protocol 18-4
18.3 Workflow to configure a PIM or IGMP multicast protocol 18-7
18.4 VRRP virtual router configuration 18-7
18.5 Workflow to configure a virtual router 18-8

19 — Customer and residential subscriber configuration management 19-1

19.1 Customer and residential subscriber configuration overview 19-2
Customer 19-2
Residential subscriber 19-2

19.2 Workflow to configure and manage customers 19-3
19.3 Workflow to configure and manage residential subscribers 19-3

20 — Policy configuration management 20-1

20.1 Policy configuration overview 20-2
Service management policies 20-2
Routing management policies 20-4
Network management policies 20-4
Policy distribution mode 20-5
Policy configuration mode 20-6
20.2 Workflow to configure ACL IP and MAC filter policies 20-7
20.3 Workflow to configure routing management policies 20-7

21 — Service configuration management 21-1

21.1 Service configuration overview 21-2
VLL 21-2
VPLS 21-3
IES 21-4
VPRN 21-5
VLAN 21-6
Mirror 21-7
Optical transport service 21-9
Object life cycle 21-10
21.2 Workflow to configure a VLL, VPLS, IES, VPRN, or VLAN service 21-10
21.3 Workflow to configure a mirror service 21-12
21.4 Workflow to configure an optical transport service 21-12
21.5 Composite services 21-13
Network discovery of composite services 21-14
Connector types 21-14
Sample composite service configuration 21-16
21.6 Workflow to configure a composite service 21-17
21.7 IGMP snooping 21-18
21.8 Workflow to configure IGMP snooping 21-18
21.9 DHCP relay configuration 21-18
21.10 Workflow to configure DHCP relay 21-19
21.11 IPsec management 21-19
Typical IPsec configuration example 21-19
5620 SAM-O IPsec configuration 21-20
OSSI configuration requirements for static tunnel type 21-20
OSSI requirements for dynamic tunnel type 21-20
21.12 Workflow to configure IPsec — option A 21-20
21.13 Workflow to configure IPsec — option B 21-21

22 — XML API service template configuration 22-1

22.1 XML API service template overview 22-2
XML API template script format 22-3
22.2 Workflow to manage XML API service templates 22-4

Integrated products

23 — 5650 CPAM integration configuration 23-1

23.1 5650 CPAM integration overview 23-2
23.2 5650 CPAM OSS interface overview 23-4
Multiple topologies 23-4
5650 CPAM administrative domains 23-4
Topology checkpoints 23-5
Route management 23-8
Path monitoring 23-10
Fault management 23-10
23.3 5650 CPAM OSS packages 23-11

Appendices

A. Troubleshooting A-1

A.1 Troubleshooting client OSS application problems A-2
Procedure A-1 The client OSS application cannot communicate with the server A-2
Procedure A-2 An attempt to log in to the 5620 SAM server fails A-3
Procedure A-3 Receive insufficient privileges to perform this operation on an object exception when performing an action on a 5620 SAM object A-4
Procedure A-4 An XML request fails A-4
Procedure A-5 Unable to perform an action using the 5620 SAM-O A-4
Procedure A-6 Identifying XML messages from specific users A-5
Procedure A-7 Receive a java.lang.UnsupportedClassVersionError when sending scripts using the PostXML tool A-7
Procedure A-8 Receive a java.net.ConnectException when sending scripts using the PostXML tool A-7
Procedure A-9 Receive a java.net.ConnectException when sending HTTP requests to the 5620 SAM-O server A-8
Procedure A-10 The OSS client cannot connect with the HTTP or JMS server A-9
Procedure A-11 The OSS client cannot perform find or findTofile requests A-9
Procedure A-12 The accounting statistics file has been removed from its location on the 5620 SAM server A-10
Procedure A-13 The JMS client that collects statistics automatically unsubscribes and deregisters from the 5620 SAM server A-10

B. Accounting and performance B-1

B.1 Description of sample values for performance statistical counters B-2

C. 5620 SAM-O SDK library of samples C-1
C.1 Description 5620 SAM-O SDK library of XML samples C-2
Basic SDK C-2
Advanced SDK C-2

Getting started

1 — Product and support overview

2 — 5620 SAM-O release features

1 — Product and support overview

1.1 5620 SAM product suite *1-2*

1.2 5620 SAM-O overview *1-3*

1.3 Supported technologies and configuration *1-5*

1.4 Supported devices *1-6*

1.5 Before you begin *1-7*

1.6 Open interfaces professional support *1-8*

1.1 5620 SAM product suite

The 5620 SAM is a network and element management system that is used to manage devices, such as the 7750 SR, 7710 SR, and 7450 ESS, by simplifying network and element management tasks. The 5620 SAM also provides limited management functions for non-Alcatel-Lucent NEs that support SNMPv1 and later.

The 5620 SAM supports IP/MPLS-based network convergence, and provides subscriber services such as VLL, VPLS, VPRN, IES, and VLAN over a common network infrastructure. The 5620 SAM allows network operators to manage their networks at the service, customer, and subscriber levels.

The 5620 SAM software architecture is built on industry standards including open standards such as SOAP and XML, the Java and J2EE framework, multi-tier layering, and web service interfaces. This use of standard interfaces allows the 5620 SAM to integrate with other network management systems such as the 5620 NM and the 5750 SSC, to add management of IP/MPLS and metro Ethernet services to multiservice networks.

The 5620 SAM supports the configuration of IPv6 addresses for some routing-protocol functions, such as BGP peering, in the base routing instance of the 7750 SR, Release 4.0 and later. See the *5620 SAM User Guide* for more information.

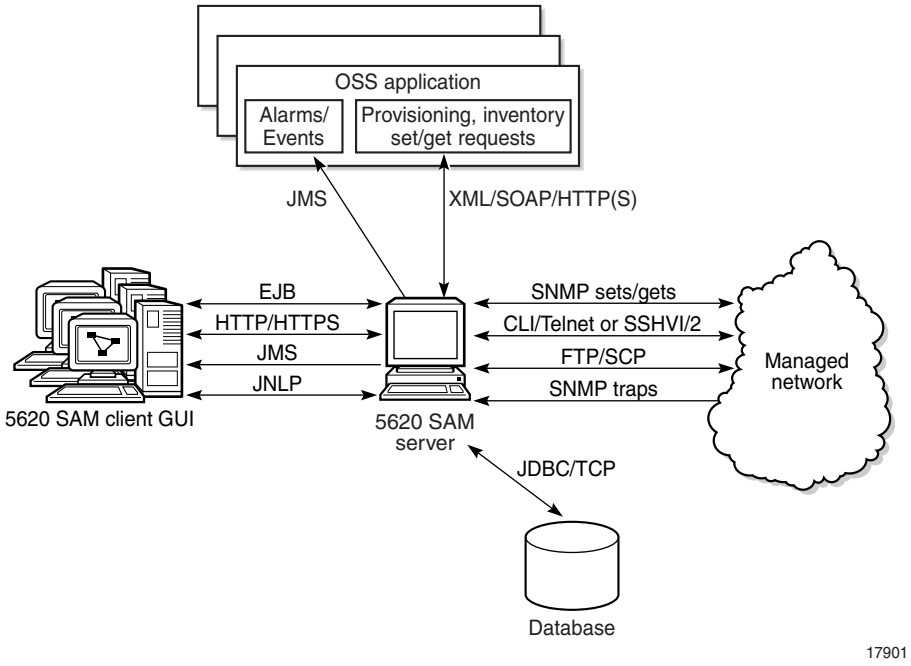
The distributed 5620 SAM elements, collectively known as the 5620 SAM portfolio, run on independent PCs or workstations. Table 1-1 list the components of the 5620 SAM product suite.

Table 1-1 5620 SAM product suite components

Component		Function
server		Network management processing engine written in Java for Sun Solaris or Microsoft Windows platforms. The server includes several third-party components, such as an application server, web server, protocol stack set, and database adaptor.
database		Customized Oracle relational database that provides persistent storage for the network data. The database usually runs on the same platform as the 5620 SAM server software.
clients	GUI	Java-based GUI that runs on Sun Solaris or Microsoft Windows.
	OSS	OSS applications that exchange platform-neutral XML/SOAP messages with the 5620 SAM server.

Figure 1-1 shows the architecture of the 5620 SAM product suite.

Figure 1-1 5620 SAM product suite architecture



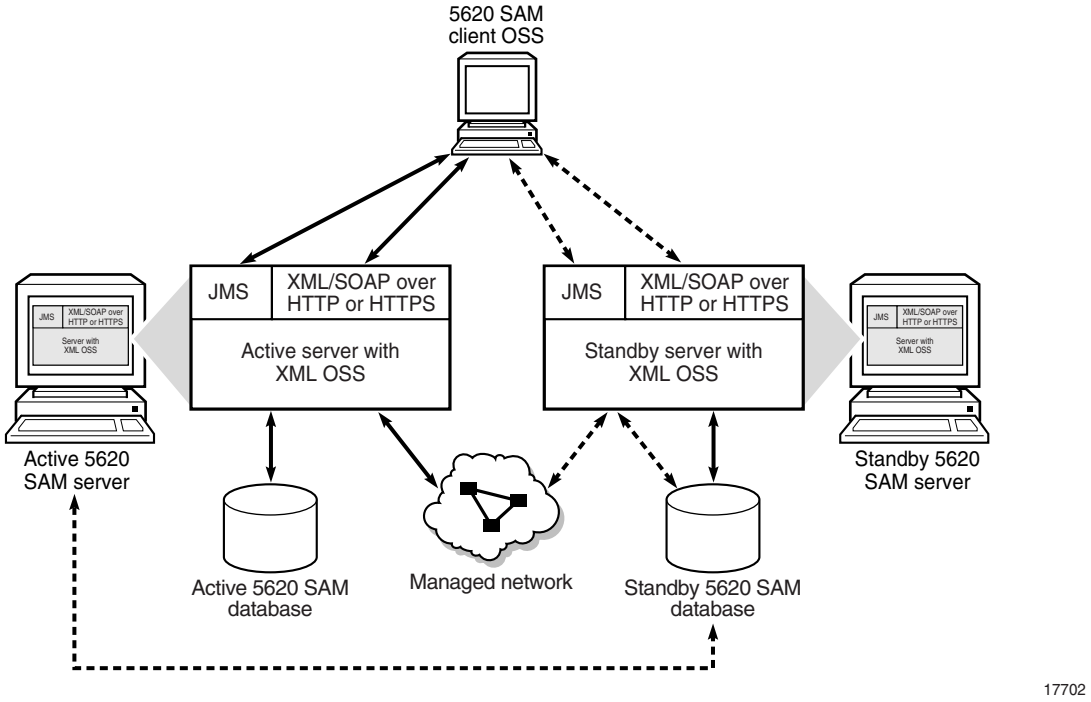
1.2 5620 SAM-O overview

The 5620 SAM-O is an open interface on the 5620 SAM through which an OSS client application can perform the following tasks:

- configure or access network management information in the 5620 SAM database
- send requests and receive responses between the 5620 SAM-O server and the OSS client
- receive information about managed objects
- manipulate managed objects
- receive event information from the 5620 SAM server using a persistent or non-persistent JMS connection

Figure 1-2 shows the architecture of the 5620 SAM-O, which includes a JMS connection for the collection of near-real-time alarms and events.

Figure 1-2 5620 SAM-O architecture



The 5620 SAM-O client sends an XML-encoded, SOAP-enabled message to a port on the HTTP(S) server that runs on the 5620 SAM server. The default port for incoming messages is 8080. An XML application at port 8080 parses the SOAP XML message and sends the request to the 5620 SAM.

Warning 1 — The 5620 SAM-O has a limit of 10 concurrent and active HTTP connections.

Warning 2 — A limit of 10 concurrent XML/SOAP requests can be run over the 10 HTTP connections.

Alcatel-Lucent strongly recommends that you use only one HTTP connection for your XML/SOAP requests.

If more than one HTTP connection is required, for example, for performance and scalability, you must make the number of HTTP connections configurable at run time (after installation), so that system integrators can manage the HTTP connections across the OSS applications that connect to the same 5620 SAM server. Contact your Alcatel-Lucent OIPS technical support representative for more information.

The 5620 SAM-O interface is installed with the core 5620 SAM software components. To use the 5620 SAM-O interface, you must perform the following configuration tasks.

- Enter a 5620 SAM license key that activates the OSS functionality.
- Use a user group that is assigned the OSS management scope of command role.

Note — You can create a scope of command profile to group the OSS management scope of command role with additional roles. See the *5620 SAM User Guide* for more information about 5620 SAM user groups and the associated scope of command roles and permissions.

5620 SAM-O client

The 5620 SAM-O client implementation can range from a CLI to a third-party application. The format for the client user interface varies with the function of the client application; for example, rolling up statistics into a third-party billing application. The 5620 SAM-O clients function the same way regardless of the front-end implementation: XML scripts are posted to the 5620 SAM server using an HTTP client. See section 3.1 for more information.

1.3 Supported technologies and configuration

The 5620 SAM-O supports the following technologies:

- Java 1.6
- JMS 1.1
- SOAP 1.2

Redundant network configurations

The 5620 SAM server manages all transactions in the 5620 SAM database. The 5620 SAM supports database and server redundancy, which provides network visibility if hardware or software fails on one or more 5620 SAM components; for example, a server. If the active component fails, the standby component is configured to automatically take control. You can also perform a manual switch to test redundancy or to prepare for a network upgrade.

The 5620 SAM product suite supports collocated and distributed redundant configurations. In a collocated installation, the 5620 SAM server and database are on the same workstation. In a distributed installation, the 5620 SAM server and the 5620 SAM database are on separate workstations. The servers and databases are four separate entities regardless of the physical or geographical deployment.

Note — Redundant network configuration is only supported on Solaris workstations.

You must know the following information to configure redundant 5620 SAM connections:

- the IP address of the primary and secondary servers
- which server is currently active

Redundancy is configured during the installation of each 5620 SAM application. See the *5620 SAM | 5650 CPAM Installation and Upgrade Guide* for more information about configuring 5620 SAM applications for redundancy.

You can use any of the following methods to monitor the connection status between the 5620 SAM OSS client and the active server:

- JMS exception to monitor client connections to the JMS server
- monitoring near-real-time events using JMS client connections to the JMS server
- XML API ping to monitor the HTTP connection to the 5620 SAM web server

See section 3.3 for more information about monitoring the connection status between the 5620 SAM OSS client and the active server.

See "JMS and redundancy" in chapter 4 for more information about JMS and redundancy.

1.4 Supported devices

The 5620 SAM supports the following devices:

- 7750 SR
- 7710 SR
- 7705 SAR
- 7450 ESS
- 7210 SAS-M and 7210 SAS-E
- 7250 SAS, 7250 SAS-ES, and 7250 SAS-ESA
- Telco
- OS 6850
- OS 6855
- OS 6400
- 9500 MPR
- generic NEs

The 5620 SAM supports up to three different major releases of the devices. For example, the 5620 SAM can manage a 7750 SR at Release 7.0, Release 6.1, Release 6.0, or Release 5.0. Older device releases are not supported.

Note — Releases 6.0 and 6.1 are part of the same major release.

The 5620 SAM supports the minor releases of a device when the major release is supported.

Note — There may be restrictions if a device does not follow the basic rules for MIB changes. When a device introduces maintenance releases, you may need to upgrade the 5620 SAM to a release that supports the changes on the device.

The following documents describe the compatibility and functionality for the devices managed by the 5620 SAM.

- See the *5620 SAM NE Compatibility Guide* for information on compatible and supported device releases.
- See the 7750 SR, 7710 SR, 7705 SAR, 7450 ESS, 7250 SAS, and Telco user documentation for information about device-specific CLI commands, parameters, and installation. Contact your Alcatel-Lucent support representative for information about specific network or facility considerations.

Note — The 5620 SAM functions and features vary for the supported devices. The *5620 SAM-O OSS Interface Developer Guide* does not describe the major variations in support, or device-specific procedures for specific implementations. See the *5620 SAM User Guide*.

The *5620 SAM Parameter Guide* describes device-specific support for parameters and variations in default values, when applicable.

1.5 Before you begin

Alcatel-Lucent recommends that you review the following information before you start to work in the 5620 SAM-O XML development environment.

- Review the requirements for license key configuration and platform size considerations. See the *5620 SAM | 5650 CPAM Installation and Upgrade Guide* for more information.
- Preconfigure the routers that you want the 5620 SAM to manage. See the *5620 SAM User Guide* and *5620 SAM Parameter Guide* for more information.
- Backup the 5620 SAM database. Alcatel-Lucent recommends a database backup before you implement any major, network-wide configuration changes. A database backup utility is available from the 5620 SAM client GUI. See the *5620 SAM User Guide* and *5620 SAM Parameter Guide* for more information.
- Understand the following technology and network management concepts:
  - OSS interfaces and standard XML schema constructions
  - Managed devices and 5620 SAM functionality. See the 5620 SAM documentation listed in the “Preface”
- Contact your Alcatel-Lucent OIPS representative for development support. See section 1.6 for more information.

1.6 Open interfaces professional support

Alcatel-Lucent provides developer support for third-party application integration with the 5620 SAM-O. The Alcatel-Lucent OIPS portfolio provides OSS developers with network management integration solutions for the 5620 SAM and 5650 CPAM. OSS integration initiatives include project review, design consultation, development support, and training for integration projects.

Key elements of each service offering include:

- Design consultation and high-level project reviews. Extensive software design consultation provides architectural review of a proposed design and guidance on whether the design methodology is suited to the proposed application given Alcatel-Lucent experience with the platform and interface.
- Software development support and technical guidance
- Sample code that provide examples of configuration and management concepts, to facilitate rapid OSS integration
- OSS product training that minimizes project down time and optimizes development efficiency
- OSS Product Certification support to ISVs identified as Connected Partners, involving extensive interoperability testing of OSS applications with the 5620 SAM and 5650 CPAM.

Alcatel-Lucent OIPS can customize the support content and contract length to meet the unique development requirements of service providers.

2 — 5620 SAM-O release features

2.1 5620 SAM-O Release 9.0 functionality *2-2*

2.2 5620 SAM-O Release 8.0 functionality *2-2*

2.3 5620 SAM-O Release 7.0 functionality *2-4*

2.4 5620 SAM-O Release 6.1 functionality *2-5*

2.5 5620 SAM-O Release 6.0 functionality *2-5*

2.6 5620 SAM-O Release 5.0 functionality *2-7*

2.7 5620 SAM-O Release 4.0 functionality *2-11*

2.8 Understanding changes, modifications, and deprecations to the 5620 SAM object model *2-13*

2.9 Upgrade implications *2-15*

2.1 5620 SAM-O Release 9.0 functionality

Table 2-1 lists the 5620 SAM-O release features and documentation changes for 5620 SAM Release 9.0.

Table 2-1 5620 SAM-O Release 9.0 features, problem fixes, and documentation changes

Feature or function	Description	Reference for more information
Release 9.0 R1 features		
Optical transport service support	The 5620 SAM supports optical transport service configuration. The API is new and evolving, and is subject to change in future releases.	See "Optical transport service" in chapter 21 for more information.
Release 9.0 R1 problem fixes		
—	—	—
Release 9.0 R1 documentation content enhancements		
Generic methods	Added information about the following: Generic methods for error recovery includes the generic.GenericObject.getDeployersShallow method.	See section 16.3 for more information about deployers.
Filter construction	Added information about the following: Filter construction requires that you know the class to which you want to apply the filter. The calls can be defined by the class attribute in the top-level <filter> element.	See section 14.3 for more information about filters.
Methods for configuring objects	Added references to generic methods and object hierarchy.	—
Output format of a GUI operation in the server log file	Updated the subsection.	See section 10.3 for more information.
Exceeding find or findtoFile request limit	Added information about the following: If 5 concurrent <find> or <findToFile> requests is reached, subsequent <find> or <findToFile> requests are rejected with an XMLException or IMException response respectively.	See section 14.1 for more information.
Release 9.0 R1 documentation structure changes		
—	—	—

2.2 5620 SAM-O Release 8.0 functionality

Table 2-2 lists the 5620 SAM-O release features and documentation changes for 5620 SAM Release 8.0.

Table 2-2 5620 SAM-O Release 8.0 features, problem fixes, and documentation changes

Feature or function	Description	Reference for more information
Release 8.0 R5 documentation content enhancements		
SDK library	The structure of the SDK library appendix has been updated.	See Appendix C for more information.
Troubleshooting	The troubleshooting information has been enhanced to include the following: • The OSS client cannot perform find or findTofile requests • The accounting statistics file has been removed from its location on the 5620 SAM server • The JMS client that collects statistics automatically unsubscribes and deregisters from the 5620 SAM server	See Appendix A for more information.
Release 8.0 R3 documentation content enhancements		
JMS events	Added sample output for IncrementalRequestEvent.	See section 4.4 for more information.
5620 SAM object model hierarchy	The parent-child relationships in the object hierarchy are captured by the Children-Set properties that are documented in the 5620 SAM-O XML Reference.	See section 14.2 for more information.
Client interfaces	Added overview information about: • event monitoring using JMS • request and response using the 5620 SAM-O • XML and SOAP	See section 3.3 for more information.
PostXML	Added clarification about PostXML.	See section 5.1 for more information.
Leaf filter element types	Added examples of the leaf filter elements.	See Table 14-6 for more information.
XML schema	Removed information about netw.Network.findNe because it is deprecated.	See section 14.3 for more information.
Release 8.0 R1 features		
OSS user and group priority	Users and groups can be assigned priorities that determine which user sessions take priority over others when system resources are limited.	See "Session management" in chapter 3 for more information.
Maximum number of OSS connections per user	The system administrator can specify the maximum number of active user connections per user or user group.	See "Session management" in chapter 3 for more information.
Span of control enhancements on JMS and HTTP	You must create a scope of command profile to group the OSS Management scope of command role with additional roles. If a 5620 SAM-O user is not assigned the appropriate scope of command privilege for a class type, or the appropriate span of control for an object, exceptions are returned. Unlike in HTTP or HTTPS where span of control rules are implicitly enforced, span of control rules are not implicitly enforced via JMS. JMS requires the administrator to explicitly specify the required span IDs within in the JMS filter.	See "5620 SAM-O security features controlled through the 5620 SAM client GUI" in chapter 3 for more information.
JmsMissedEvents enhancements	JmsMissedEvents apply now to both durable and non-durable subscriptions.	See chapter 4 for more information.

(1 of 2)

Feature or function	Description	Reference for more information
KeepAliveEvent change	A KeepAliveEvent is received when you are subscribed to any of the JMS topics, not just to the topic-xml.	See "Monitoring for incoming events" in chapter 4 for more information.
Maximum HTTP connection limit change	The maximum number of concurrent OSS connections is 10 or more and varies based on server hardware.	See the *5620 SAM Planning Guide* for information about platform sizing.
<findToFile> concurrent requests limit change	The maximum number of concurrent findToFile/find requests is limited to five.	See "Synchronous and asynchronous requests" in chapter 14 for more information.
Release 8.0 R1 problem fixes		
—	—	—
Release 8.0 R1 documentation content enhancements		
JMS record and playback tool	Information added about the JMS record and playback tool.	See section 4.10 for more information.
LogFile recovery	Information added about recovery statistics where OSS is using registerLogToFile and findToFile.	See section 15.6 for more information.
Release 8.0 R1 documentation structure changes		
—	—	—

(2 of 2)

2.3 5620 SAM-O Release 7.0 functionality

Table 2-3 lists the 5620 SAM-O release features and documentation changes for 5620 SAM Release 7.0.

Table 2-3 5620 SAM-O Release 7.0 features, problem fixes, and documentation changes

Feature or function	Description	Reference for more information
Release 7.0 R3 features		
samOss.jar	You can download the samOss.jar file from the following location: http://<*ip_address_SAM_server*>:8080/integration.samOss.jar	See section 4.8 in chapter 4 for more information.
Release 7.0 R1 features		
Alarm details	The alarmDetails.xml file has been added to provide OSS users with alarm descriptions.	See chapter 13 for more information about alarms.
Release 7.0 R1 problem fixes		
JMS subscription best practices	The connection monitoring and error recovery procedure has been updated.	See section 4.6 in chapter 4.
TerminateClientSession	TerminateClientSession is no longer a broadcast message.	See section JMS event classes in chapter 4.

(1 of 2)

Feature or function	Description	Reference for more information
Release 7.0 R1 documentation content enhancements		
Deprecated jars	The following jars are deprecated and will be removed in a future release. samOss.jar should be used to connect to the 5620 SAM server: • samOssJBoss.jar • samOssAnyServer.jar	See section 4.8 in chapter 4 for more information.
Release 7.0 R1 documentation structure changes		
5650 CPAM integration chapter	—	See chapter 23 for more information about the 5650 CPAM-O.

(2 of 2)

2.4 5620 SAM-O Release 6.1 functionality

Table 2-4 lists the 5620 SAM-O release features and documentation changes for 5620 SAM Release 6.1.

Table 2-4 5620 SAM-O Release 6.1 features, problem fixes, and documentation changes

Feature or function	Description	Reference for more information
Release 6.1 R1 features		
JMS architecture change for large latent networks	Clients can specify an acknowledgement mode for JMS sessions. The DUPS_OK_ACKNOWLEDGE acknowledgement mode is now supported.	See chapter 4 for more information.
Release 6.1 R1 problem fixes		
—	—	—
Release 6.1 R1 documentation structure changes		
—	—	—

2.5 5620 SAM-O Release 6.0 functionality

Table 2-6 lists the 5620 SAM-O release features and documentation changes for 5620 SAM Release 6.0.

Table 2-5 5620 SAM-O Release 6.0 features, problem fixes, and documentation changes

Feature or function	Description	Reference for more information
Release 6.0 R3 features		
—	—	—

(1 of 2)

Feature or function	Description	Reference for more information
Release 6.0 R3 problem fixes		
Scheduled data export	Sample code in Table 14-2 is updated.	See chapter 14 for more information.
Release 6.0 R3 documentation structure changes		
Reorganization of existing content	New chapter on XML API configuration templates.	See chapter 22 for more information.
	New chapter on the core methods that a 5620 SAM-O client can use to perform specific actions.	See chapter 11 for more information.
Release 6.0 R1 features		
JBoss AS 4.2 infrastructure	JBoss 4.2 uses JBoss Messaging. JBoss MQ is no longer used.	—
New log directory structure	The following changes have been made to the log directory structure: • log/client The client log file EmsClientLog.txt is now called EmsClient.log • log/server The server log file EmsServerLog.txt is now called EmsServer.log • log/jmsserver The server log file JmsServerLog.txt is now called JmsServer.log • The server log file AuxServerLog.txt is now called AuxServer.log	—
Durable subscription message limit	The default value of 100 000 JMS messages applies to each durable subscription. In previous releases, this limit was shared among all durable subscribers. Durable clients are no longer disconnected when message limits are exceeded.	See chapter 13 for more information.
Release 6.0 R1 problem fixes		
—	—	—
Release 6.0 R1 documentation content enhancements		
Java library files for PostXML	The commons-codec.jar Java library file has been added to the PostXML file requirements.	See chapter 3 for more information.
Raise visibility of location of samOss.jar	The samOss.jar file is located in /integration/SAM_O/samOss.jar on the 5620 SAM installation DVD-ROM. The samOss.jar is also installed on the 5620 SAM server in the /opt/5620sam/server/nms/integration/SAM_O directory.	See chapter 13 for more information.
Updated samples	JMS filter samples have been updated.	See chapter 13 for more information.
Release 6.0 documentation structure changes		
—	—	—

(2 of 2)

2.6 5620 SAM-O Release 5.0 functionality

Table 2-6 lists the 5620 SAM-O release features and documentation changes for 5620 SAM Release 5.0.

Table 2-6 5620 SAM-O Release 5.0 features, problem fixes, and documentation changes

Feature or function	Description	Reference for more information
Release 5.0 R1 features		
Action on Failure	The option <continueOnFailure> is supported in the 5620 SAM Release 5.0. If the <continueOnFailure> option is set to true, the execution of requests continues even during a failure. Responses and exceptions are returned for each success or failure, respectively. The options <execute> and <on Failure> are supported in the 5620 SAM Release 5.0, to allow for request rollbacks. The <timeout> option is supported in the 5620 SAM Release 5.0. If a request takes longer than the value specified by the <timeout> option to execute, the request is terminated and an exception is returned.	See chapter 6 for more information.
New jar files	The requirement for the samOss.jar file has been added to the workflow to configure near-real-time events and alarms using JMS. In addition, the following two jars have been created and can be used in the cases described in chapter 13: • samOssJBoss.jar • samOssAnyServer.jar	See chapter 13 for more information.
New method to close JMS sessions	The stopOssSession method can be used to close a JMS session.	See chapter 13 for more information about JMS connections.
New JMS event classes	The StatsEvent identifies the type of statistics (polling or accounting), the collection time, and the network element for which the statistics are collected. The ExceptionEvent reports policy distribution errors.	See chapter 13 for more information.
New option for the <find> and <findToFile> methods	The <timeout> option specifies the time, in milliseconds, after which the <findToFile> and <find> operations time out.	See chapter 14 for more information.
New option for the <findToFile> method	The <timestamp> option includes the timestamp in the filename of the saved file.	See chapter 14 for more information.
Scheduled <findToFile>	The <findToFile> method supports the scheduled export of data using the existing 5620 SAM scheduler framework.	See chapter 14 for more information.
Release 5.0 R3 features		
No new 5620 SAM-O specific features		
Release 5.0 R5 features		
No new 5620 SAM-O specific features		
Release 5.0 R6 features		
No new 5620 SAM-O specific features		
Release 5.0 R7 features		

(1 of 5)

Feature or function	Description	Reference for more information
New method for statistics retrieval	The <registerLogToFile> method can be used to create up-to-date export files with accounting statistics for NEs.	See chapter 15 for more information.
Obsolete class for accounting statistics	The log.CurrentData class and its children classes are obsolete for accounting statistics. The classes that inherit from log.LogRecord must be used. The log.CurrentData class continues to be used for performance statistics.	—
Release 5.0 R1 problem fixes		
Limit on HTTP connections	Raise visibility of the limit of 10 concurrent and active HTTP connections in the 5620 SAM-O.	See chapter 1 for more information.
Limit on XML/SOAP requests	Raise visibility of the limit of 10 concurrent XML/SOAP requests that can be run over the 10 HTTP connections, and the recommendation that you use only one HTTP connection for XML/SOAP requests.	See chapter 1 for more information.
Limit on concurrent JMS connections	Raise visibility of the limit of 10 concurrent JMS connections from OSS applications in the 5620 SAM.	See chapter 13 for more information.
Release 5.0 R3 problem fixes		
samOss.jar	Raise visibility of location of samOss.jar.	See chapter 13 for more information.
Release 5.0 R5 problem fixes		
Maximum limits for the <findToFile> method	Removed configuration information related to xmlFindPoolSize and xmlFindQueueSize methods. The following maximum limits apply to the use of the <findToFile> method: • 5 concurrent <findToFile> methods • 10 <findToFile> methods queued beyond the maximum number of concurrent methods	See Synchronous and asynchronous requests in section 14.1 for more information.
OAM service validation	OAM validation tests to verify the operational status of a service or service tunnel	See section OAM service validation in section 13.3 for more information.
Release 5.0 documentation content enhancements		
Reorganization of existing content	Structure optimized for the presentation of information on synchronous and asynchronous requests.	See chapter 14 for more information.
	Configuration chapter restructured to optimize the presentation of workflows to configure the various functional areas of the 5620 SAM.	See chapter 16 for more information.
	Structure optimized for the presentation of information on synchronous and asynchronous deployments. New section on deployer failures. Recommendations on the clearing of failed deployments to minimize resource use and the blocking of synchronous OSS requests.	See chapter 16 for more information.
Upgrade implications	When you perform an upgrade of the 5620 SAM, FDNs may change. 5620 SAM-O clients using FDNs to reference objects in the 5620 SAM must update their FDNs appropriately.	See section Upgrade implications in this chapter for more information.

(2 of 5)

Feature or function	Description	Reference for more information
Monitoring events using JMS	Method to monitor the connection status between the 5620 SAM OSS client and the active server and recommendation for monitoring events.	See chapter 3 for more information about monitoring the 5620 SAM OSS client and server connection status.
JMS client ID format	The JMS client ID format *<unique_name>@<numeric_id>@<client_ipaddress>* for JMS connections has been deprecated.	See chapter 4 for more information about JMS client ID formats.
HA (High Availability) Port for JMS connections	To ensure that the 5620 SAM server reconnects after an activity switch, you must verify that the HA Port for JMS connections is set to the JBoss High Availability JNDI service port value.	See chapter 4 for more information.
<findToFile> synchronous requests	The <findToFile> method can be run synchronously. When synchronous requests are used, the request blocks further requests until the <findToFile> request is complete.	See chapter 14 for more information about the <findToFile> method.
<find> and <findToFile> input parameters	Strengthened recommendation to use the <filter> input parameter to filter on properties of the class corresponding to fullClassName when using the <find> and <findToFile> methods. Strengthened recommendation to use the <resultFilter> input parameter for narrowing down the information returned per object when using the <find> and <findToFile> methods.	See chapter 14 for more information about the <find> and <findToFile> methods.
Updated equipment containment hierarchy	The containment hierarchy tree that is associated with equipment types has been updated.	See chapter 14 for more information about the physical object hierarchy.
Updated logical object hierarchy	The hierarchy of logical objects, such as services, and the children objects that belong to them has been updated.	See chapter 14 for more information about the logical object hierarchy.
JMS XML statistics topic	Raise visibility of the publishing of polled statistics measurements to OSS clients using JMS on the XML statistics topic.	See chapter 15 for more information about accounting and performance statistics.
Security enhancements	Raise visibility of the configuration of SSL for secure JMS-XML message encryption.	See chapter 3 for more information about SSL.

(3 of 5)

Feature or function	Description	Reference for more information
Updated samples	The following samples have been added or updated: • Sample XML schema definition for a class • Sample XML schema definition for an element • Sample XML schema definition for a method • Sample request to create a VPLS site • Sample request after a failure during VPLS site creation • Sample request with execute and onFailure blocks • Sample exception message for failed router configuration • Sample request from the ossiuserRequest1.log XML message log file • Sample response from the ossiuserResponse1.log XML message log file • Sample timestamp and server-specific version request • Sample response to timestamp and server-specific version request • Sample request to configure the equipment.LinkDown alarm policy • Sample DeployerEvent output • Sample StatsEvent output • Sample filter example - composite and leaf filters used together for a top level object and children	See chapters 5, 3, 4, and 14 for more information.
Release 5.0 R3 documentation content enhancements		
5620 SAM-O SDK library of XML sample requests	Appendix C describes the XML configuration samples that are available in the 5620 SAM-O SDK library of XML sample requests.	See Appendix C for more information.
Faults and exceptions	Updated section Faults and exceptions with sample faults and exceptions that can appear in an XML response.	See chapter 10 for more information.
Release 5.0 R5 documentation content enhancements		
Database storage of JMS messages for durable subscribers	Additional information on how to configure the maximum number of JMS messages to store in the database for durable subscribers.	See section 4.2 for more information.
Collection of statistics on an object	Non-scheduled collection of statistics on an object by using the generic.GenericObject.triggerCollect method.	See section 15.5 for more information.
Resynchronization of an object	Non-scheduled resynchronization by using the generic.GenericObject.triggerResync method.	See Faults and exceptions in section 10.1 for more information.
SSL message encryption	Additional information on configuration requirements.	See section 3.4 for more information.
OAM service validation	OAM validation tests to verify the operational status of a service or service tunnel	See section OAM service validation in section 13.3 for more information.
Policy configuration and distribution enhancements	Policy draft and released modes. Distribution of global policies.	See Policy distribution mode and Policy configuration mode in section 20.1 for more information.
Release 5.0 documentation structure changes		

(4 of 5)

Feature or function	Description	Reference for more information
Reorganization of existing content	Removal of XML configuration samples. The most recent XML configuration samples are now available in the 5620 SAM-O SDK library of XML sample requests.	Contact your Alcatel-Lucent OIPS technical support representative for XML samples from the 5620 SAM-O SDK library of XML sample requests.
	Appendix B refers you to the *5620 SAM Statistics Management Guide* for a complete list and description of performance statistical counters.	See Appendix B for more information. See the *5620 SAM Statistics Management Guide* for more information about network performance statistical counters.
	Structure optimized for the presentation of information on durable and non-durable JMS subscriptions.	See chapter 4 for more information.
Release 5.0 R3 documentation structure changes		
Reorganization of existing content	Structure optimized for the presentation of configuration information: • configuration management overview • equipment configuration management • routing protocol configuration management • customer and residential subscriber configuration management • policy configuration management • service configuration management	See chapters 16, 17, 18, 19, 20, and for more information.
Release 5.0 R5 documentation structure changes		
No significant document structure changes.		

(5 of 5)

2.7 5620 SAM-O Release 4.0 functionality

Table 2-7 lists the 5620 SAM-O release features and documentation changes for 5620 SAM Release 4.0.

Table 2-7 5620 SAM-O Release 4.0 features, problem fixes, and documentation changes

Feature or function	Description	Reference for more information
Release 4.0 R5 features		
Dynamic ACL policies	You can use the 5620 SAM, 5620 SAM-O, and Velocity templates to dynamically manage dynamic ACL policies for 5750 SSC subscriber services.	See chapter 16 for more information about dynamic ACL policies for 5750 SSC subscriber services.
Release 4.0 R3 features		

(1 of 3)

Feature or function	Description	Reference for more information
No new 5620 SAM-O specific features	—	—
Release 4.0 R1 features		
Result filter enhancements	The <resultFilter> element can be used for all methods that support filtered results.	See chapter 6 for more information about filter element types.
	A nested <resultFilter> element can contain the class attribute that specifies the following filtering types: • Package-qualified class name of the returned child object; for example <resultFilter class="equipment.Port">. • Generic base class name for all managed objects; for example, resultFilter class="child class name|ManagedObject">	
	The recursive attribute for the <children> element allows you to filter children objects using the same rules as the parent object; for example, <children recursive="yes"/>. You do not need to use nested <resultFilter> elements if you enable the recursive attribute.	
Durable subscription removal	You can use the 5620 SAM GUI client to close and remove a durable subscription when you no longer require a durable JMS connection. See the *5620 SAM User Guide* for more information about removing durable subscriptions.	See chapter 4 for more information about managing durable subscriptions.
Enhancements to the <findToFile> method	The <findToFile> method runs asynchronously from the invocation of the request. Errors during the query or file creation result in a JMS notification.	See chapter 14 for more information about managing the collection of data from the 5620 SAM server or database.
	You can configure the <findToFile> method to record the query results to a file on a remote server using FTP.	
OSS templates	The OSS templates are 5620 SAM-O scripts that contain parameters that require configuration. You can invoke a template by sending a request that contains the template name and a set of configuration parameters.	—
Content type information in the HTTP response header	You can include the content type information in the HTTP response header by using the <xmlapi> element in the nms-server.xml file. The xmlContentTypeattribute defines the content; for example, the default attribute value of text/xml; charset=ISO-8859-1. You can use any valid string for configuration of the attribute.	See chapter 6 for more information about SOAP message responses.
Release 4.0 R3 problem fixes		
JMS filters	Raise visibility of recommendation that OSS clients must use JMS filters for persistent and non-persistent JMS connections to help limit the number of logged messages.	See chapter 4 for more information.
Release 4.0 R1 problem fixes		
MPLS LSP paths	To configure the <fastRerouteEnabled> and <rsvpStyle> elements for MPLS LSP paths, the 5620 SAM-O logical object model requires that you define the elements on mpls.LspExtension child object.	—
Release 4.0 R3 documentation content enhancements		
SCP-to-SCP connectors	Update to supported encapsulation types.	See chapter 16 for more information about SCP-to-SCP connectors.

(2 of 3)

Feature or function	Description	Reference for more information
Release 4.0 R1 documentation content enhancements		
l2fib and l2fwd	Updated package descriptions.	See chapter 8 for more information about Layer 2 and Layer 3 forwarding classes.
XML JMS messages	All XML JMS messages no longer contain the <correlationId> element in the message header.	See chapter 13 for more information about near-real-time events and alarms using JMS.
	The JMS message terminateClientSessionEvent is broadcast to all sessions.	
	Updated sample JMS *.java file to receive real-time events.	
Release 4.0 R3 documentation structure changes		
Reorganization of existing content	Structure optimized for the presentation of information on persistent and non-persistent JMS connections.	See chapter 4 for more information.
Release 4.0 R1 documentation structure changes		
Reorganization of existing content	The following new chapters have been created based on content from the chapter previously titled, "Developing applications using the 5620 SAM-O": • 5620 SAM-O information model overview • Communication with the 5620 SAM server • Inventory management	See chapters 6, 3, and 14 for more information.

(3 of 3)

2.8 Understanding changes, modifications, and deprecations to the 5620 SAM object model

Each major and minor 5620 SAM-O release includes a schema changes file that documents the following activities, release over release and between releases:

- objects, such as classes, removed or changed from the object model
- packages, such as netw, generic, and snmp, that are added or removed from the object model
- types, such as bitmasks and enumerations, that are added, changed, or removed from the object model
- parameters, such as logFullAction, that are removed or added to classes in the object model

You can use the information in the SchemaChangesReadme HTML file to:

- ensure that existing third-party or in-house OSS applications are correctly referencing the latest 5620 SAM-O object model
- track changes to the object model as third-party or in-house OSS applications are developed using different 5620 SAM-O releases

Note — You should use the SchemaChangesReadme HTML file in addition to the 5620 SAM *XML OSS Interface Developer Guide* of current and previous releases to assess the impact of interface updates.

The SchemaChangesReadme.html file is located on the product DVD-ROM. When you install a client, the file is installed with the software, in the *installation_directory*/nms/distribution/User_Documentation/5620_SAMO_documentation/XML_Reference directory.

The following shows a sample condensed output from the SchemaChangesReadme HTML file:

```
Release 5.0 R4 to 5.0 R5

Packages Added

The following packages are new: schedule

Objects Removed

Types:

 sas.Scope of type: Enumeration

Objects Added

Types:

 sas.TestGroupRole of type: Enumeration

 sas.ExecutionStrategy of type: Enumeration

  Classes:

 mpls.LspTraceDefinition

 mpls.LspPingDefinition

Objects Changed

Types:

 sas.ReturnCode of type: Enumeration

  Added enum downstreamNotInMfib

   Classes:

 mpls.Interface

  Added relativeName [interface-$interfaceId]
```

```
Added property mpls.ActualHop-Set
```

2.9 Upgrade implications

When you upgrade the 5620 SAM, the FDNs may change. 5620 SAM-O clients that use FDNs to reference 5620 SAM objects must update their FDNs accordingly. The FDNs that change during an upgrade are logged to files in the 5620 SAM database installation directory. One log file is created for each upgrade transition in which an FDN changes in the full upgrade path.

For example, if you upgrade from the 5620 SAM, Release 7.0 R1, to the 5620 SAM, Release 8.0 R1, the upgrade process creates FDN mapping log files that have the following names:

- FDN_Mapping_7_0_R1_to_7_0_R2.*date*.log
- FDN_Mapping_7_0_R2_to_7_0_R3.*date*.log
- FDN_Mapping_7_0_R3_to_7_0_R4.*date*.log
- FDN_Mapping_7_0_R4_to_7_0_R5.*date*.log
- FDN_Mapping_7_0_R5_to_7_0_R6.*date*.log
- FDN_Mapping_7_0_R6_to_7_0_R7.*date*.log
- FDN_Mapping_7_0_R7_to_7_0_R8.*date*.log
- FDN_Mapping_7_0_R8_to_8_0_R1.*date*.log

where *date* is the date of the upgrade

Note — The upgrade process automatically migrates data from the start to the end release without the need to upgrade to each intermediate release.

Code 2-1 shows an example of the contents of an FDN mapping log file.

Code 2-1: Sample FDN mapping log file

```
<?xml version='1.0' ?>
<version from="A.0.R1" to="B.0.R1">
<old
Fdn="network:142.165.76.50:router-1:ospf:areaSite-0.0.0.11:interface
-204.83.242.2.0" class="ospf.Interface">
        <new
Fdn="network:142.165.76.50:router-1:ospf-v2:areaSite-0.0.0.11:interf
ace-204.83.242.2.0" class="ospf.Interface"/>
    </old>
    <old
Fdn="network:142.165.76.50:router-1:ospf:areaSite-0.0.0.11:interface
-204.83.242.2.0:neighbor-204.83.242.1.0" class="ospf.Neighbor">
        <new
Fdn="network:142.165.76.50:router-1:ospf-v2:areaSite-0.0.0.11:interf
ace-204.83.242.2.0:neighbor-204.83.242.1.0" class="ospf.Neighbor"/>
    </old>
    <old
Fdn="network:142.165.76.50:router-1:ospf:areaSite-0.0.0.11:interface
-204.83.242.2.0:md 5-1" class="ospf.Md5Key">
        <new
Fdn="network:142.165.76.50:router-1:ospf-v2:areaSite-0.0.0.11:interf
ace-204.83.242.2.0:md5-1" class="ospf.Md5Key"/>
    </old>
    <old
```

```
Fdn="network:142.165.76.50:router-1:ospf:areaSite-0.0.0.11:interface
-204.83.242.9.0" class="ospf.Interface">
        <new
Fdn="network:142.165.76.50:router-1:ospf-v2:areaSite-0.0.0.11:interf
ace-204.83.242.9.0" class="ospf.Interface"/>
    </old>
    <old
Fdn="network:142.165.76.50:router-1:ospf:areaSite-0.0.0.11:interface
-204.83.242.9.0:md5-1" class="ospf.Md5Key">
        <new
Fdn="network:142.165.76.50:router-1:ospf-v2:areaSite-0.0.0.11:interf
ace-204.83.242.9.0:md5-1" class="ospf.Md5Key"/>
    </old>
</version>
```

5620 SAM-O communications

3 — Communication with the 5620 SAM server

4 — Event monitoring using JMS

5 — XML requests

3 — Communication with the 5620 SAM server

3.1 Overview *3-2*

3.2 5620 SAM system components *3-2*

3.3 Client interfaces *3-3*

3.4 Secure communication *3-5*

3.5 Workflow to set up and operate 5620 SAM-O *3-10*

3.1 Overview

An OSS can communicate with the 5620 SAM through the following 5620 SAM interfaces:

- the JMS event stream and SOAP XML API, which are collectively called the 5620 SAM XML OSS interface
- the 3GPP OSS inteface

The 5620 SAM XML OSS interface receives the JMS event stream, which is used for near-real-time event notification of changes in the network. The XML API is used to change configurations and to request status information.

The 3GPP OSS interface is a CORBA-based API that provides a 3GPP-compliant layer for client application communication with a 5620 SAM-managed LTE network. See the *5620 SAM-O 3GPP OSS Interface Developer Guide* for more information about the 5620 SAM 3GPP OSS interface.

Developers that create OSS applications are expected to be familiar with the following:

- Standard XML schema constructions
- OSS interfaces
- Managed devices
- 5620 SAM functionality.

The following sections describe the main components of the 5620 SAM system and the interfaces in the 5620 SAM server communications with the 5620 SAM-O clients.

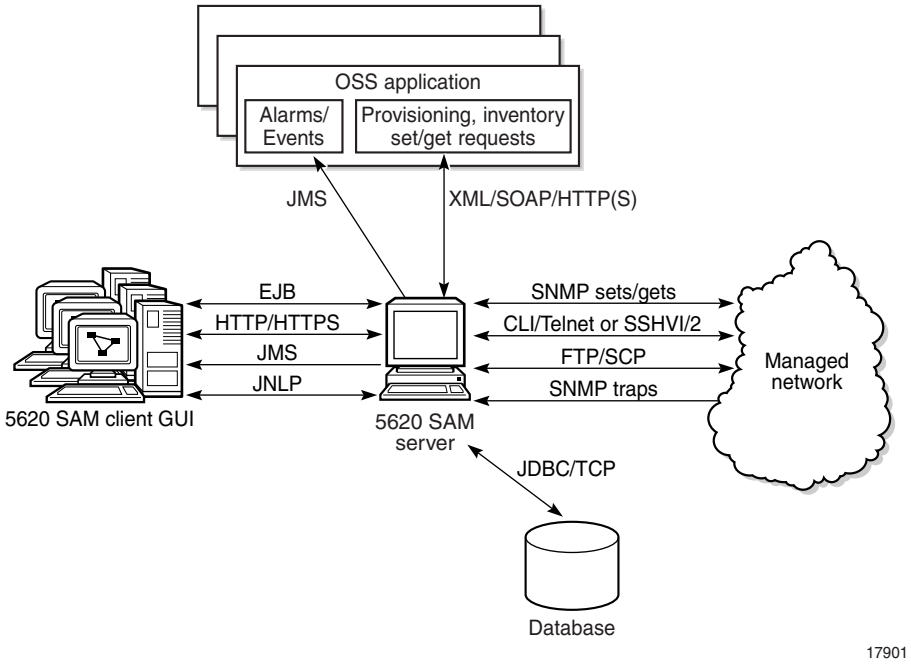
3.2 5620 SAM system components

The following are the main components of a 5620 SAM system.

- The main server is a network-management processing engine written in Java that runs on a Solaris platform. The server includes several third-party components, such as an application server, JMS server, web server, protocol stack set, and database adaptor. Server functionality can be concentrated on one physical platform, or distributed across multiple dedicated stations.
- The clients are OSS applications or Java-based 5620 SAM GUIs. The OSS applications can run on any platform, because they use platform-neutral protocols to communicate with the 5620 SAM main server. The 5620 SAM GUI and 5620 SAM-O clients do not communicate directly with auxiliary servers.
- The database is a customized Oracle relational database that provides persistent storage for the network data. The database can run on the same station as the 5620 SAM main server software, or on a different station.

The 5620 SAM component interfaces use industry-standard protocols for communication between servers and the database, managed network devices, and clients, as shown in Figure 3-1.

Figure 3-1 5620 SAM interfaces

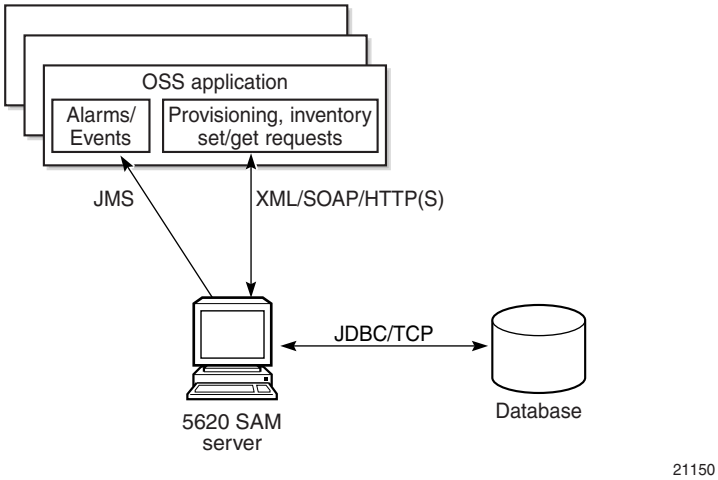


3.3 Client interfaces

Client interfaces provide access to the 5620 SAM main server and to the managed network, as shown in Figure 3-2. The 5620 SAM-O clients can:

- send requests to the 5620 SAM main server to view and change objects, and to perform network operations
- monitor the 5620 SAM, and network events and alarms.

Figure 3-2 Client interfaces



Event monitoring using JMS

Clients can monitor events in the 5620 SAM and the network by subscribing to event streams. This event-driven function allows a client to maintain a near-real-time view of events in the network. However, the function is unidirectional and does not allow clients to initiate changes in the 5620 SAM or the network.

To use JMS event streams to monitor events, the OSS developers need to create a JMS client. The client must subscribe to one or more event streams, which can be used to monitor events. Some applications are:

- monitoring alarms
- maintaining up-to-date inventory information by monitoring configuration changes in the network

See chapter 4 for more information about JMS and event monitoring.

Request and response using the 5620 SAM-O

Clients can request information and initiate changes in the 5620 SAM and in the managed network using the 5620 SAM-O API. This function is bidirectional. Clients send requests and receive responses that contain the information requested or the results of an operation.

The 5620 SAM-O API uses HTTP. The OSS developers must create a client that can send HTTP requests and receive responses. Some applications are:

- retrieving network inventory information
- changing the network configuration
- provisioning

The 5620 SAM-O API is also referred to as the XML API, although both the API and event monitoring use XML.

See chapter 5 for more information about using 5620 SAM-O API.

XML and SOAP

The JMS events and HTTP requests and responses are written in XML using the SOAP standard. XML is a standard format used to describe data. The 5620 SAM-O XML schema is defined in XML style sheets, which are distributed with the 5620 SAM-O documentation. API requests and responses, and JMS events are wrapped in SOAP envelopes.

For more information, see the related W3C specifications and other related material available from technical publishers and the Internet. Some suggested resources are:

- http://www.w3.org/XML
- http://www.w3.org/TR/soap
- Ray, Eric T. Learning XML (2nd Ed). O'Reilly, 2003

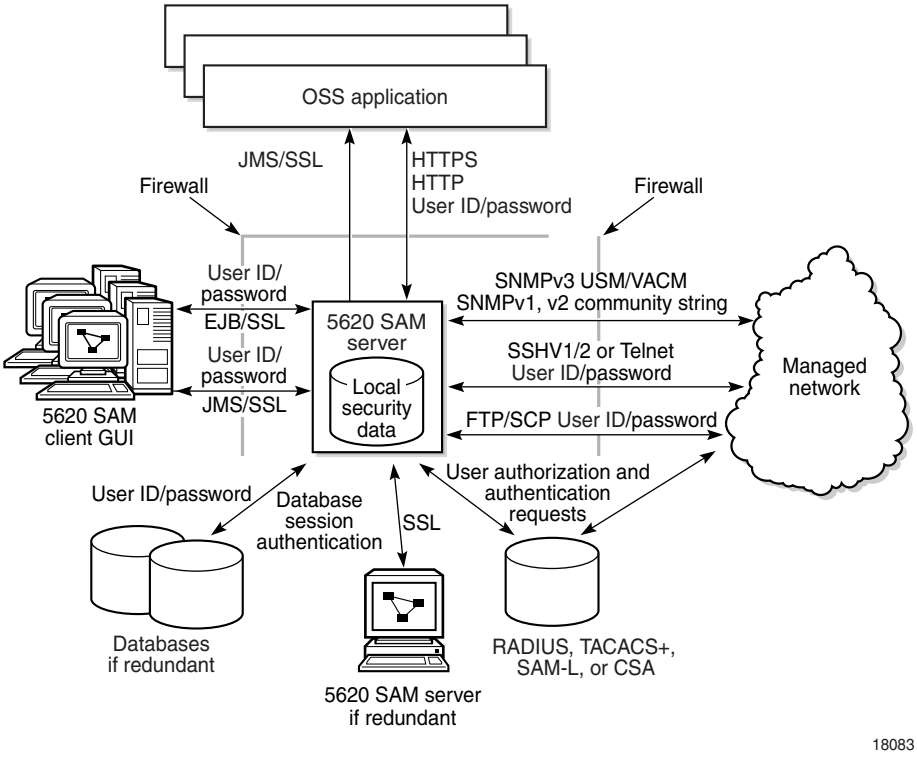
3.4 Secure communication

The 5620 SAM-O provides the following features for the secure communications between the 5620 SAM client OSS and the 5620 SAM-O server:

- HTTPS for XML requests and responses
- MD5-hashed passwords for requests
- SSL message encryption for JMS events

Figure 3-3 shows the 5620 SAM components and the available security mechanisms.

Figure 3-3 5620 SAM security



SSL message encryption

5620 SAM-O clients can receive JMS XML messages securely by encrypting them through SSL. The SSL Certificate is configured on the 5620 SAM server. You must also configure the following client components:

- SSL keystore file on the OSS client
- Java to use the keystore file on the OSS client

See Procedure 4-3 and the Java user documentation for more information on configuring Java for SSL.

HTTPS communication with 5620 SAM-O

The 5620 SAM-O provides an optional web-based encryption protocol for sending requests and receiving responses between the 5620 SAM-O server and client. See the *5620 SAM User Guide* for information on how to configure the 5620 SAM server for HTTP or HTTPS.

Session management

Effective session management requires authentication, authorization, and accounting (AAA) functionality. Authentication is the verification of a user identification and password. Authorization is the assignment of different levels of access permissions to users. Accounting is the recording of user actions. A 5620 SAM operator can configure AAA functionality using the local security capability of the 5620 SAM server, a third-party authentication server, or a combination of local and third-party mechanisms.

- Local authentication on the 5620 SAM server is provided with a local database of users and a local security scheme to verify logon attempts and assign permission levels for command execution.
- Supported third-party authentication servers are RADIUS and TACACS+. These products run on their own platforms, with their own user lists and administration processes.

5620 SAM user accounts consist of a user name, password, priority, and an associated user group with scope of command and span of control profiles. User groups are used to assign and control user authorization levels, and to control the extent of access to such entities as customers, services, or faults. Users and groups can be assigned priorities that determine which user sessions take priority over others when system resources are limited. The system administrator can specify the maximum number of active user connections per user or user group. The system administrator can also limit the type of user access per managed device; for example, users might not be able to open a console.

The system administrator can use the 5620 SAM client GUI to manage 5620 SAM-O user account privileges, user sessions, SNMP format, user and group priority, and user authentication. See the *5620 SAM User Guide* for more information.

Client sessions

All 5620 SAM-O client sessions have username and password protection.

- Each OSS client XML/SOAP message is individually authenticated using cached information from an authorization server.
- JMS messages are protected by the user name and password for the JMS connection.

You can use an MD5-hashed or a clear text password to access the 5620 SAM server.

MD5-hashed password format

Alcatel-Lucent recommends that you send the password for a request in an MD5-hashed format. You can use the md5hash password utility to generate MD5-hashed passwords to access the 5620 SAM server. See Procedure 3-1 for more information about how to generate an MD5-hashed password for 5620 SAM server access.

Clear text password format

OSS users can access the 5620 SAM server using a clear text password. A clear text password provides no security for transmitted passwords unless HTTPS is used, but ensures greater compatibility with RADIUS and TACACS+ password storage. Clear text passwords should only be used if:

- the 5620 SAM-O uses an external mechanism for password authentication (such as RADIUS or TACACS)

  and
- a dedicated HTTPS channel is used for all XML/SOAP requests

Code 3-1 shows an example of the clear text password format.

Code 3-1: Sample clear text password format

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope
  xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP:Header>
        <header xmlns="xmlapi_1.0">
            <security>
                <user>user name</user>
                <password hashed="false">password</password>
            </security>
            <requestID>clientName@requestId</requestID>
        </header>
    </SOAP:Header>
<SOAP:Body>
         <ping xmlns="xmlapi_1.0"/>
</SOAP:Body>
</SOAP:Envelope>
```

Procedure 3-1 To generate an MD5-hashed password for 5620 SAM main server access

1 Log on to a 5620 SAM main server station as the samadmin user.

2 Open a console window.

3 Enter the following to navigate to the server binary directory:

```
bash$ cd path/server/nms/bin ↵
```

where *path* is the main server installation directory, typically /opt/5620sam/server

4 Enter the following to generate an MD5-hashed password:

```
bash$ md5hash password string ↵
```

where *password string* is the list of words that you want to convert to a password in MD5-hashed format

For example, the MD5 hashed password for the word hello is the following:

5d41402abc4b2a76b9719d11017c592

5 Add the MD5-hashed password to the password field of the SOAP header for the specified user account.

5620 SAM-O security features controlled through the 5620 SAM client GUI

A system administrator can use the security management forms in the 5620 SAM client GUI to control the following security-related features for 5620 SAM-O:

- User account privileges
- Session monitoring
- Session shutdown
- SNMPv3 for secure NE polling
- Radius and TACAS+ authentication for 5620 SAM users
- Session logs of 5620 SAM client GUI and OSS activity. The 5620 SAM-O client ID must use the format for the JMS client ID described in section 4.3 to be uniquely identified as a 5620 SAM-O session.

You must create a scope of command profile to group the OSS Management scope of command role with additional roles. Otherwise, the 5620 SAM returns the following SOAP exception message:

```
<SOAP:Fault>
<faultcode>SOAP:Client</faultcode>
<faultstring>[security] Users require OSS Management privileges to
use SAM-O.</faultstring>
<faultactor>XmlApi</faultactor>
- <detail>
<requestID>XmlApiClient:0</requestID>
</detail>
</SOAP:Fault>
```

See the *5620 SAM User Guide* for information about managing security on 5620 SAM user groups and the associated scope of command and span of control profiles.

If a 5620 SAM-O user supplies an incorrect username or password, or if the username or password is missing, the 5620 SAM returns the following SOAP exception message:

```
<SOAP:Fault>
<faultcode>SOAP:Client</faultcode>
<faultstring>[security] Login failure.</faultstring>
```

```
<faultactor>XmlApi</faultactor>
- <detail>
<requestID>XmlApiClient:0</requestID>
</detail>
</SOAP:Fault>
```

If a 5620 SAM-O user is not assigned the appropriate scope of command privilege for a class type, the following exception is returned:

```
[ app: generic ] [ class: GenericObject ] [ instance: instance:object ]
[ cause: Method Invocation Failed (13) ]
[ descr: Insufficient privileges to perform this operation.
User does not have create-update-delete access to class classname ]
[ nested exception: null ]
```

If a 5620 SAM-O user is not assigned the appropriate span of control for an object, the following exception is returned:

```
[ app: generic ] [ class: generic.GenericObject ] [ instance:
-unknown- ] [ descr: SecurityManager: Authorization Failure:
User does not have access to objectFullName. Object is not in user's span.
[exceptionClass: SecurityException]
```

See the *5620 SAM User Guide* for information about how listing is filtered by span, and for how configuration is span enforced.

Unlike in HTTP or HTTPS where span of control rules are implicitly enforced, span of control rules are not implicitly enforced via JMS. JMS requires the administrator to explicitly specify the required span IDs within the JMS filter. For example, without the correct JMS filter, it may not be possible to listen to events from objects that are not in the 5620 SAM-O user's span of control. Table 3-1 lists example JMS filter entries.

Table 3-1 Example JMS filter entries

JMS filter	Description
ALA_span like '%:2:%''	A JMS query that uses this filter results in JMS messages being sent from objects in span 2 (for example, Default Router Span). Events from objects in other spans are not published.
ALA_span like '%:21:%'' or ALA_span like ‘%:0:%’	A JMS query that uses this filter results in JMS messages being sent from objects in span 21 or non-span objects. Events from objects in other spans are not published.
(ALA_span like ‘%:1:%’ or ALA_span like ‘%:2:%’ or ALA_span like ‘%:3:%’ or ALA_span like ‘%:4:%’ or ALA_span like ‘%:5:%’ or ALA_span like ‘%:6:%’ or ALA_span like ‘%:7:%’ or ALA_span like ‘%:8:%’ or ALA_span like '%:0:%') and (ALA_span not like ‘%:22:%’)	A JMS query that uses this filter results in JMS messages being sent from all objects (all of Default Spans listed) except those blocked in span 22.

3.5 Workflow to set up and operate 5620 SAM-O

Use the following steps to set up and operate the 5620 SAM-O.

1. Ensure that the 5620 SAM server is installed and that the license key supports the use of the 5620 SAM-O software. See the *5620 SAM | 5650 CPAM Installation and Upgrade Guide* for more information.

2. Set up communication between the OSS client application and the 5620 SAM server. Specify the active server if you are connecting to the 5620 SAM server in redundant mode. See Procedure 5-1 for more information.

3. Develop an understanding of the information model and the 5620 SAM-O XML message structure. See chapter 6.

4. Determine whether the OSS application is request-response driven, or acts as a passive event listener (many OSS applications do both). See "Client interfaces" for an overview of both approaches.

Note — Training and developer support are available from Alcatel-Lucent. Contact your Alcatel-Lucent account or OIPS technical support representative for more information.

5. If you are communicating with the 5620 SAM-O connection using JMS, see "Workflow to set up and operate a JMS connection".

   If you are communicating with the 5620 SAM-O connection using XML, see "Workflow to set up and operate an HTTP XML request-response connection".

4 — Event monitoring using JMS

4.1 Overview *4-2*

4.2 JMS connections *4-2*

4.3 JMS subscriptions *4-3*

4.4 JMS Events *4-7*

4.5 JMS and redundancy *4-23*

4.6 Connection monitoring and error recovery *4-24*

4.7 Managing and monitoring JMS sessions *4-29*

4.8 JMS consumer creation *4-31*

4.9 Workflow to set up and operate a JMS connection *4-33*

4.10 JMS record and playback tool *4-34*

4.1 Overview

The 5620 SAM-O clients use JMS channels to receive real-time network event information from the server. The event types that 5620 SAM-O clients receive include:

- managed network alarm notifications
- managed network configuration changes
- server activity-switch notifications
- 5620 SAM database connectivity errors

A JMS consumer is a JMS client that consumes events from a JMS provider. In the 5620 SAM-O, the 5620 SAM is the JMS provider and the OSS is the consumer. An OSS developer must write a JMS client that connects to the 5620 SAM, subscribes to the topic or topics that are relevant to the OSS, and processes events.

The following sections provide the information necessary to create a JMS client that communicates with the 5620 SAM, in addition to sample code for a JMS client. You must have a thorough understanding of JMS to monitor 5620 SAM-O events and understand the topics discussed in this chapter. JMS is a published industry standard and is beyond the scope of the 5620 SAM *XML OSS Interface Developer Guide*. See http://www.oracle.com/technetwork/java for more information.

4.2 JMS connections

An OSS application monitors events by establishing a JMS connection to the 5620 SAM, then by creating a subscription to one of the available JMS topics.

Note — The 5620 SAM supports a maximum of 10 JMS connections at a time. Since these resources are shared, Alcatel-Lucent strongly recommends that an OSS use only one connection. If an OSS uses more than one connection, Alcatel-Lucent requires that the number of connections be configurable at runtime, that one connection be a valid option, and that the number of connections used and how to configure this be clearly documented.

You require the following information to establish a JMS connection to the 5620 SAM:

- a server address and port
- a 5620 SAM-O user name and password

The server address can be the IP address of the server or the server name. The default port for 5620 SAM standalone installations is port 1099. The default port is 1100 for redundant installations. See section 4.5 for more information about how to connect to a redundant setup. The 5620 SAM server ports and addresses must be configurable in each client OSS application. If a firewall exists between the 5620 SAM server and client OSS application, multiple ports must be opened in the firewall. See the *5620 SAM Planning Guide* for the list of ports required for JMS and other connections.

When you create a connection, you require a valid 5620 SAM username and password. The password can be MD5-hashed or in plain text. The 5620 SAM user must use a scope of command profile that includes the OSS role. See the *5620 SAM User Guide* for more information about how to manage 5620 SAM users and profiles.

4.3 JMS subscriptions

After you establish a JMS connection, you require the following to create a new subscription:

- a client ID
- a topic
- a filter
- an acknowledgement mode
- a persistence mode

Note — Alcatel-Lucent strongly recommends that an OSS establishes only one JMS subscription for each JMS connection.

Client ID

You must establish a client ID for the JMS client before you create a JMS subscription. The client ID is used to identify a client in the JMS and 5620 SAM logs, and in the 5620 SAM GUI. In addition, the client ID is used to send JMS events to specific subscribers, such as missed events—when only a single subscriber is impacted—or file available events for files requested by a specific client. The client ID must also be used in the subscription filter.

The 5620 SAM supports the following format for the client ID:

<unique_name>@<numeric_id>

The *<unique_name>* must be a string that identifies the client to which the subscription belongs, usually the name of the OSS product. All of the subscriptions from the same OSS must use the same <unique_name>. The *<numeric_ID>* is used to differentiate between different subscriptions that belong to the same OSS.

JMS topics

A JMS client controls the events which it receives by subscribing to the appropriate JMS topic and filtering the events. This section describes the topics that available from the 5620 SAM. JMS clients can subscribe to any of the topics listed in Table 4-1. See section "Message filters" for more information about filters.

Table 4-1 JMS topics

Topic	Type	Description
5620-SAM-topic-xml	XML	Subscribe to all events, such as object creation and deletion, in an XML format. This topic publishes all events with no dependency on other topics.
5620-SAM-topic-xml-general	XML	Subscribe to general events, such as object creation and deletion, in an XML format. This topic contains objects that do not fall into the fault, file, or stats topics.
5620-SAM-topic-xml-fault	XML	Subscribe to fault events in an XML format
5620-SAM-topic-xml-file	XML	Subscribe to findToFile events in an XML format
5620-SAM-topic-xml-stats	XML	Subscribe to events related to statistics collection in an XML format

The 5620-SAM-topic-xml topic allows you to receive all events. You can use the other topics to receive specific events—for example, the 5620-SAM-topic-xml-fault topic specifies fault-specific messages.

Message filters

A JMS subscription must include a filter that defines the types of messages the client receives. The message filter is defined when the subscription is created. You cannot change the message filter unless you unsubscribe and resubscribe.

A JMS filter is a string with a maximum 3000 characters that uses the same syntax as the WHERE clause in an SQL query, following the ANSI SQL-92 standard.

The following is an example of a simple filter:

```
ALA_clientId in ('ossname@1', '')
```

This simple filter includes all the events in the topic that are intended for the subscriber ossname@1, and all the events in the topic that are intended for all subscribers.

Note — All subscriptions must filter on the client ID. Subscriptions that do not include this filter are rejected.

Alcatel-Lucent recommends that you create filters to be as restrictive as possible so that only necessary events are included. If the filter is not restrictive enough, unnecessary events are included, queued, and processed, which can lead to the following:

- Increased likelihood of queue overflows resulting in missed events. See section “Missed events” for more information.
- Wasted bandwidth
- Negative performance impact on both the 5620 SAM and the OSS

Filters can use any of the properties from the JMS header to filter events. Some common properties used for filtering include the notification type (MTOSI_NTType), the event category (ALA_category), and the object type (MTOSI_objectType). See Table 4-4 for a complete list of properties, and Table 4-5 for properties that are specific to alarm-related events. Properties are formally defined in xmlApiJmsHeader.xsd in the OSS XML Schema. See chapter 9 for more information about using the XML Schema.

Table 4-2 lists additional examples of JMS message filters.

Table 4-2 JMS filter samples

JMS filter	Description
ALA_clientId in ('ossi@2', '')	JMS filter to send general messages, and messages qualified with a client ID, directly to the specified client. This filter blocks messages sent to other clients. You must specify this information in all XML topic based filters.
ALA_clientId in ('ossi@2', '') and ALA_category = 'FAULT' and MTOSI_perceivedSeverity = 'Warning'	JMS filter on fault messages with a warning severity class. This filter excludes some mandatory events. See section Mandatory events for more information.
ALA_clientId in ('ossi@2', '') and ALA_category not in ('STATISTICS', 'ACCOUNTING')	JMS filter to stop client statistics, accounting, and keepalive messages
ALA_clientId in ('ossi@2', '') and (MTOSI_NTType in ('NT_ATTRIBUTE_VALUE_CHANGE', 'NT_OBJECTDELETION', 'NT_OBJECTCREATION') and MTOSI_objectType in ('equipment.PhysicalPort', 'equipment.DaughterCard', 'equipment.BaseCard'))	JMS filter to include only object creation, deletion, and attribute changes for cards and ports. This filter excludes some mandatory events. See section Mandatory events for more information.
ALA_clientId in ('ossi@2', '') and (MTOSI_NTType in ('NT_OBJECTDELETION', 'NT_OBJECTCREATION') and MTOSI_objectType like 'equipment.%')	JMS filter to include only object creation and deletion for objects in the equipment package. This filter excludes some mandatory events. See section Mandatory events for more information.

Mandatory events

There are several JMS events that an OSS client must process that you must ensure are not inadvertently filtered. You must define the filter so that the following mandatory events are not filtered:

- KeepAliveEvent
- StateChangeEvent
- TerminateClientSessionEvent
- JmsMissedEvent
- all events destined for the client ID

The following sample filter ensures that all mandatory events are included.

```
ALA_clientId in ('ossname@1', '')
  AND (
   (<filter to include events of specific interest to the OSS>)
   OR
```

```
   MTOSI_objectType in ('KeepAliveEvent', 'StateChangeEvent',
'TerminateClientSession')
   )
```

Optional properties

Most header properties apply to all events. However, some alarm-related properties —for example, ALA_alarmType—apply only to certain events. When you create a filter, you must consider that a specific property will exclude all the events that do not have the property.

For example, the following filter includes all events except equipment alarms:

```
ALA_clientId in ('ossname@1', '') and ALA_alarmType not in
('equipmentAlarm')
```

However, this filter also excludes all events that do not include the ALA_alarmType property.

The following filter excludes events with an ALA_alarmType of equipmentAlarm, and includes all other events:

```
ALA_clientId in ('ossname@1','') and (ALA_alarmType is null or
ALA_alarmType not in ('equipmentAlarm'))
```

Alarm-specific properties do not apply to all alarm events. When you create a filter using alarm-specific properties, you must include the 'is null' case.

The following Boolean example is a filter applied to exclude correlated alarms, but include alarms that do not have the ALA_isCorr property present. Note that in Boolean examples, “quotes” are not required for true or false values.

```
ALA_clientId in ('ossname@1','') and (ALA_isCorr is null or ALA_isCorr
= true)
```

Acknowledgement modes

Clients can specify an acknowledgement mode for JMS sessions. The following acknowledgement modes are supported:

- **AUTO_ACKNOWLEDGE**
  The JMS client automatically acknowledges each message received from the server and another message is not sent by the server until an acknowledgement is received from the JMS client. This mode does not efficiently use the available bandwidth, and performance degrades significantly as latency increases.
- **DUPS_OK_ACKNOWLEDGE**
  This mode allows new messages to be sent to the JMS client before earlier messages are acknowledged. This mode allows for a higher throughput for messages in high-latency networks. In addition, this mode reduces session overhead and per message delay by minimizing the work the session does to prevent duplicates. Available bandwidth to the client is used more efficiently and performance does not degrade significantly as latency increases.

Alcatel-Lucent recommends that you use the DUPS_OK_ACKNOWLEDGE mode because of the improvements in performance. However, consider the following when you implement the JMS consumer. If there is a connectivity failure, the 5620 SAM re-sends all unacknowledged messages once connectivity is restored. When you use the AUTO_ACKNOWLEDGE mode, this can result in the client receiving a single duplicate message—if that message was received by the client but the connectivity failure prevented the 5620 SAM from receiving the acknowledgement. When you use the DUPS_OK_ACKNOWLEDGE mode, there may be several messages at any given time that have been sent by the 5620 SAM for which the 5620 SAM has not received an acknowledgement. All of these messages are resent once connectivity is restored.

Code 4-1 shows DUPS_OK__ACKNOWLEDGE as the message acknowledgement mode.

Code 4-1: JMS client message acknowledgement mode

```
topicSession =
topicConnection.createTopicSession(false,TopicSession.DUPS_OK_ACKNOWLEDGE );
```

Persistence modes

There are two persistence modes available to subscribers—durable and non-durable. Durable subscriptions can continue to exist even if the client is not connected. If the client is not connected, the 5620 SAM queues messages until the client reconnects. Non-durable subscriptions are removed if the client is disconnected, and messages are not queued.

The tolerance of the OSS for missed event determines whether durable or non-durable subscriptions should be used. If there is a loss of connectivity, non-durable subscriptions lose events and may also lose events silently as a result of overflows of internal server queues. For durable subscriptions, events are queued if there is a connectivity failure, and are available when connectivity is restored.

For both durable and non-durable subscriptions, if an internal queue overflow results in a missed event, the subscriber receives notification of missed events which allows them to take corrective action. See section "Missed events" in section 4.6 for more information about missed events.

4.4 JMS Events

An OSS subscribes to a JMS topic to listen to events. The following sections describe the format of these events, typical events, a list of the available event classes, and event documentation in the XML schema files.

JMS XML schemas

JMS XML schemas describe the structure and allowed elements of a document, similar to a DTD.

The DVD-ROM for the 5620 SAM application contains information about the 5620 SAM-O schemas, objects, methods, parameters, and release deltas. You can access the information using the index file in the user documentation directory.

Table 4-3 list the files associated with the JMS XML schema.

Table 4-3 JMS XML schema files

File	Description
xmlApiJms.xsd	JMS general messages
xmlApiJmsHeader.xsd	JMS messages for filtering
xmlApiJmsTypes.xsd	Types used in the JMS header and body
xmlApiJmsBody.xsd	Event body format

Some events contain complete objects defined in the standard 5620 SAM XML schemas. See section "JMS event classes" for more information.

See section "XML and SOAP" in chapter 3 for information on the W3C standards for XML.

JMS event header properties

The following tables outline the properties included in the JMS message headers. Table 4-4 lists the properties that are included in all events. Table 4-5 lists properties that are included in some alarm-related events.

Table 4-4 Event header properties—all events

Property	Values	Description
MTOSI_NTType	ALA_OTHER NT_OBJECTDELETION NT_OBJECTCREATION NT_ATTRIBUTE_VALUE_CHANGE NT_STATE_CHANGE NT_ALARM NT_HEARTBEAT ALA_RELATIONSHIPCHANGE	Defines the type of notification
MTOSI_osTime	Time in milliseconds	Defines when the event was created on the server
MTOSI_objectName	Object name	Defines the object name for the event

(1 of 2)

Property	Values	Description
MTOSI_objectType	KeepAliveEvent AlarmStatusChangeEvent DBActivityEvent DBProxyStateChangeEvent DBErrorEvent DBConnectionStateChangeEvent StateChangeEvent FileAvailableEvent DeployerEvent TerminateClientSession Any network object names in the form of *packageName.ClassName* for the following object events: AttributeValueChangeEvent ObjectCreationEvent ObjectDeletionEvent RelationshipChangeEvent StatsEvent	Defines the object type for the event
ALA_category	SERVICE—service-related events (SAP, VPLS, VPRN, and so on) EQUIPMENT—physical equipment events (port, card, and so on) ACCOUNTING—accounting statistics GENERAL—values not covered by other values (for example, fault, statistics, and database) FAULT—fault management events (alarms) STATISTICS—statistics, except accounting statistics DATABASE—database state messages (connection up, proxy state change, and so on) SOFTWARE—not currently used CPAM—events related to routing topology management	Defines the category of the event
ALA_allomorphic	Allomorphic class for the object type	Defines the allomorphic class for the object type
ALA_topic	JMS Topic	Identifies the JMS topic for the message
ALA_clientId	Specific client ID or empty	Identifies the client ID for the message, if required
ALA_OLC	0 (object has no OLC state) 1 (maintenance) 2 (in service)	Object life cycle state. Applies only to object creation, attribute changes, and deletion on objects with an OLC state. If the object does not have an OLC state, the default value is 0. See the *5620 SAM User Guide* for more information about the OLC state.
ALA_eventName	See Table 4-6 for a list of the supported objects	Defines the class name of the JMS event. For example, ObjectDeletion for the deletion event.

(2 of 2)

Table 4-5 Event header properties—alarm events

Property	Values	Description
MTOSI_aliasNameList	Alarm Name	Defines the alarm name
MTOSI_probableCause	Probable Cause	Defines the probable cause of the alarm
MTOSI_perceivedSeverity	Severity	Defines the severity of the alarm
MTOSI_serviceAffecting	True if service affecting, False otherwise	Identifies if the alarm is service affecting
ALA_alarmType	Alarm Type	Identifies the type of alarm
ALA_isCorr	True if alarm is correlated False otherwise	Indicates whether the alarm is correlated, that is, caused by another alarm. See the *5620 SAM User Guide* for more information about correlated alarms.

JMS event classes

Table 4-6 lists the JMS event classes.

Table 4-6 JMS event classes

Type of class	Description	See subsection for more information
StateChangeEvent	Identifies different state changes in the server, such as alarm information loaded or alarm count changed. An example of an important state change event is the SystemInfoEvent, which is sent each time a consumer creates a new subscription or connects to an existing subscription.	StateChangeEvent
KeepAliveEvent	Keep-alive messages are used to ensure that the server is running, and in communication with the OSS client that runs the OSS application.	KeepAliveEvent
AlarmObject	All alarms are defined as an AlarmObject	—
AlarmStatusChangeEvent	Correlates alarms. This event class identifies the status changes on all alarm-based attributes.	AlarmStatusChangeEvent
AttributeValueChangeEvent	Changes to one or more attributes in the database, based on the class that changes	AttributeValueChangeEvent
AttributeChange	Represents an attribute change, and contains both the old and the new values of the attribute. The AttributeValueChangeEvent contains the TiAttributeChange.	—
ObjectCreationEvent	Created object details	ObjectCreationEvent
ObjectDeletionEvent	Deleted object details	ObjectDeletionEvent
DBActivityEvent	Identifies database switchover or failover status	DBActivityEvent
DBProxyStateChangeEvent	Identifies database proxy status	DBProxyStateChangeEvent

(1 of 2)

Type of class	Description	See subsection for more information
DBErrorEvent	Raises an event when a database error has occurred, or a previous database error has been fixed	DBErrorEvent
DBConnectionStateChangeEvent	Indicates that the state of the database connection has changed	DBConnectionStateChangeEvent
RelationshipChangeEvent	Indicates that a relationship has changed between objects	RelationshipChangeEvent
RelationshipChangeDefinition	Defines a single relationship change. The RelationshipChangeEvent contains the RelationshipChangeDefinition.	—
FileAvailableEvent	Indicates that a file is available. This event is applicable to asynchronous and synchronous findToFile requests.	FileAvailableEvent
TerminateClientSessionEvent	Identifies a client that should be closed. This is a security-related event.	TerminateClientSessionEvent
DeployerEvent	Identifies the success or failure of an asynchronous request for deployment. This event can only be viewed on XML topics.	DeployerEvent
StatsEvent	Identifies the type of statistics (polling or accounting), the collection time, and the network element for which the statistics are collected. The StatsEvent is sent at the start of polling and at the end of polling for the specified network element, as indicated by the <state> parameter.	StatsEvent
ScriptExecutionEvent	Sent when a script is executed through OSS or the SAM client. If specified, the message is sent to just the client (based on filter) that executed the script. Identifies if the script execution was successful, the result full name, and the location of the file containing the result.	ScriptExecutionEvent
ExceptionEventXMLFormat	Reports when an exception occurs. For example, deployment or policy distribution errors.	ExceptionEventXMLFormat
LogFileAvailableEvent	Notifies of accounting export file creation for statistics retrieval.	LogFileAvailableEvent
ManagedRouteEvent	Identifies route objects managed by the 5650 CPAM.	managedRouteEvent
IncrementalRequestEvent	Generated for activity related to generic.GenericObject.triggerIncrementalRequest.	IncrementalRequestEvent

(2 of 2)

StateChangeEvent

Table 4-7 describes the different state change events.

Table 4-7 State change events

Event type	Description
SystemInfoEvent	This event is sent each time a client connects to a new subscription or reconnects to an existing durable subscription.

(1 of 2)

Event type	Description
JmsMissedEvents	This event is sent to inform both durable and non-durable subscribers that events have been missed.
AlarmInformationLoaded	This event is sent to indicate changes in the status of the alarm list. See the XML Schema for the possible status values.
AlarmCountChanged	This event is sent when the number of alarms crosses above or below the maximum number of alarms defined in the nms-server.xml file.

(2 of 2)

Code 4-2 shows sample event output for the SystemInfoEvent class.

Code 4-2: SystemInfoEvent output

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP:Header>
      <header xmlns="xmlapi_1.0">
        <eventName>SystemInfoEvent</eventName>
        <MTOSI_osTime>1224527204299</MTOSI_osTime>
        <ALA_clientId>oss_client@1</ALA_clientId>
        <MTOSI_NTType>ALA_OTHER</MTOSI_NTType>
        <MTOSI_objectType>StateChangeEvent</MTOSI_objectType>
        <ALA_category>GENERAL</ALA_category>
        <ALA_isVessel>false</ALA_isVessel>
        <ALA_allomorphic/>
        <ALA_eventName>SystemInfoEvent</ALA_eventName>
        <MTOSI_objectName/>
        <ALA_OLC>0</ALA_OLC>
      </header>
    </SOAP:Header>
    <SOAP:Body>
      <jms xmlns="xmlapi_1.0">
        <stateChangeEvent>
          <eventName>SystemInfoEvent</eventName>
          <state>systemInfo</state>
          <sysStandbyIp>138.120.182.253</sysStandbyIp>
          <sysPrimaryIp>138.120.169.94</sysPrimaryIp>
          <jmsStartTime>1224525628189</jmsStartTime>
          <sysStartTime>1224525628189</sysStartTime>
          <sysType>Redundant</sysType>
        </stateChangeEvent>
      </jms>
    </SOAP:Body>
  </SOAP:Envelope>
```

Code 4-3 shows sample event output for the JmsMissedEvents class.

Code 4-3: Sample JmsMissedEvents output

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP:Header>
      <header xmlns="xmlapi_1.0">
        <eventName>JmsMissedEvents</eventName>
        <MTOSI_osTime>1222891102168</MTOSI_osTime>
        <ALA_clientId>oss_client@1</ALA_clientId>
        <MTOSI_NTType>ALA_OTHER</MTOSI_NTType>
        <MTOSI_objectType>StateChangeEvent</MTOSI_objectType>
        <ALA_category>GENERAL</ALA_category>
        <ALA_isVessel>false</ALA_isVessel>
        <ALA_allomorphic/>
        <ALA_eventName>JmsMissedEvents</ALA_eventName>
        <MTOSI_objectName/>
        <ALA_OLC>0</ALA_OLC>
```

```
      </header>
    </SOAP:Header>
    <SOAP:Body>
      <jms xmlns="xmlapi_1.0">
        <stateChangeEvent>
          <eventName>JmsMissedEvents</eventName>
          <state>jmsMissedEvents</state>
        </stateChangeEvent>
      </jms>
    </SOAP:Body>
  </SOAP:Envelope>
```

KeepAliveEvent

Code 4-4 shows sample event output for the KeepAliveEvent class.

Code 4-4: Sample KeepAliveEvent output

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP:Header>
      <header xmlns="xmlapi_1.0">
         <MTOSI_osTime>1137166251642</MTOSI_osTime>
         <ALA_clientId />
         <MTOSI_objectType>KeepAliveEvent</MTOSI_objectType>
         <MTOSI_NTType>NT_HEARTBEAT</MTOSI_NTType>
         <ALA_category>GENERAL</ALA_category>
         <ALA_allomorphic />
         <MTOSI_objectName />
      </header>
   </SOAP:Header>
   <SOAP:Body>
      <jms xmlns="xmlapi_1.0">
         <keepAliveEvent />
      </jms>
   </SOAP:Body>
</SOAP:Envelope>
```

AlarmStatusChangeEvent

Code 4-5 shows sample event output for the AlarmStatusChangeEvent class.

Code 4-5: Sample AlarmStatusChangeEvent output

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP:Header>
      <header xmlns="xmlapi_1.0">
         <MTOSI_osTime>1138396703631</MTOSI_osTime>
         <ALA_clientId />
         <MTOSI_objectType>AlarmStatusChange</MTOSI_objectType>
         <MTOSI_NTType>NT_ATTRIBUTE_VALUE_CHANGE</MTOSI_NTType>
         <ALA_category>FAULT</ALA_category>
         <ALA_allomorphic>TiEvent</ALA_allomorphic>
         <MTOSI_objectName>svc-mgr:service-3:10.1.1.143</MTOSI_objectName>
      </header>
   </SOAP:Header>
   <SOAP:Body>
      <jms xmlns="xmlapi_1.0">
         <alarmStatusChangeEvent>
            <objectFullName>svc-mgr:service-3:10.1.1.143</objectFullName>
            <attribute>
               <attributeName>alarmStatus</attributeName>
               <value>2</value>
            </attribute>
         </alarmStatusChangeEvent>
      </jms>
```

```
   </SOAP:Body>
</SOAP:Envelope>
```

AttributeValueChangeEvent

Code 4-6 shows sample event output for the AttributeValueChangeEvent class.

For more information about the attributes of specific classes, see the *5620 SAM-O XML Reference*. For attributes that are enumerations, the integer value is used for AttributeValueChangeEvents, while the string value is used in ObjectCreationEvents.

Code 4-6: Sample AttributeValueChange output

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP:Header>
      <header xmlns="xmlapi_1.0">
         <MTOSI_osTime>1138123820148</MTOSI_osTime>
         <ALA_clientId />
         <MTOSI_objectType>netw.NetworkElement</MTOSI_objectType>
         <MTOSI_NTType>NT_ATTRIBUTE_VALUE_CHANGE</MTOSI_NTType>
         <ALA_category>GENERAL</ALA_category>
         <ALA_allomorphic>netw.NetworkElement</ALA_allomorphic>
         <MTOSI_objectName>network:10.1.186.218</MTOSI_objectName>
      </header>
   </SOAP:Header>
   <SOAP:Body>
      <jms xmlns="xmlapi_1.0">
         <attributeValueChangeEvent>
            <objectFullName>network:10.1.186.218</objectFullName>
            <attribute>
               <attributeName>resyncStatus</attributeName>
               <newValue>
                  <int>2</int>
               </newValue>
               <oldValue>
                  <int>3</int>
               </oldValue>
            </attribute>
            <attribute>
               <attributeName>lastTimeResyncStarted</attributeName>
               <newValue>
                  <long>1138123819632</long>
               </newValue>
               <oldValue>
                  <long>1138123519629</long>
               </oldValue>
            </attribute>
         </attributeValueChangeEvent>
      </jms>
   </SOAP:Body>
</SOAP:Envelope>
```

ObjectCreationEvent

Code 4-7 shows sample event output for the ObjectCreationEvent class.

Code 4-7: Sample ObjectCreationEvent output

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP:Header>
      <header xmlns="xmlapi_1.0">
<MTOSI_osTime>1138128165344</MTOSI_osTime>
         <ALA_clientId />
```

```
        <MTOSI_objectType>aclfilter.MacFilterEntry</MTOSI_objectType>
        <MTOSI_NTType>NT_OBJECTCREATION</MTOSI_NTType>
        <ALA_category>GENERAL</ALA_category>
        <ALA_allomorphic>aclfilter.MacFilterEntry</ALA_allomorphic>
        <MTOSI_objectName>MAC Filter:3:entry-1</MTOSI_objectName>
      </header>
   </SOAP:Header>
   <SOAP:Body>
      <jms xmlns="xmlapi_1.0">
         <objectCreationEvent>
            <aclfilter.MacFilterEntry>
               <frameType>unspecified</frameType>
               <sourceMacAddress>10-00-00-00-00-10</sourceMacAddress>
               <sourceMacAddressMask>00-00-00-00-00-00</sourceMacAddressMask>
               <destinationMacAddress>10-00-00-00-00-10</destinationMacAddress>
               <destinationMacAddressMask>00-00-00-00-00-00</destinationMacAddressMask>
               <dot1pValue>notSet</dot1pValue>
               <dot1pMask>unspecified</dot1pMask>
               <ethernetType>-1</ethernetType>
               <dsap>-1</dsap>
               <dsapMask>-1</dsapMask>
               <ssap>-1</ssap>
               <ssapMask>-1</ssapMask>
               <snapPid>-1</snapPid>
               <snapOui>off</snapOui>
               <action>default</action>
               <logId>0</logId>
               <administrativeState>notInService</administrativeState>
               <containingPolicyDisplayedName>ACL Mac Filter 2</containingPolicyDisplayedName>
               <containingPolicyId>3</containingPolicyId>
               <policyType>macAcl</policyType>
               <isLocal>false</isLocal>
               <siteId>0.0.0.0</siteId>
               <siteName>N/A</siteName>
               <displayedName>Filter 10-00-00-00-00-10</displayedName>
               <description>Filter Entry # 1</description>
               <id>1</id>
               <globalPolicy>N/A</globalPolicy>
               <deploymentState>0</deploymentState>
               <objectFullName>MAC Filter:3:entry-1</objectFullName>
               <name>Filter 10-00-00-00-00-10</name>
               <selfAlarmed>false</selfAlarmed>
               <children-Set />
            </aclfilter.MacFilterEntry>
         </objectCreationEvent>
      </jms>
   </SOAP:Body>
</SOAP:Envelope>
```


ObjectDeletionEvent

Code 4-8 shows sample event output for the ObjectDeletionEvent class.

Code 4-8: Sample ObjectDeletionEvent output


```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP:Header>
    <header xmlns="xmlapi_1.0">
      <eventName>ObjectDeletion</eventName>
      <MTOSI_osTime>1224518670453</MTOSI_osTime>
      <ALA_clientId/>
      <MTOSI_NTType>NT_OBJECTDELETION</MTOSI_NTType>
      <MTOSI_objectType>equipment.PhysicalPort</MTOSI_objectType>
      <ALA_category>EQUIPMENT</ALA_category>
      <ALA_isVessel>false</ALA_isVessel>
      <ALA_allomorphic>equipment.PhysicalPort</ALA_allomorphic>
      <ALA_eventName>ObjectDeletion</ALA_eventName>
```

```
      <ALA_span>:0:</ALA_span>
      <MTOSI_objectName>network:15.1.1.89:shelf-1:cardSlot-1:card:daughterCa
rdSlot-1:daughterCard:port-53</MTOSI_objectName>
      <ALA_OLC>2</ALA_OLC>
    </header>
  </SOAP:Header>
  <SOAP:Body>
    <jms xmlns="xmlapi_1.0">
      <objectDeletionEvent>
        <objectFullName>network:15.1.1.89:shelf-1:cardSlot-1:card:daughterCa
rdSlot-1:daughterCard:port-53</objectFullName>
      </objectDeletionEvent>
    </jms>
  </SOAP:Body>
</SOAP:Envelope>
```

DBActivityEvent

Code 4-9 shows sample event output for the DBActivityEvent class.

Code 4-9: Sample DBActivityEvent output

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP:Header>
      <header xmlns="xmlapi_1.0">
         <MTOSI_osTime>1138395709210</MTOSI_osTime>
         <ALA_clientId />
         <MTOSI_objectType>DBActivityEvent</MTOSI_objectType>
         <MTOSI_NTType>NT_STATE_CHANGE</MTOSI_NTType>
         <ALA_category>DATABASE</ALA_category>
         <ALA_allomorphic />
         <MTOSI_objectName />
      </header>
   </SOAP:Header>
   <SOAP:Body>
      <jms xmlns="xmlapi_1.0">
         <dbActivityEvent>
            <state>failoverEnd</state>
         </dbActivityEvent>
      </jms>
   </SOAP:Body>
</SOAP:Envelope>
```

DBProxyStateChangeEvent

Code 4-10 shows sample event output for theDBProxyStateChangeEvent class.

Code 4-10: Sample DBProxyStateChangeEvent output

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP:Header>
      <header xmlns="xmlapi_1.0">
         <MTOSI_osTime>1138395709210</MTOSI_osTime>
         <ALA_clientId />
         <MTOSI_objectType>DBProxyStateChangeEvent</MTOSI_objectType>
         <MTOSI_NTType>ALA_OTHER</MTOSI_NTType>
         <ALA_category>DATABASE</ALA_category>
         <ALA_allomorphic />
         <MTOSI_objectName />
      </header>
   </SOAP:Header>
   <SOAP:Body>
      <jms xmlns="xmlapi_1.0">
         <dbProxyStateChangeEvent>
            <state>databaseProxyPrimaryUp</state>
```

```
        </dbProxyStateChangeEvent>
      </jms>
   </SOAP:Body>
</SOAP:Envelope>
```

DBErrorEvent

Code 4-11 shows sample event output for the DBErrorEvent class.

Code 4-11: Sample DBErrorEvent output

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP:Header>
      <header xmlns="xmlapi_1.0">
         <MTOSI_osTime>1138395709210</MTOSI_osTime>
         <ALA_clientId />
         <MTOSI_objectType>DBErrorEvent</MTOSI_objectType>
         <MTOSI_NTType>NT_STATE_CHANGE</MTOSI_NTType>
         <ALA_category>DATABASE</ALA_category>
         <ALA_allomorphic />
         <MTOSI_objectName />
      </header>
   </SOAP:Header>
   <SOAP:Body>
      <jms xmlns="xmlapi_1.0">
         <dbErrorEvent>
            <state>clear</state>
            <error />
         </dbErrorEvent>
      </jms>
   </SOAP:Body>
</SOAP:Envelope>
```

DBConnectionStateChangeEvent

Code 4-12 shows sample event output for the DBConnectionStateChangeEvent class.

Code 4-12: Sample DBConnectionStateChangeEvent output

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP:Header>
      <header xmlns="xmlapi_1.0">
         <MTOSI_osTime>1138395709210</MTOSI_osTime>
         <ALA_clientId />
         <MTOSI_objectType>DBConnectionStateChangeEvent</MTOSI_objectType>
         <MTOSI_NTType>NT_STATE_CHANGE</MTOSI_NTType>
         <ALA_category>DATABASE</ALA_category>
         <ALA_allomorphic />
         <MTOSI_objectName />
      </header>
   </SOAP:Header>
   <SOAP:Body>
      <jms xmlns="xmlapi_1.0">
         <dbConnectionStateChangeEvent>
            <state>connectionUp</state>
         </dbConnectionStateChangeEvent>
      </jms>
   </SOAP:Body>
</SOAP:Envelope>
```

RelationshipChangeEvent

Code 4-13 shows sample event output for the RelationshipChangeEvent class.

Code 4-13: Sample RelationshipChangeEvent output

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP:Header>
      <header xmlns="xmlapi_1.0">
         <MTOSI_osTime>1138386054650</MTOSI_osTime>
         <ALA_clientId />
         <MTOSI_objectType>epipe.Epipe</MTOSI_objectType>
         <MTOSI_NTType>ALA_RELATIONSHIPCHANGE</MTOSI_NTType>
         <ALA_category>SERVICE</ALA_category>
         <ALA_allomorphic />
         <MTOSI_objectName>svc-mgr:service-3</MTOSI_objectName>
      </header>
   </SOAP:Header>
   <SOAP:Body>
      <jms xmlns="xmlapi_1.0">
         <relationshipChangeEvent>
            <objectFullName>svc-mgr:service-3</objectFullName>
            <relationship>
               <changeType>added</changeType>
               <fromObjectName>svc-mgr:service-3</fromObjectName>
               <fromObjectClass>epipe.Epipe</fromObjectClass>
               <toObjectName>subscriber:1</toObjectName>
               <toObjectClass>subscr.Subscriber</toObjectClass>
            </relationship>
         </relationshipChangeEvent>
      </jms>
   </SOAP:Body>
</SOAP:Envelope>
```

FileAvailableEvent

Code 4-14 shows sample event output for the FileAvailableEvent class.

Code 4-14: Sample FileAvailableEvent output

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP:Header>
      <header xmlns="xmlapi_1.0">
         <MTOSI_osTime>1138129016027</MTOSI_osTime>
         <ALA_clientId />
         <MTOSI_objectType>FileAvailableEvent</MTOSI_objectType>
         <MTOSI_NTType>ALA_OTHER</MTOSI_NTType>
         <ALA_category>GENERAL</ALA_category>
         <ALA_allomorphic />
         <MTOSI_objectName />
      </header>
   </SOAP:Header>
   <SOAP:Body>
      <jms xmlns="xmlapi_1.0">
         <fileAvailableEvent>
            <fileName>equipmentPhysicalPort.xml</fileName>
         </fileAvailableEvent>
      </jms>
   </SOAP:Body>
</SOAP:Envelope>
```

TerminateClientSessionEvent

Code 4-15 shows sample event output for the TerminateClientSessionEvent class.

Note — The JMS message terminateClientSessionEvent is broadcast to all sessions.

Code 4-15: Sample TerminateClientSession output

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP:Header>
      <header xmlns="xmlapi_1.0">
         <MTOSI_osTime>1138395709210</MTOSI_osTime>
         <ALA_clientId />
         <MTOSI_objectType>TerminateClientSession</MTOSI_objectType>
         <MTOSI_NTType>NT_STATE_CHANGE</MTOSI_NTType>
         <ALA_category>GENERAL</ALA_category>
         <ALA_allomorphic />
         <MTOSI_objectName />
      </header>
   </SOAP:Header>
   <SOAP:Body>
      <jms xmlns="xmlapi_1.0">
         <terminateClientSessionEvent>
            <clientId>ossi@1</clientId>
         </terminateClientSessionEvent>
      </jms>
   </SOAP:Body>
</SOAP:Envelope>
```

DeployerEvent

Code 4-16 shows sample event output for the DeployerEvent class.

Code 4-16: Sample DeployerEvent output

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP:Header>
   <header xmlns="xmlapi_1.0">
      <MTOSI_osTime>1165863370530</MTOSI_osTime>
      <ALA_clientId />
      <MTOSI_objectType>DeployerEvent</MTOSI_objectType>
      <MTOSI_NTType>ALA_OTHER</MTOSI_NTType>
      <ALA_category>GENERAL</ALA_category>
      <ALA_allomorphic />
      <MTOSI_objectName />
    </header>
 </SOAP:Header>
<SOAP:Body>
<jms xmlns="xmlapi_1.0">
<deployerEvent>
  <requestId>XmlApiClient:3</requestId>
 <successList>
    <deployerId>Default.DeployerBank:depl-486</deployerId>
 </successList>
 <failedList>
    <deployerId>Default.DeployerBank:depl-487</deployerId>
 </failedList>
</deployerEvent>
</jms>
</SOAP:Body>
</SOAP:Envelope>
```

StatsEvent

Code 4-17 shows sample event output for the StatsEvent class.

Code 4-17: Sample StatsEvent output

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP:Header>
      <header xmlns="xmlapi_1.0">
         <MTOSI_osTime>1170864259082</MTOSI_osTime>
         <ALA_clientId />
         <MTOSI_objectType>StatsEvent</MTOSI_objectType>
         <MTOSI_NTType>ALA_OTHER</MTOSI_NTType>
         <ALA_category>STATISTICS</ALA_category>
         <ALA_allomorphic />
         <MTOSI_objectName />
      </header>
   </SOAP:Header>
   <SOAP:Body>
      <jms xmlns="xmlapi_1.0">
         <statsEvent>
            <state>end</state>
            <networkElement>10.1.200.71</networkElement>
            <statsType>accounting</statsType>
            <time>1170864259082</time>
         </statsEvent>
      </jms>
   </SOAP:Body>
</SOAP:Envelope>
```

ScriptExecutionEvent

Code 4-18 shows sample event output for the ScriptExecutionEvent class.

Code 4-18: Sample ScriptExecutionEvent output

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP:Header>
<header xmlns="xmlapi_1.0">
         <MTOSI_osTime>1173717919050</MTOSI_osTime>
<ALA_clientId>JMSTEST@1</ALA_clientId>
<MTOSI_objectType>ScriptExecutionEvent</MTOSI_objectType>
<MTOSI_NTType>ALA_OTHER</MTOSI_NTType>
<ALA_category>GENERAL</ALA_category>
<ALA_allomorphic />
 <MTOSI_objectName />
 </header>
</SOAP:Header>
<SOAP:Body>
   <jms xmlns="xmlapi_1.0">
 <scriptExecutionEvent>
<targetScriptResultFullName>script-manager:script-4:target-script-6:target-s
cript-result-1173467653236</targetScriptResultFullName>
<fileName>scriptResult/target-script-6_1173467653236.txt </fileName>
 <status>Unknown</status>   <errorMessage>Trying to connect to an already
connected session.
    </errorMessage>
   <failedCommands />
</scriptExecutionEvent>
    </jms>
 </SOAP:Body>
</SOAP:Envelope>
```

ExceptionEventXMLFormat

Code 4-19 shows sample event output for the ExceptionEventXMLFormat class.

Code 4-19: Sample ExceptionEvent output

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP:Header>
      <header xmlns="xmlapi_1.0">
         <MTOSI_osTime>1174926836987</MTOSI_osTime>
         <ALA_clientId />
         <MTOSI_objectType>aengr.Policy</MTOSI_objectType>
         <MTOSI_NTType>ALA_OTHER</MTOSI_NTType>
         <ALA_category>GENERAL</ALA_category>
<ALA_allomorphic />
         <MTOSI_objectName>Access Egress:2100</MTOSI_objectName>
      </header>
   </SOAP:Header>
   <SOAP:Body>
      <jms xmlns="xmlapi_1.0">
         <ExceptionEventXMLFormat>
            <className>aengr.Policy</className>
            <operation>distribute on node 10.168.1.91</operation>
            <objectName>Access Egress:2100</objectName>
            <requestId>client1:0</requestId>
            <errorMessage>[ app: autoconfigChild ] [ class: aengr.Policy ] [ instance:
network:10.168.1.91:Access Egress:2100 ] [ descr: invalid action bitmask: 3 : Object deletion
is in progress, the object cannot be configured :Parent = network:10.168.1.91:Access
Egress:2100: child = queue-1 ]</errorMessage>
         </ExceptionEventXMLFormat>
      </jms>
   </SOAP:Body>
</SOAP:Envelope>
```

LogFileAvailableEvent

Code 4-20 shows sample event output for the LogFileAvailableEvent class.

Code 4-20: Sample LogFileAvailableEvent output

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP:Header>
      <header xmlns="xmlapi_1.0">
         <eventName>LogFileAvailable</eventName>
         <MTOSI_osTime>1184770068031</MTOSI_osTime>
         <ALA_clientId>JMS_Client_ID</ALA_clientId>
         <MTOSI_objectType>LogFileAvailableEvent</MTOSI_objectType>
         <MTOSI_NTType>ALA_OTHER</MTOSI_NTType>
         <ALA_category>GENERAL</ALA_category>
         <ALA_allomorphic />
         <MTOSI_objectName />
      </header>
   </SOAP:Header>
   <SOAP:Body>
      <jms xmlns="xmlapi_1.0">
         <logFileAvailableEvent>
<fileName>client_directory\NE_IP_address_1184770068031.xml</fileName>
            <serverIpAddress>server_IP_address</serverIpAddress>
            <routerId>NE_IP_address</routerId >
         </logFileAvailableEvent>
      </jms>
   </SOAP:Body>
</SOAP:Envelope>
```

managedRouteEvent

Code 4-21 shows sample event output for the managedRouteEvent class.

Code 4-21: Sample managedRouteEvent output

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP:Header>
<header xmlns="xmlapi_1.0">
<eventName>ManagedRouteEvent</eventName>
<ALA_category>CPAM</ALA_category>
<ALA_OLC>0</ALA_OLC>
<ALA_application>1</ALA_application>
<ALA_routeManager>tpgy-mgr:application-diane</ALA_routeManager>
<ALA_isVessel>false</ALA_isVessel>
<ALA_allomorphic>Hermes</ALA_allomorphic>
<MTOSI_osTime>1250787917539</MTOSI_osTime>
<MTOSI_NTType>ALA_MANAGEDROUTE</MTOSI_NTType>
<MTOSI_objectName/>
<ALA_clientId/>
<MTOSI_objectType>ManagedRouteEvent</MTOSI_objectType>
<ALA_eventName>ManagedRouteEvent</ALA_eventName>
</header>
</SOAP:Header>
<SOAP:Body>
<jms xmlns="xmlapi_1.0">
<managedRouteEvent>
<item>
<key>
<topology.RouteKey>
<sourceType>ipv4</sourceType>
<source>38.120.202.175</source>
<sourceLen>32</sourceLen>
<destType>ipv4</destType>
<dest>38.120.202.171</dest>
<destLen>32</destLen>
</topology.RouteKey>
</key>
<value>
<topology.RouteResult>
<errorCode>mspfCouldNotDeterminAs</errorCode>
<errorMessage>Could not determine administrative domain.</errorMessage>
<type>standard</type>
<vertices/>
<detailedErrors>
<topology.DetailedError>
<errorCode>mspfCouldNotDeterminAs</errorCode>
<message>Could not determine administrative domain.</message>
</topology.DetailedError>
</detailedErrors>
</topology.RouteResult>
</value>
</item>
</managedRouteEvent>
</jms>
</SOAP:Body>
</SOAP:Envelope>
```

IncrementalRequestEvent

Code 4-22 shows sample event output for the IncrementalRequestEvent class.

Code 4-22: Sample IncrementalRequestEvent output

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP:Header>
```

```
    <header xmlns="xmlapi_1.0">
      <eventName>IncrementalRequestEvent</eventName>
      <ALA_category>GENERAL</ALA_category>
      <ALA_OLC>0</ALA_OLC>
      <ALA_isVessel>false</ALA_isVessel>
      <ALA_allomorphic></ALA_allomorphic>
      <MTOSI_osTime>1276545012449</MTOSI_osTime>
      <MTOSI_NTType>ALA_OTHER</MTOSI_NTType>
      <MTOSI_objectName></MTOSI_objectName>
      <ALA_clientId></ALA_clientId>
      <MTOSI_objectType>IncrementalRequestEvent</MTOSI_objectType>
      <ALA_eventName>IncrementalRequestEvent</ALA_eventName>
    </header>
  </SOAP:Header>
  <SOAP:Body>
    <jms xmlns="xmlapi_1.0">
      <IncrementalRequestEvent>
        <requestId>4</requestId>
        <incrementalAction>RE_EVAL_MSAPS_ON_MSAP_POLICY</incrementalAction>
        <serviceType>default</serviceType>
        <status>finished</status>
        <originalParam>
          <string>msapPolicy:msappolicy</string>
        </originalParam>
        <payload>
          <boolean>true</boolean>
        </payload>
      </IncrementalRequestEvent>
    </jms>
  </SOAP:Body>
</SOAP:Envelope>
<find xmlns="xmlapi_1.0">
  <fullClassName>equipment.PhysicalPort</fullClassName>
  <filter>
    <equal name="siteName" value="sim202_93"/>
  </filter>
  <resultFilter>
    <attribute>objectFullName</attribute>
    <attribute>operationalState</attribute>
    <attribute>administrativeState</attribute>
    <attribute>mode</attribute>
    <children></children>
  </resultFilter>
</find>
</SOAP:Body>
```


4.5 JMS and redundancy

JMS subscribers in a redundant system must be notified of switchover events and disconnections of clients. Each time a JMS subscriber connects to a topic, a message is sent containing the current system information. The client ID of the connected client is included in the JMS header under the property ALA_clientId. OSS developers must verify the client ID to ensure that the message is specific to the client, as this message is sent to all clients on that topic. This event is a StateChangeEvent with a state of System Information (state value of 8). Table 4-8 lists additional objects that are attached in the form of a map.

Table 4-8 Additional map objects

Property name	Description
sysType	Identifies the system type as a string; redundant or standalone

(1 of 2)

Property name	Description
sysStandbyIp	IP address of the standby server as a string; only in redundant systems
sysPrimaryIp	IP address of the primary server as a string
sysStartTime	Last time a server restarted in long format (milliseconds)
jmsStartTime	Last time a JMS server restarted in long format (milliseconds)

(2 of 2)

Storing attribute information for the subscriber can help identify whether:

- a switchover event occurred the next time the subscriber connects to the client (change in sysPrimaryIp)
- the server was restarted (change in sysStartTime)

See section "StateChangeEvent" for more information.

4.6 Connection monitoring and error recovery

There are two scenarios for which an OSS must monitor and, if necessary, recover from:

- loss of connectivity
- loss of events

These scenarios can occur separately or together. For example, when using a durable subscription, a loss of connectivity may occur without a loss of events. Loss of events can also occur when the message system is under heavy load, without connectivity loss, and is flagged by a jmsMissedEvents message. Finally, any loss of connection which also involves a subscription being removed—which can occur in several situations, including but not limited to an activity switch or a server restart.

The manner in which an OSS manages these scenarios depends their specific needs and the expectations of their customers. For example, for some applications, a loss of connectivity should always result in the OSS application reconnecting to the 5620 SAM. In other situations, the desired behavior may depend on the cause of the connectivity loss—for example, if a 5620 SAM administrator intentionally disconnects an OSS, it may not be desirable to immediately reconnect.

How to recover from lost events also depends on the OSS. If an OSS uses JMS events to maintain a local information store that mirrors some data set in the 5620 SAM—such as a network inventory or a list of current alarms—then a loss of events results in the OSS being out of sync with the 5620 SAM.

The following scenarios could be used to determine whether the OSS needs to resync the database of inventory and alarm information:

- If a durable OSS disconnects but remains subscribed to the 5620 SAM, on reconnect if a JmsMissedEvent message is not received and the sysStartTime is not changed, it is not necessary to do a resync of inventory and alarm information because all events will be queued on the 5620 SAM server waiting for reconnection with the OSS.
- If a disconnected durable subscription is removed from an active 5620 SAM server, a JmsMissedEvents message will be generated when the client reconnects using the same client ID. This indicates to the OSS that a resync of inventory and alarm information needs to be performed because events have been missed.

After the OSS detects it is out of sync, possible recovery scenarios could include:

- immediately resynchronizing with the 5620 SAM, such as by retrieving inventory information or the latest alarm list through the XML API
- notifying the OSS administrator of the problem, and allowing the administrator to select a recovery approach—for example, an immediate resync or a scheduled resync for a later time

When you implement a recovery procedure, you must consider that events will continue to occur in the network while an OSS is busy resynchronizing or populating its database for the first time. For this reason, an OSS must to process events while they are resynchronizing, and may need to recover from error conditions that require them to reconnect or restart the resync.

Procedure 4-1 To resync an OSS database with the 5620 SAM

1 Establish a JMS subscription, listen for events, and buffer events locally. Alcatel-Lucent recommends that the OSS maintain two local buffer queues:

   - one queue for critical system-related events—such as the KeepAliveEvent, SystemInfoEvent, or JmsMissedEvent—which the OSS can use to monitor the state of the JMS connection.
   - one queue for inventory or alarm events in which the OSS is interested.

Note — The OSS must not begin processing the events from the second buffer until step 3 is successfully completed.

2 Begin retrieving inventory information or the latest alarm list.

Note — While inventory is being retrieved, the OSS must monitor the first buffer queue, as described in step 1. If the OSS detects connectivity loss with the 5620 SAM, or a loss of JMS events, the OSS should terminate the current inventory retrieval process, if possible. The internal buffer queues should be cleared. Perform recovery procedures to restore JMS connectivity, if necessary, then repeat step 2.

3 After the inventory retrieval completes successfully, populate the OSS database with the buffered events from step 1.

JMS exception

You must use JMS exceptions if you have a JMS connection to a 5620 SAM server. JMS internally monitors client connections to the JMS server and throws an exception if a connection between the JMS server and a client is lost. You must implement the javax.jms.ExceptionListener interface on the JMS client to enable the monitoring of exceptions. The interface contains a call-back method that is invoked for all exceptions. A typical implementation of the call-back method attempts to reconnect and possibly take another action, such as generating an event.

Monitoring for incoming events

In addition to handling exceptions, an OSS must monitor incoming events to ensure that the JMS connection is active. A KeepAliveEvent is published at approximately 30-second intervals to each of the JMS topics even when no other messages are sent.

KeepAliveEvents may be received ahead of other event types.

If no events are received within a reasonable time period, you must investigate the status of the 5620 SAM server.

Events may be subject to latency issues related to network throughput, the volume of events being processed on a 5620 SAM mainserver, and the processing speed of the client. If the client does not receive any expected events within a reasonable period, such as 5m, you must investigate the cause of the excessive latency. Potential problems can be related to the 5620 SAM main server status. See the *5620 SAM Troubleshooting Guide* for information on how to check the server status, or contact your Alcatel-Lucent OIPS technical-support representative for information about how to diagnose the problem and implement corrective actions.

Note 1 — You must define the JMS filter to include keep alive messages. See section "Message filters" for more information.

Note 2 — To prevent the loss of keep alive messages, you must ensure that the system clocks of the OSS clients and 5620 SAM main server are synchronized.

5620 SAM-O session termination

You can use the 5620 SAM GUI client to close and remove a durable subscription when you no longer require a durable client. See the *5620 SAM User Guide* for more information about removing durable subscriptions.

The 5620 SAM-O uses the following JMS event to notify OSS client applications of a session termination:

```
TerminateClientSession
```

The TerminateClientSession event indicates that a client JMS session is about to be closed. The client must clean up the disconnected session when the message is received. Additional session termination behavior is dependent on the requirements of your OSS client application. For example, you can also configure the requirement to close the OSS client application.

Missed events

Both durable and non-durable subscriptions can be used by OSS clients that cannot tolerate losing events without taking corrective action. However, in certain situations, events can be missed and subscribers are notified with a JmsMissedEvents message.

JMS messages are queued in the 5620 SAM until they are acknowledged by the JMS consumer, or 5620 SAM-O client. If the JMS message queue overflows, the following occurs for both durable and non-durable JMS subscriptions:

- Connected non-durable subscribers
  - the 5620 SAM stops queuing messages until there is capacity in the queues
  - a JmsMissedEvents message may or may not be sent depending on the internal cause of the overflow and whether there is capacity in the queues.
- Connected durable subscribers
  - queued messages are removed
  - a JmsMissedEvents message is added to the queue. See code 4-23 for more information.
  - new events continue to be queued or sent
- Non-connected non-durable subscribers
  - Not applicable since a non-connected non-durable subscriber becomes unsubscribed as soon as it disconnects from the 5620 SAM server.
- Non-connected durable subscribers
  - queued messages are removed
  - the subscriber is unsubscribed
  - no new events are queued

The following are examples for which messages may be missed:

- the client is slow, the message rate is high, or there is high latency in the network
- disconnected clients with active subscriptions, resulting in messages being queued until the client reconnects

Table 4-9 describes the alarms that are related to JMS clients and messages.

Table 4-9 JMS-specific alarms

Alarm	Description
JMSDurableClientReset	Raised when a durable JMS client is reset as the result of a JMS server restart or activity switch
JMSClientMessagesRemoved	Raised when a JMS client has messages removed after exceeding the configured message limit. This applies to OSS durable subscribers only.
JMSDurableClientUnsubscribed	Raised when a durable JMS client is automatically unsubscribed. This occurs when a disconnected durable client exceeds the configured message limit.

JmsMissedEvents message

The 5620 SAM sends a JmsMissedEvents message to indicate that events have been lost, allowing subscribers to detect missed events. The JmsMissedEvents message is a StateChange Event, as shown in Code 4-23.

Code 4-23: JmsMissedEvent

```
<SOAP:Envelope
xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP:Header>
      <header xmlns="xmlapi_1.0">
        <eventName>JmsMissedEvents</eventName>
        <MTOSI_osTime>1222891102168</MTOSI_osTime>
        <ALA_clientId>oss_client@1</ALA_clientId>
        <MTOSI_NTType>ALA_OTHER</MTOSI_NTType>
        <MTOSI_objectType>StateChangeEvent</MTOSI_objectType>
        <ALA_category>GENERAL</ALA_category>
        <ALA_isVessel>false</ALA_isVessel>
        <ALA_allomorphic/>
        <ALA_eventName>JmsMissedEvents</ALA_eventName>
        <MTOSI_objectName/>
        <ALA_OLC>0</ALA_OLC>
      </header>
    </SOAP:Header>
    <SOAP:Body>
      <jms xmlns="xmlapi_1.0">
        <stateChangeEvent>
          <eventName>JmsMissedEvents</eventName>
          <state>jmsMissedEvents</state>
        </stateChangeEvent>
      </jms>
    </SOAP:Body>
  </SOAP:Envelope>
```

Recovery from missed events

Clients must be able to recognize when events are missed and take recovery action that is appropriate for their application. For example, in an OSS application for which events are being used to maintain inventory information, the recovery action could include:

- an immediate resync of inventory information
- notification being sent to an administrator, allowing them to perform an immediate resync or schedule a future resync

Although an OSS application must have measures in place to recover from missed events, the OSS application can implement prevention methods. The following are examples of ways an OSS application can prevent missed events:

- use restrictive filtering wherever possible. See section "Message filters" for more information.
- increase OSS efficiency in processing events
- minimize the amount of time that an OSS is disconnected from the 5620 SAM, but still subscribed (where events are being queued but not processed)
- queue messages internally to facilitate the handling of temporary message bursts
- use DUPS_OK_ACKNOWLEDGE acknowledgement mode to facilitate the handling of network latency. See section "Acknowledgement modes" for more information.

4.7 Managing and monitoring JMS sessions

The following sections describe how you can manage and monitor JMS connections.

Monitoring sessions

You can manage and monitor the state of JMS connections and error conditions using the 5620 SAM GUI.

You can view the status of current sessions and, if necessary, close or remove sessions. When you close a non-durable subscription, the client is disconnected—after sending a TerminateClientSession event—and the subscription is removed. When you close a durable subscription, a TerminateClientSession event is sent and the client is disconnected. However, the subscription is not removed. The session continues to be visible in the session list and the subscription remains active so events are being queued. However, the status indicates that the session is disconnected. You can remove a disconnected session. See procedure 4-2 for more information.

You can also use the stopOssSession method in the security.SamSessionManager class over HTTP to close a JMS session. You must be an admin user or the user that initiated the session to close the JMS session using this method.

The 5620 SAM raises alarms for events that impact JMS sessions. See Table 4-9 in section 4.2 for more information about the alarms that are related to JMS clients and messages.

JMS logging

The 5620 SAM JMS server logs all connections to JMS, including the JMS filter information. The 5620 SAM JMS server records the connection information in the jmsServer.log file. To obtain the log file, go to the 5620 SAM JMS server installation directory and navigate to the /nms/log/jmsserver directory. You can then open the log file using the 5620 SAM LogViewer utility. The log file is backed up when it reaches a configured value. You can view the backup files in the same directory as the original file. See "Log file for XML requests, responses, and exceptions" in chapter 10 for more information. See the *5620 SAM Troubleshooting Guide* for information about using the LogViewer utility.

Procedure 4-2 To manage a 5620 SAM-O JMS client session or remove a durable subscription

1. Using an account with an assigned security scope of command role, choose Administration→Security→5620 SAM User Security from the 5620 SAM main menu. The 5620 SAM User Security - Security Management (Edit) form opens.
2. Click on the Sessions tab button.
3. Specify a filter to create a filtered list 5620 SAM-O JMS client sessions and click on the Search button. The list of currently active client sessions opens.
4. Choose a JMS session from the list and click on the Close Sessions button to shut down the client session. The connection to the server is terminated when you close a durable JMS session, however, the subscription continues to store JMS messages. When you close a non-durable session, the session is also removed.
5. Perform one of the following steps.
   a. To remove a durable subscription and shut down the client session, choose a JMS session from the list and click on the Remove Session button. The server stops storing the JMS messages for the session.

      Note — When you remove a durable subscription, the OSS client can still attempt to connect to the 5620 SAM-O server. You can prevent an OSS client from attempting to connect by suspending the OSS user account. See the *5620 SAM User Guide* for more information.

   b. Go to step 6.
6. Validate the action.
7. Close the form.

4.8 JMS consumer creation

You can use procedure 4-3 to create a simple JMS consumer. You can find JMS sample code in the *Installation_directory*/nms/distribution/User_Documentation/5620_SAM-O_documentation/JMSTestExample directory, where *Installation_directory* is the client installation folder or directory.

Procedure 4-3 To compile and create a sample JMS client *.java file

Before you perform the following procedure, which is for a UNIX-based system, you need to understand the objects.

You need the samOSS.jar file for your classpath, which you can retrieve from the following locations:

- /integration/SAM_O/samOss.jar on the 5620 SAM installation DVD-ROM.
- Installed on the 5620 SAM server in the /opt/5620sam/server/nms/integration/SAM_O directory.
- Download from http://<*ip_address_SAM_server*>:8080/integration/samOss.jar. OSS clients can extract the jar file programmatically, or an administrator can download the file with a Web browser pointed at the 5620 SAM server.

Warning — The OSS client must use the correct the samOss.jar that corresponds with the installed 5620 SAM release. Using the incorrect samOss.jar can cause the 5620 SAM server or clients to become unstable. The 5620 SAM release information is in the JAR manifest file.

Note — The samOss.jar contains all of the classes that are needed to connect to the 5620 SAM (for use with running outside an application server or JMS server).

If the OSS client must connect to multiple versions of the 5620 SAM, the OSS must run a separate JVM with the samOss.jar that corresponds with each version of the 5620 SAM. 5620 SAM version information is included in the jar manifest file. Alternatively, the OSS client can use multiple class loader instances in the same JVM to access multiple versions of the samOss.jar.

The following jars are deprecated and will be removed in a future release. samOss.jar should be used to connect to the 5620 SAM server:

- samOssJBoss.jar
  Does not contain any application server classes, only specific 5620 SAM classes (for use when running within a JBoss application server).
- samOssAnyServer.jar
  Contains the same classes as the samOss.jar, in addition to necessary classes for communication with the JMS server (for use when running within another application server).

It is only necessary to include one jar specific to the OSS application needs. If you choose a jar other than the samOss.jar you must update your classpath accordingly.

1 Compile the *.java file. The following code is for a *.java file named JmsTest.java:

```
javac

-classpath .:samOss.jar JmsTest.java
```

You must know the path to the samOss.jar file on the server.

Note — Java 1.6.0_14 or later is required.

2 Run the compiled program:

```
java -classpath .:samOss.jar JmsTest -t <topic> -s <App server
IP:port>

{-r <HA App server IP:port>} -u <user> -p <password> -f "<filter>"
```

```
{-persistent} -c <uniqueid>
```

where
App Server IP is the IP address of the 5620 SAM server
port is the JMS port number configured during installation
uniqueid, *user*, and *password* are valid account information configured by the 5620 SAM
filter is a valid JMS filter. See section "Message filters" for more information.

Note 1 — If the 5620 SAM server is configured to use SSL, you must run JmsTest with the following command:

```
-Djavax.net.ssl.trustStore=<keystore_file>
```

where *keystore_file* is the fully qualified path to the 5620 SAM main server keystore file.

For example:

```
java -classpath.:samOss.jar
-Djavax.net.ssl.trustStore=<keystore_file> JmsTest
```

See the *5620 SAM User Guide* for more information about how to configure SSL.

Note 2 — To ensure that the 5620 SAM server reconnects after an activity switch for HA (High Availability) JMS connections, you must set the HA port for JMS connections to the JBoss High Availability JNDI service port value of 1100.

See the *5620 SAM Planning Guide* for more information on the list of ports required on the 5620 SAM server deployment.

4.9 Workflow to set up and operate a JMS connection

Use the following steps to set up and operate a JMS connection.

1. Ensure that you complete the workflow in section 3.5 to set up and operate the 5620 SAM-O.
2. Develop an understanding of the 5620 SAM-O JMS message interface. See section 4.1 for more information.
3. Determine the topic or topics that your application must monitor. See Table 4-1 for more information about the available topics.
4. Determine the events that your application must monitor. See section "Message filters" for more information about defining a message filter to include the specific messages required by the OSS application.
5. Develop a JMS client. See Procedure 4-3 for more information.
6. Open a session to the 5620 SAM-O server and monitor events and exceptions.

7 Subscribe to topics that are of interest to the application.

8 Manage the connection and any errors related to communication failure.

9 Close the session and remove the subscription if it is no longer needed.

4.10 JMS record and playback tool

You can use the JMS record and playback tool to record JMS messages coming from a 5620 SAM network by capturing and writing the JMS message flow to an XML file. The playback function then takes the messages in the XML file and plays them back to an OSS or the 5620 SAM.

Note 1 — The playback function is a JMS exercise and there is no interaction with the underlying 5620 SAM network or 5620 SAM database.

Note 2 — There is no memory management built into this tool. You must stop the recording of messages to avoid disk partition overflow.

Note 3 — You can modify the messages and delays by adding or moving attributes.

Note 4 — The delay is in milliseconds.

Note 5 — Only the syntax is validated. Invalid syntax prevents playback.

Procedure 4-4 To record, stop and playback JMS messages

Warning — The playback function may have an impact on the performance of the 5620 SAM server, 5620 SAM GUI, or the OSS client behavior.

1 Record JMS messages:

```
../5620sam/server/nms/bin/unsupported/jmsTools.bash

jmsTools.bashjmsRecorder start <FileName>
```

where <*FileName*> is the absolute path to a file that will be used to store the XML-formatted messages and delays.

2 Stop JMS messages:

```
jmsTools.bashjmsRecorder stop
```

3 Playback JMS messages:

```
jmsTools.bashjmsReplayer <FileName> <delay>
```

where
<*FileName*> is the absolute path to the file containing the messages
<delay> should be set to Yes to use all delays that are captured during JMS recording.

5 — XML requests

5.1 XML communication with the 5620 SAM server *5-2*

5.2 Monitoring the status of the 5620 SAM-O connection using XML API ping *5-5*

5.3 Workflow to set up and operate an HTTP XML request-response connection *5-7*

5.4 Viewing XML requests ignored by the server *5-7*

5.5 5620 SAM-O server-specific commands *5-8*

5.1 XML communication with the 5620 SAM server

The XML API is accessible using a servlet that runs on the 5620 SAM server. The 5620 SAM-O uses HTTP and HTTPS protocols to transport the XML requests between the client application and the server. See the *5620 SAM User Guide* for more information about how to configure HTTP or HTTPS access.

Warning — The maximum number of concurrent OSS connections is 10 or more and varies based on server hardware. See the *5620 SAM Planning Guide* for information about platform sizing.

In a 5620 SAM network with multiple OSS applications, Alcatel-Lucent recommends that each OSS application use only one HTTP connection for each XML/SOAP request. This helps to avoid having one OSS application consuming all connections.

If more than one HTTP connection is required, for example, for performance and scalability, you must make the number of HTTP connections configurable at run time (after installation), so that system integrators can manage the HTTP connections across the OSS applications that connect to the same 5620 SAM server. Contact your Alcatel-Lucent OIPS technical support representative for more information.

Note — You must use the HTTP POST method to send XML API requests.

PostXML

For the purpose of demonstrating the functionality for sending XML requests to the 5620 SAM server, this subsection uses the PostXML.java application as a simple example. The source code contains methods that post XML requests to a remote web server using the HTTP POST method.

Note — PostXml.java software was developed by voluntary contributions made by individuals on behalf of the Apache Software Foundation as part of the Apache Jakarta Project. Alcatel-Lucent does not support or endorse the postXML.java application, or any other third-party application. For a more robust solution, Alcatel-Lucent recommends developing code that is customized for the OSS application to generate XML and send it over an HTTP connection without creating files for each request. The HTTP connections should be managed directly by the OSS process.

The PostXML.java file includes the actual function that posts the XML code to the 5620 SAM server, and requires the following Java library files:

- commons-httpclient.jar
- commons-logging.jar
- commons-codec.jar

PostXML requires the same version of the files as the server. Table 5-1 lists the locations for Java library files on the 5620 SAM server.

Table 5-1 Location for Java library files

5620 SAM component	Location

server	`C:\<5620sam_home>\server\nms\jboss\lib\`
	or
	`/opt/<5620sam_home>/server/nms/jboss/lib`

PostXML is run from within a Java session that is configured with the appropriate class path setting to the 5620 SAM server. Code 5-1 shows a sample batch file configuration that could be used to initialize the Java session.

Code 5-1: PostXML batch file

```
@echo off
set CP=.;commons-httpclient.jar;jecho "classpath=%CP%"
java -classpath %CP% PostXML
http://<ip_address_sam_server>:8080/xmlapi/invoke %1
```

You can reference the PostXML batch file in your request to the 5620 SAM server. Code 5-2 shows a sample command to post a request to create a service.

Code 5-2: Example of how to post a request

```
call PostXML.bat createEpipe.xml
```

The 5620 SAM server responds to the request and uses the PrettyPrint.xsl stylesheet to format the display sent to the client command line interface. The stylesheet is available from your Alcatel-Lucent account or OIPS technical support representative. The response can contain the following information:

- notification of the successful or failed request
- basic information such as the fully distinguished name for the configured object
- all known information for the configured object

Procedure 5-1 To post an XML request to the 5620 SAM server

You must be a valid user on the 5620 SAM. User and password information is sent with each request to the 5620 SAM server.

Note 1 — You must use a user account with OSS Mgmt permission to access the 5620 SAM-O. See the *5620 SAM User Guide* for more information about the 5620 SAM user accounts and privileges.

Note 2 — This procedure uses the following components as examples.

- PostXML to post the XML requests to the 5620 SAM server. The utility is available from the website for the Apache Jakarta Project.
- PrettyPrint.xsl to format the response from the 5620 SAM server. The stylesheet is available from your Alcatel-Lucent account or OIPS technical support representative.

You can choose a different utility and stylesheet for your 5620 SAM-O implementation.

1 Specify the HTTP or HTTPS access address.

a For HTTP access, use:

```
http://server_name:port#/xmlapi/invoke
```

where
server_name is the host name or IP address of the 5620 SAM server
port# is a port on the 5620 SAM server configured for HTTP or HTTPS, as described in the *5620 SAM | 5650 CPAM Installation and Upgrade Guide*

b For HTTPS access, use:

```
https://server_name:port#/xmlapi/invoke
```

where
server_name is the host name or IP address of the 5620 SAM server
port# is a port on the 5620 SAM server configured for HTTP or HTTPS, as described in the *5620 SAM | 5650 CPAM Installation and Upgrade Guide*

The XML application servlet is running and available from the appropriate port. The SOAP-enabled XML can now be sent to the server. For more information about communication between the OSS and the server, contact your Alcatel-Lucent account or OIPS technical support representative.

2 Put the following files in the C:\<*5620sam_home*>\client\nms\ directory on the client station:

- PostXML.class
- PostXML.bat
- commons-httpclient.jar
- commons-logging.jar
- commons-codec.jar
- PrettyPrint.xsl

3 Create an XML request script that you want to post to the 5620 SAM server.

4 Put the XML request script in the same directory as the 5620 SAM-O client files. The 5620 SAM-O client files are typically in the following directory:

- on a Windows station, C:\5620sam\client\sam-o_client
- on a Solaris station, /opt/5620sam/client/sam-o_client

5 Create a script to post the request to the 5620 SAM main server. See Code 5-1 for an example script file.

6 Open a CLI session.

7 Post the request to the 5620 SAM server by entering the following command:

```
call PostXML.bat <xml_request_name>.xml
```

where <*xm_request_name*> is the name of your XML request

The 5620 SAM server responds with the request execution status and the requested information. See Table 10-2 for more information about the execution status that is provided by the HTTP response codes.

5.2 Monitoring the status of the 5620 SAM-O connection using XML API ping

You can use XML API ping to monitor the HTTP connection to the 5620 SAM web server.

Note — See chapter 4 for more information about using the following methods to monitor the status of the 5620 SAM-O connection:

- JMS exception to monitor client connections to the JMS server
- monitoring near-real-time events using JMS to monitor client connections to the JMS server

XML API ping

The XML API ping method is used to monitor the HTTP connection to the 5620 SAM web server.

Code 5-3 shows an XML API ping that you can use to test the ability of the OSS application to access the 5620 SAM web server. Information in the SOAP/XML request includes the:

- standard SOAP user and password encoding
- ping command to look for the release of XML on the server, in this case version 1.0

Caution — The following sample message is an example of the request format. Use the sample as a base to build your request. Ensure that you test your request before network deployment.

Code 5-3: XML interface check

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope>
    <SOAP:Header>
        <header xmlns="xmlapi_1.0">
            <security>
                <user>user name</user>
                <password>MD5-hashed user password</password>
            </security>
            <requestID>clientName@requestId</requestID>
        </header>
    </SOAP:Header>
    <SOAP:Body>
        <ping xmlns="xmlapi_1.0"/>
    </SOAP:Body>
</SOAP:Envelope>
```

Code 5-4 is an example of a successful response message to the ping request. The response indicates that the correct version 1.0 of XML on the server responded to the ping.

Code 5-4: XML interface check response

```
<SOAP:Envelope>
   <SOAP:Header>
      <header xmlns="xmlapi_1.0">
         <requestID>clientName@requestId</requestID>      </header>
   </SOAP:Header>
   <SOAP:Body>
      <pingResponse xmlns="xmlapi_1.0"/>
   </SOAP:Body>
</SOAP:Envelope>
```

An exception response to a failed ping could result in numerous types of return messages. For example:

- socket timeouts
- HTTP 404 errors
- connection exceptions

5.3 Workflow to set up and operate an HTTP XML request-response connection

Use the following steps to set up and operate the 5620 SAM-O.

1. Ensure that you complete the workflow in section 3.5 to set up and operate the 5620 SAM-O.
2. Develop the application that requests data from, or sends data to, the server and/or the database.
   - **i** Construct the request by specifying:
     - methods that define the executed function
     - package classes used that define the kinds of objects, and the values of the objects (parameters) requested
     - filters to limit or customize the scope of the request
     - information structure and data types
   - **ii** Send the request. See Procedure 5-1 for more information
3. Manage the responses from the server.
   - **a** View the error or success messages.
   - **b** Handle the retrieved information appropriately.
4. Close the session. See section 4.6 for more information about terminating the session.

5.4 Viewing XML requests ignored by the server

If an OSS application includes an element that is not in the schema in the XML request, the element is ignored and the OSSI processes the request. All XML requests with matching start and end tags are accepted by the OSSI, including spelling and case errors. The 5620 SAM-O logs a message to the serverLogFile.

Note — Enabling all logging for the XML API layer is only suitable for development environments.

Procedure 5-2 To enable server log file logging of ignored XML requests

Caution — The server log file should only be enabled on servers in a development environment. When you enable the server log on a server that is running in live network environment, server performance may be affected.

1 Open the nms-server.xml file on the 5620 SAM server. The file is located in the *install directory*/nms/config directory, where *install directory* is the installation directory.

2 Find the <Filter> element within the <SystemLog> element.

3 Add the following code directly after the line beginning with <!-- Don't change this section END -->

```
<include severity="debug" package=".*xmlapi.*"/>
```

Caution — Do not modify other nms-server.xml parameters. Modifying the file can seriously affect the network management and performance of the 5620 SAM.

4 Restart the 5620 SAM server. All XML requests not in the XML schema are now logged to the server log file.

5.5 5620 SAM-O server-specific commands

The 5620 SAM-O provides XML commands that allow you to specify completion of the following actions:

- obtain the local server time for the request
- identify the server software load

Obtaining the local server time for a request

You can include the <timeStamp xmlns="xmlapi_1.0"/> element in your request to obtain the server time for the request in milliseconds. Code 5-5 shows a sample of the <timeStamp xmlns="xmlapi_1.0"/> element and server-specific version commands used in a request. Code 5-6 shows a sample of the response to the request in Code 5-5.

Note — The server time for the request in the SOAP response header indicates when a request stream was opened. It does not indicate when a request completed.

You can insert multiple <timeStamp xmlns="xmlapi_1.0"/> elements for a SOAP message that contains multiple message requests. You can use the delta between the timestamp commands to determine the milliseconds required to execute the intermediate commands. See "Requests" in section 10.1 for more information about multiple requests.

Identifying the server software load

You can include the <version xmlns="xmlapi_1.0"/> element within the body of your XML request to determine the software load installed on the 5620 SAM server. The server returns the software load information in an XML response with the following format.

```
5620 SAM Version <version>.<major.minor>.<build_number>.<patch_number>
```

The 5620 SAM server does not return patch information in the XML response if there are no issued patches for the server software.

Code 5-5: Sample timestamp and server-specific version request

```
<SOAP:Envelope>
<SOAP:Header>
<header xmlns="xmlapi_1.0">
  <requestID>clientName@requestId</requestID>
</header>
</SOAP:Header>
<SOAP:Body>
  <timeStamp xmlns="xmlapi_1.0"/>
  <ping xmlns="xmlapi_1.0"/>
  <timeStamp xmlns="xmlapi_1.0"/>
  <version xmlns="xmlapi_1.0"/>
  <timeStamp xmlns="xmlapi_1.0"/>
</SOAP:Body>
</SOAP:Envelope>
```

Code 5-6: Sample response to timestamp and server-specific version request

```
<SOAP:Body>
<timeStampResponse xmlns="xmlapi_1.0">
         <millis>1168877827625</millis>
</timeStampResponse>
         <pingResponse xmlns="xmlapi_1.0" />
<timeStampResponse xmlns="xmlapi_1.0">
         <millis>1168877827626</millis>
</timeStampResponse>
<versionResponse xmlns="xmlapi_1.0">
<version>5620 SAM Version 0.0.0.0</version>
<baseVersion>0.0</baseVersion>
<build>0</build>
<patch>0</patch>
</versionResponse>
<timeStampResponse xmlns="xmlapi_1.0">
         <millis>1168877827627</millis>
</timeStampResponse>
   </SOAP:Body>
```

5620 SAM-O information model

6 — 5620 SAM-O information model overview

7 — 5620 SAM-O reference

8 — 5620 SAM-O XML packages

9 — 5620 SAM-O XML schemas

10 — XML message structure

6 — 5620 SAM-O information model overview

6.1 5620 SAM-O information model overview *6-2*

6.2 5620 SAM-O documentation *6-3*

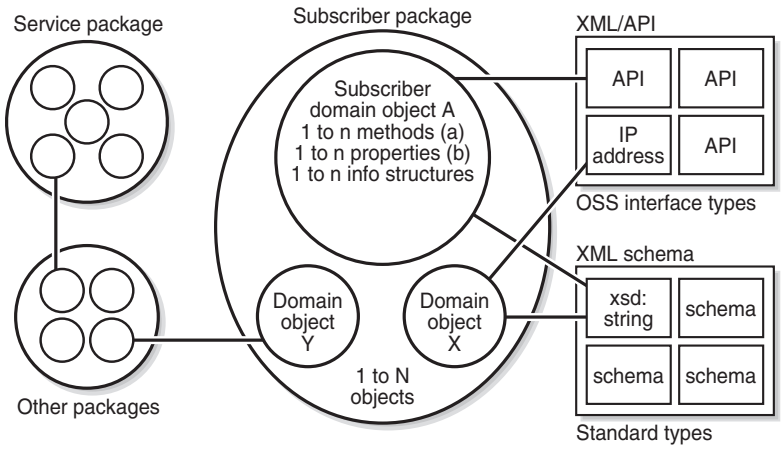
6.1 5620 SAM-O information model overview

The information model consists of:

- packages with one or more groups of related domain objects (classes), data types, bitmasks, and enumerations
- domain objects (classes) that contain the properties of the class, and NE-specific properties
- an xmlApiTypes.xsd file that contains:
  - standard XML schemas; for example, strings and integers
  - extended XML schemas; for example, IpAddresses
  - other XML schemas; for example, bitmasks
- an xmlApiMethods.xsd file that documents the available support methods; for example, pings to test connectivity
- information structures that contain both the domain objects (classes) and the properties of the information, which are called elements in the 5620 SAM-O but are also known as parameters in the CLI and 5620 SAM GUI
- inheritance and containment of domain objects

The information model provides a type of object model of the 5620 SAM-managed network. Figure 6-1 shows an example of the information model in terms of subscriber configuration.

Figure 6-1 Information model - subscriber package



a) specify method
b) specify properties <inparam... >
Notes: Packages organize related objects under the same name.
Lines represent the relationship between objects.
Circles represent an object or package, the squares represent the types.

17303

Each package provides functionality that the OSS applications can access. More package information is available from the *5620 SAM-O XML Reference* and XML schemas. See chapters 7 and 9 for more information.

- See the *PackageName*Types.xsd file for more information about the domain objects (classes), information structures, bitmasks, enumerations, and data types.
- See the *PackageName*Methods.xsd file for more information about the methods, where *PackageName* is the name of the 5620 SAM-O package type.
- See the xmlApiTypes.xsd file for more information about common XML schema data types and information structures.

Note — The DVD-ROM for the 5620 SAM application contains information about the 5620 SAM-O schemas, objects, methods, parameters, inheritance, and release deltas.

You can access the information using the index.html file in the user documentation directory.

See chapter 7 for more information about using the *5620 SAM-O XML Reference*.

6.2 5620 SAM-O documentation

The 5620 SAM-O documentation directory contains the following links to the 5620 SAM-O information model:

Note — Use the index.html file to access the *5620 SAM-O XML Reference*, XML schema, and OSS documentation. See procedure 7-1 in chapter 7 for more information about how to access the index.html file.

- **5620 SAM-O XML Reference**
  Includes the following:
  - package descriptions
  - classes
  - UML inheritance diagrams
  - properties and property values
  - NE-specific properties
  - deprecated methods and properties
  - schema changes
  - supported devices
  - 5620 SAM-O JMS interface changes
- **5620 SAM-O schemas**
  Describes the structure and allowed elements of a document, similar to a DTD
- **5620 SAM XML OSS Interface Developer Guide**
  PDF file of the 5620 SAM *XML OSS Interface Developer Guide*
- **MIB**
  MIB information for supported devices

7 — 5620 SAM-O reference

7.1 5620 SAM-O XML reference overview *7-2*

7.1 5620 SAM-O XML reference overview

You can use the *5620 SAM-O XML Reference* to obtain information on the XML interface, including package descriptions, object property values and descriptions, and UML inheritance drawings. In addition, the *5620 SAM-O XML Reference* also provides a list of NE-specific information, such as supported properties on specific NE releases. Perform Procedure 7-1 to access the *5620 SAM-O XML Reference*.

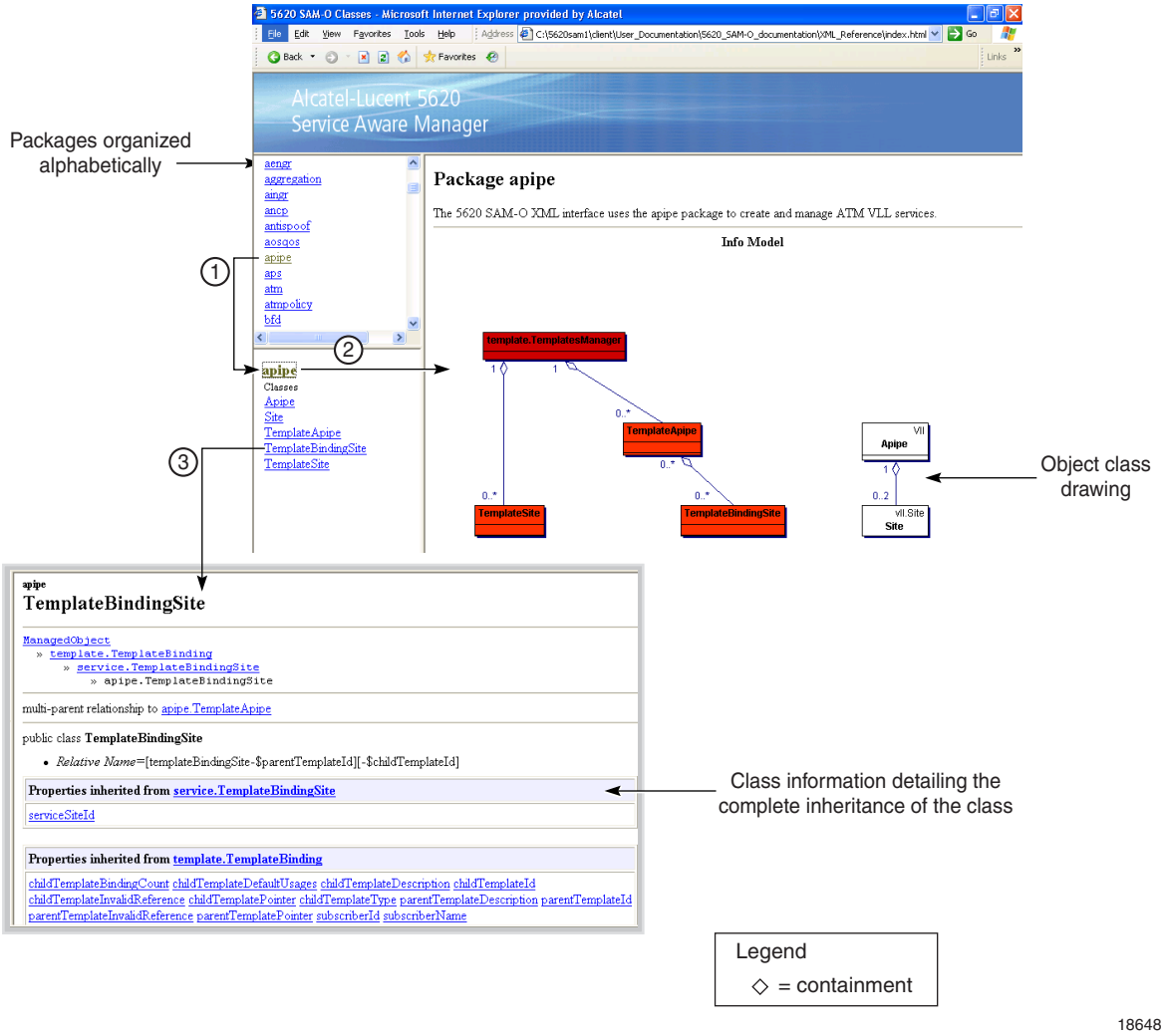
The *5620 SAM-O XML Reference* also provides information on properties and methods that are no longer supported, and changes to the schema from release to release. Schema changes can include the update, removal, or addition of the following:

- packages
- objects (classes or types)
- methods
- properties

See Procedure 7-3 for more information about how to view schema changes between releases.

Figure 7-1 shows the *5620 SAM-O XML Reference* with an XML sample object model.

Figure 7-1 Service object model



Note — If you are planning on using an existing 5620 SAM-O client that has been tested against a previous release of the 5620 SAM, you must review your 5620 SAM-O usage with the list of schema changes before you begin development to identify changes that must be made to packages, objects (classes or types), methods, or properties used by your 5620 SAM-O client.

Procedure 7-1 To access the *5620 SAM-O XML Reference*

1 Open the index.html file in the ...\client\User_Documentation\ directory on the 5620 SAM server.

Note — You can also access the OSS XML Reference through the 5620 SAM GUI. Choose Help→5620 SAM User Documentation from the 5620 SAM main menu to open the index.html file.

2 Click on the OSS XML Reference link.

3 Click on a package from the list or expand the navigation tree. The packages are described in section 8.1.

4 Click on the drawing icon to show the relationship and inheritance drawing.

5 Navigate through the drawing, the navigation tree, or the listed class information to learn more about the relationship and inheritance between classes.

Procedure 7-2 To view information about schema changes

1 Perform Procedure 7-1 to access the OSS XML Reference link.

2 Scroll down to the 5620 SAM-O Schema Changes section in the main window.

3 Click on the Schema Changes for Release *release_number* to view information about schema differences between releases.

Warning — Changes to, or removal of packages, objects (classes or types), methods, or properties, may require developers to change old 5620 SAM-O messages that have been used in previous releases.

Procedure 7-3 To view information about general methods and types

1 Perform Procedure 7-1 to access the OSS XML Reference link.

2 Click on the general methods/types link the top-left pane to view a list and description of methods and types used in the 5620 SAM-O.

Procedure 7-4 To view deprecated methods and properties

1 Perform Procedure 7-1 to access the OSS XML Reference link.

2 Click on the deprecated properties/methods link the top-left pane to view a list of the properties and methods that have been deprecated in a given release.

Warning — Changes to, or removal of packages, objects (classes or types), methods, or properties, may require developers to change old 5620 SAM-O messages that have been used in previous releases.

Procedure 7-5 To view supported devices

1 Perform Procedure 7-1 to access the OSS XML Reference link.

2 Click on the supported products link the top-left pane to view the devices that are supported in a given release.

Procedure 7-6 To view JMS changes

1 Perform Procedure 7-1 to access the OSS XML Reference link.

2 Scroll down to the 5620 SAM-O JMS Changes section in the main window.

3 Click on the JMS changes for release *release_number* to view information about changes that affect the JMS interface between releases.

8 — 5620 SAM-O XML packages

8.1 5620 SAM-O XML packages overview *8-2*

8.1 5620 SAM-O XML packages overview

The information model is defined by the XML schema, which is organized within packages. Each package represents a functional area of the 5620 SAM and contains methods and types files. See Procedure 8-1 for more information about how to view package information.

The following procedure describes how to view package information using the OSS XML Reference.

Procedure 8-1 To view package information

1. Perform Procedure 7-1 to access the OSS XML Reference link.
2. Locate the alphabetical list of packages on the top-left pane, as shown in Figure 7-1.
3. Click on a package link. The classes and types associated with the package appear in the bottom-left pane, as shown in Figure 7-1.
4. Perform one of the following:
   - **a** Click on the package name in the bottom-left pane to view the package description and information model in the center pane.
   - **b** Click on a class name in the bottom-left pane to view properties, methods, and NE-specific properties for the class.
   - **c** Click on a class name in the bottom-left pane to view properties, methods, and NE-specific properties for the class.
   - **d** Click on a type name in the bottom-left pane to view property values.

9 — 5620 SAM-O XML schemas

9.1 5620 SAM-O XML schemas *9-2*

9.1 5620 SAM-O XML schemas

XML schemas describe the structure and allowed elements of a document, similar to a DTD. Files with a *.xsd definition contain the XML schema definitions. Use XML schemas to:

- determine the elements and parameters that can appear, and the data types for the elements
- create and enforce the rules for the number and order of child elements
- determine default and fixed values for elements and parameters

Note — The DVD-ROM for the 5620 SAM application contains information about the 5620 SAM-O schemas, objects, methods, parameters, and release deltas. You can access the information using the index.html file in the user documentation directory.

Standard and specific XML schema definitions are available from the xmlApiTypes.xsd and xmlApi.xsd files. Standard and specific XML methods are available from the xmlApiMethods.xsd file.

Other XML schema definitions that are specific to a 5620 SAM function are contained in the following files:

- <*xx*>Method.xsd
- <*xx*>Types.xsd

  where <*xx*> is the package type

The <*xx*>Method.xsd files contain the methods that are associated with the packages. The <*xx*>Types.xsd files contain the objects (classes), information structures, bitmasks, and enumerations that are associated with the packages. For example, one of the packages listed in section 8.1 is the fm package. The fm package functionality is described in the fmTypes.xsd file and fmMethods.xsd files.

See the W3C websites listed in the Preface for more general information about XML schemas.

Table 9-1 describes codes 9-1 through 9-3, which show sample XML schema definitions.

Table 9-1 XML schema definition samples

Sample	Description	File
Code 9-1	Sample XML definition for a class	fmTypes.xsd
Code 9-2	Sample XML definition for an element	fmTypes.xsd
Code 9-3	Sample XML definition for a method	fmMethods.xsd

Code 9-1: Sample XML schema definition for a class

```
<!--
=
      =  Name:         fm.AdditionalTextAttribute
      =  Type:         Class
      =  Abstract:     no
      =  Singleton:    no
      =
      =  Category:     fault
      =  Deployable:   no
      =  Read Access:  all
      =  Write Access: admin, operations, fault
      =
      =  Inheritance Hierarchy:
      =
      =   <root>
      =      |
      =   fm.AdditionalTextAttribute
      =
      -->
    <xsd:complexType name="fm.AdditionalTextAttribute">
        <xsd:all>
                <!-- standard properties -->
                <xsd:element name="objectFullName" type="xsd:string" minOccurs="0"/>
                <xsd:element name="actionMask" type="generic.ModifierActionMask"
minOccurs="0"/>
                <xsd:element name="children-Set" type="Children-Set" minOccurs="0"/>
                <xsd:element name="selfAlarmed" type="xsd:boolean" minOccurs="0"/>
<!--
                  -   Name:       attributeName
                  -   Defined in: fm.AdditionalTextAttribute
                  -   Default:
                  -   Access:     Read-Create
                  -->
                <xsd:element name="attributeName" type="xsd:string" minOccurs="0"/>
<!--
                  -   Name:       order
                  -   Defined in: fm.AdditionalTextAttribute
                  -   Min/Max:    0 to 65535
                  -   Default:    0
                  -   Access:     Read-Write
                  -->
                <xsd:element name="order" type="xsd:int" minOccurs="0"/>
                <!-- Direct children -->
                <!-- Read only common properties -->
                <xsd:element name="name" type="xsd:string" minOccurs="0"/>
                <xsd:element name="deploymentState" type="DeploymentState" minOccurs="0"/>
        </xsd:all>
    </xsd:complexType>
    <xsd:element name="fm.AdditionalTextAttribute" type="fm.AdditionalTextAttribute"/>
    <xsd:element name="fm.AdditionalTextAttribute-Set" type="Children-Set"/>
    <!--
      =
      =  Name:         fm.AdditionalTextPolicy
      =  Type:         Class
      =  Abstract:     no
      =  Singleton:    no
      =
      =  Category:     fault
      =  Deployable:   no
      =  Read Access:  all
      =  Write Access: admin, operations, fault
      =
      =  Inheritance Hierarchy:
      =
      =   <root>
      =      |
      =   fm.AdditionalTextPolicy
      =
```

```
      -->
    <xsd:complexType name="fm.AdditionalTextPolicy">
        <xsd:all>
                <!-- standard properties -->
                <xsd:element name="objectFullName" type="xsd:string" minOccurs="0"/>
                <xsd:element name="actionMask" type="generic.ModifierActionMask"
minOccurs="0"/>
                <xsd:element name="children-Set" type="Children-Set" minOccurs="0"/>
                <xsd:element name="selfAlarmed" type="xsd:boolean" minOccurs="0"/>
                <!--
                  -   Name:       domain
                  -   Defined in: fm.AdditionalTextPolicy
                  -   Default:
                  -   Access:     Read-Create
                  -->
                <xsd:element name="domain" type="xsd:string" minOccurs="0"/>
                <!--
                  -   Name:       objectType
                  -   Defined in: fm.AdditionalTextPolicy
                  -   Default:
                  -   Access:     Read-Create
                  -->
                <xsd:element name="objectType" type="xsd:string" minOccurs="0"/>
                <!--
                  -   Name:       description
                  -   Defined in: fm.AdditionalTextPolicy
                  -   Min/Max:    0 to 80
                  -   Default:
                  -   Access:     Read-Write
                  -->
                <xsd:element name="description" type="xsd:string" minOccurs="0"/>
                <!--
                  -   Name:       overwrite
                  -   Defined in: fm.AdditionalTextPolicy
                  -   Default:    false
                  -   Access:     Read-Write
                  -->
                <xsd:element name="overwrite" type="xsd:boolean" minOccurs="0"/>
```

Code 9-2: Sample XML schema definition for an element

```
<!-- =
      =  Name: fm.Severity
      =  Type: Enumeration
      =
      -->
                    <xsd:simpleType name="fm.Severity">
                        <xsd:restriction base="xsd:string">
                            <!--
                              -  noalarm (-1)
                              -
                              -  Applicable Classes:
                              -     any class
                              -->
                            <xsd:enumeration value="noalarm"/>
                            <xsd:enumeration value="-1"/>
                            <!--
                              -  cleared (1)
                              -
                              -  Applicable Classes:
                              -     any class
                              -->
                            <xsd:enumeration value="cleared"/>
                            <xsd:enumeration value="1"/>
                            <!--
                              -  indeterminate (2)
                              -
                              -  Applicable Classes:
                              -     any class
```

```
                              -->
                              <xsd:enumeration value="indeterminate"/>
                              <xsd:enumeration value="2"/>
                              <!--
                                -  info (3)
                                -
                                -  Applicable Classes:
                                -     any class
                                -->
                              <xsd:enumeration value="info"/>
                              <xsd:enumeration value="3"/>
                              <!--
                                -  condition (4)
                                -
                                -  Applicable Classes:
                                -     any class
                                -->
                              <xsd:enumeration value="condition"/>
                              <xsd:enumeration value="4"/>
                              <!--
                                -  warning (5)
                                -
                                -  Applicable Classes:
                                -     any class
                                -->
                              <xsd:enumeration value="warning"/>
                              <xsd:enumeration value="5"/>
                              <!--
                                -  minor (6)
                                -
                                -  Applicable Classes:
                                -     any class
                                -->
                              <xsd:enumeration value="minor"/>
                              <xsd:enumeration value="6"/>
                              <!--
                                -  major (7)
                                -
                                -  Applicable Classes:
                                -     any class
                                -->
                              <xsd:enumeration value="major"/>
                              <xsd:enumeration value="7"/>
                              <!--
                                -  critical (8)
                                -
                                -  Applicable Classes:
                                -     any class
                                -->
                              <xsd:enumeration value="critical"/>
                              <xsd:enumeration value="8"/>
                          </xsd:restriction>
                      </xsd:simpleType>
    <xsd:element name="fm.Severity" type="fm.Severity"/>
```

Code 9-3: Sample XML schema definition for a method

```
<!--
      = fm.FaultManager.clearFaults
      = [MODIFIER]
      = [SCOPE: CLASS]
      =
      =
      = [IN]
      =   deployer - Deployer
      =      The deployment state.
      =
      =   synchronousDeploy - xsd:boolean
      =       (optional) Specify whether to block until the changes have been
```

```
      =       fully deployed to network.  A value of "true" means to block.
      =       A value of "false" means to return immediately.
      =       Default: false (asynchronous)
      =
      =   clearOnDeployFailure - xsd:boolean
      =       (optional) Specify whether clear the deployer if a failure occurs.
      =       A value of "true" means to clear.
      =       A value of "false" means to leave the failed deployer.
      =       Default: false
      =
      =   deployRetries - xsd:int
      =       (optional) The number of times to attempt re-deployment during
      =       synchronous deployment.  This parameter is meaningless in the
      =       asynchronous case.
      =       Default: 0
      =
      =   deployRetryInterval - xsd:long
      =       (optional) The number of milliseconds to wait between deployment
      =       retries. This parameter is meaningless in the asynchronous case.
      =       Default: 0
      =
      =   baseInstanceFullName - xsd:string
      =   alarmableInstanceFilter - FilterHolder
      =   faultFilter - FilterHolder
      =   scopeType - xsd:int
      =   scopeDepth - xsd:int
      =
      = [OUT]
      =
      = [EXCEPTIONS]
      =   fm.FaultManager.clearFaultsException
      -->
    <xsd:element name="fm.FaultManager.clearFaults">
       <xsd:complexType>
           <xsd:sequence>
                <xsd:element name="deployer" type="Deployer" minOccurs="1"/>
                <xsd:element name="synchronousDeploy" type="xsd:boolean" minOccurs="0"/>
                <xsd:element name="clearOnDeployFailure" type="xsd:boolean" minOccurs="0"/>
                <xsd:element name="deployRetries" type="xsd:int" minOccurs="0"/>
                <xsd:element name="deployRetryInterval" type="xsd:long" minOccurs="0"/>
                <xsd:element name="baseInstanceFullName" type="xsd:string" minOccurs="1"/>
                <xsd:element name="alarmableInstanceFilter" type="FilterHolder"
minOccurs="1"/>
                <xsd:element name="faultFilter" type="FilterHolder" minOccurs="1"/>
                <xsd:element name="scopeType" type="xsd:int" minOccurs="1"/>
                <xsd:element name="scopeDepth" type="xsd:int" minOccurs="1"/>
           </xsd:sequence>
       </xsd:complexType>
    </xsd:element>
    <xsd:element name="fm.FaultManager.clearFaultsResponse">
       <xsd:complexType>
           <xsd:all>
           </xsd:all>
       </xsd:complexType>
    </xsd:element>
```

Common elements of methods

Table 9-2 describes elements that are common to all 5620 SAM methods.

Table 9-2 Common elements of methods

Element	Description	Options
deployer	Specifies the deployment state.	—

(1 of 2)

Element	Description	Options
synchronousDeploy	Specifies whether synchronous or asynchronous deployment is used.	• false (default) Asynchronous deployment • true Synchronous deployment
clearOnDeployFailure	Specifies whether to clear the deployer if a failure occurs.	• false (default) The deployer is not cleared • true (recommended) Clear the deployer
deployRetries	Specifies the number of times that re-deployment is attempted during a synchronous deployment. This is an optional parameter.	0 (default and recommended)
deployRetryInterval	Specifies the time, in milliseconds, between deployment attempts in synchronous deployments. This parameter is optional.	0 (default)
instanceFullName	Specifies the full name of the object.	—
configInfo	Specifies any class type.	

(2 of 2)

Common elements of classes (types)

The following sections describe the elements that are common in all object classes (types).

actionMask

Every class definition contains an <actionMask> element. The <actionMask> specifies what operation occurs to the object during configuration, specific to each configured object. The <actionMask> is only used during configuration. You can use the genericTypes.xsd file to specify the definition of the <actionMask> (generic.ModifierActionMask). Table 9-3 lists the valid action types and the numeric equivalents.

Table 9-3 Action types

Action Type	Numerical equivalent	Description
create	1	To create an object based on the elements (parameters) received from the XML API. Only read-write and read-create attributes are allowed. The read-create attribute is mandatory.
modify	2	To modify an existing object based on the elements (parameters) received from the XML API. Only read-write attributes are allowed.
delete	4	To delete an existing object.
reset	8	To reset the elements (parameters) of an existing object to the default values.

Note — Not all actionMasks are applicable to all classes.

Alcatel-Lucent recommends that you use one actionType for each request. However, you can combine the actionTypes. For example, when you want to create and modify an object that does not exist, you can specify the create and modify bits in the same request. Code 9-4 shows an example of a combined request. Alternately, you can use the numeric value of the bit instead of the text.

Code 9-4: Combined create and modify request

```
<actionMask>
<bit>create</bit>
<bit>modify</bit>
</actionMask>
```

Accessibility types

Each element (parameter) has an accessibility type. You can view the accessibility type in the *PackageName*Types.xsd file, where *PackageName* is the name of the 5620 SAM-O package type. The accessibility types are:

- read-only, which specifies that the element is read-only and cannot be used in a configuration request
- read-write, which specifies that the element is read-write at any time using the create or modify actionTypes
- read-create, which specifies that the element can only be written during a create actionType request

Object full name

The <objectFullName> element specifies the 5620 SAM unique identifier of the object instance. This name is unique across the entire 5620 SAM system. The object full name is also referred to as the object fully distinguished name. The format of the object full name depends on the type of object. Table 9-4 lists a sample of the object types and how they are uniquely identified.

Note — Do not compose or parse the <objectFullName> element. You can only use the <objectFullName> element in a complete format as a unique identifier or key. The element identifies or retrieves a specific 5620 SAM object.

You must use the objectFullName format when you define an <x>ObjectPointer element where <x> defines the object. For example:

```
<ingressPolicyObjectPointer>

Access Ingress:1

</ingressPolicyObjectPointer>
```

Table 9-4 Sample unique identifiers for network objects

Network object (class)	Format	Example
Network element		
Network element	Created for each managed node and forms the base of site-specific objects. network:*siteID*	network:10.1.1.88
Policies		
Policy manager	A policy manager exists for each type of policy at the network and element level and contains sets of policies of a type. For example, access and slope. *policy Manager Type* for network level or network:*siteID*:*policyManagerType* for element level	Access Egress or network:10.1.1.88:Access Egress
Access egress policy (aengr.Policy)	policy Manager Type:*ID* or *network*:*siteID*:*policyManagerType*:*policyID* for element level	Access Egress:2 or network:10.1.1.88:Access Egress:2
Access egress queue (aengr.Queue)	*network*:*siteID*:*policyManagerType*:*policyID* :queue-*ID*	network:10.1.1.88:Access Egress:2:queue-1
Access ingress policy (aingr.Policy)	*policyManagerType*:*ID* or *network*:*siteID*:*policyManagerType*:*policyID* for element level	Access Ingress:2 or network:10.1.1.88:Access Ingress:2
Access ingress forwarding class (aingr.ForwardingClass)	*policyManagerType*:*forwarding-class-type* or *network*:*siteID*:*policyManagerType*:*forwarding-class-type* for element level	Access Engress:2:forwarding-class-be or network:10.1.202.94:Access Engress:2:forwarding-class-be
Access ingress queue (aingr.Queue)	network:*siteID*:*policyManagerType*:*policyID* :queue-*ID*	network:10.1.1.88:Access Ingress:2:queue-1
Network policy (niegr.Policy)	Network:*ID* or *network*:*siteID*:Network:*ID* for element level	Network:2 network:10.1.1.88:Network:2
Network forwarding class (niegr.ForwardingClass)	Network*ID*:*forwarding-class-type* or *network*:*siteID*:Network*ID*:*forwarding-class -type* for element level	network:10.1.1.88:Network:3:forwarding-class-af
File policy (file.Policy)	File:*policyID* or *network*:*siteID*:File:*ID* for element level	File:2 or network:10.1.1.88:File:2
Slope policy (slope.Policy)	*slopeName* or *network*:*siteID*:Slope:default	network:10.1.1.88:Slope:default

(1 of 3)

Network object (class)	Format	Example
IP filter policy (aclFilter.IPFilter)	IP Filter:*policyID* or *network*:*siteID*:IP Fitler:*policyID* for element level	IP Filter:5 or network:10.1.1.88:IP Filter:5
Mediation security policy (security.MediationPolicy)	pollerManager:mediation-security-*policyID*	pollerManager:mediation-security-1
Deployer policy (mediation.DeploymentPolicy)	Default.DeployerBank:Policy	Default.DeployerBank:Policy
Deployer (generic.DeployerInfo)	Default.DeployerBank:depl-*ID*	Default.DeployerBank:depl-322
Router		
Router (rtr.VirtualRouter)	There is only one virtual router for each node. *network*:*siteID*:router-1 for element level	network:10.1.1.88:router-1
Network Interface (rtr.NetworkInterface)	*network*:*siteID*:router-1:ip-interface-*ID* for element level	network:10.1.1.88:router-1:ip-interface-2
Static route (rtr.StaticRoute)	*network*:*siteID*:router-1:SR-*ID-DestinationsiteID- IP mask* for element level	network:10.1.1.88:router-1:SR-2-12.119.60.0-255.255.255.0
OSPF		
OSPF (ospf.area)	ospf:area-*siteID*	ospf:area-2.2.2.2
OSPF site (ospf.Site)	*network*:*siteID*:router-1:ospf for element level	network:10.11.1.219:router-1:ospf
OSPF area (ospf.AreaSite)	*network*:*siteID*:router-1:ospf:areaSite-*siteID* for element level	network:10.11.1.219:router-1:ospf:areaSite-0.0.0.0
OSPF interface (ospf.interface)	*network*:*siteID*:router-1:ospf:areaSite-*siteID*:interface-*siteID* for element level	network:10.11.1.219:router-1:ospf:areaSite-0.0.0.0:interface-3.3.3.3.0
OSPF export policy (ospf.ExportPolicy)	*network*:*siteID*:router-1:ospf:exportPolicy for element level	network:10.11.1.219:router-1:ospf:exportPolicy
BGP		
BGP (bpg.Site)	*network*:*siteID*:router-1:bgp for element level	network:10.11.1.219:router-1:bgp
BGP peer group (bpg.PeerGroup)	*network*:*siteID*:router-1:bgp:group-*Group name* for element level	network:10.11.1.219:router-1:bgp:group-Sample-Client
BGP group import policy (bgp.GroupImportPolicy)	*network*:*siteID*:router-1:bgp:group-*Group name*:importPolicy for element level	network:10.11.1.219:router-1:bgp:group-Sample-Client:importPolicy
BGP group export policy (bgp.GroupExportPolicy)	*network*:*siteID*:router-1:bgp:group-*Group name*:exportPolicy for element level	network:10.11.1.219:router-1:bgp:group-Sample-Client:exportPolicy
BGP site export policy (bgp.SiteExportPolicy)	*network*:*siteID*:router-1:bgp:exportPolicy for element level	network:10.11.1.219:router-1:bgp:exportPolicy
BGP site import policy (bgp.SiteExportPolicy)	*network*:*siteID*:router-1:bgp:importPolicy for element level	network:10.11.1.219:router-1:bgp:importPolicy
BGP confederation (bgp.Confederation)	*network*:*siteID*:router-1:confederation-*ID* for element level	network:10.11.1.219:router-1:confederation-100
BGP confederation member (bgp.ConfederationMember)	*network*:*siteID*:router-1:confederation-*ID*:member-*ID* for element level	network:110.11.1.219:router-1:confederation-100:member-10
LDP		

(2 of 3)

Network object (class)	Format	Example
LDP (ldp.Site)	*network*:*siteID*:router-1:ldp for element level	network:10.11.1.219:router-1:ldp
MPLS		
MPLS (mpls.Site)	*network*:*siteID*:router-1:mpls for element level	network:10.1.1.219:router-1:mpls
MPLS interface (mpls.interface)	*network*:*siteID*:router-1:mpls:interface-*ID* for element level	network:10.1.1.219:router-1:mpls:interface-1
MPLS dynamic LSP (mpls.DynamicLSP)	lsp:from-*Source siteID*-id-*ID*	lsp:from-10.1.1.88-id-1
MPLS LSP path (mpsl.LspPath)	lsp:from-*Source siteID*-id-*ID*:lsppath-*ID*	lsp:from-10.1.1.88-id-1:lsppath-2
MPLS provisioned path (mpls.ProvisionedPath)	provisionedMplsTePath:from-*Source siteID*-id-*ID*	provisionedMplsTePath:from-10.1.1.88-id-1
MPLS provisioned hop (mpls.ProvisionedHop)	provisionedMplsTePath:from-*Source siteID*-id-*ID*:hop-*ID*	provisionedMplsTePath:from-10.1.1.88-id-1:hop-2
MPLS tunnel (mpls.Tunnel)	mplsTunnel:from-*Source siteID*-id-*ID*	mplsTunnel:from-10.1.1.88-id-1
Subscribers		
Subscriber (subscr.Subscriber)	subscriber:*ID*	subscriber:21
Services		
Service (service.Service)	serviceManager:service:*ID*	svc-mgr:service-12
Service site (service.Site)	serviceManager:service:*ID*:*siteID*	svc-mgr:service-5:10.1.1.91
Service access interface (service.AccessInterface)	serviceManager:service*ID*:*site ID*:ip-interface-*ID*	svc-mgr:service-5:10.1.1.91:interface-1/1/4-inner-tag-0-outer-tag-11 or svc-mgr:service-52:10.1.202.93:ip-interface-2051
Channels		
STS192 Channel (sonetequipment.Sts192Channel)	network:*siteID*:shelf-*ID*:cardSlot-*ID*:card:daughterCardSlot-*ID*:daughterCard:port-*ID*:*Sts192Channel*	network:10.1.1.218:shelf-1:cardSlot-2:card:daughterCardSlot-1:daughterCard:port-1:sts192-1
DS3/E3 Channel (tdmequipment.DS3E3Channel)	network:*siteID*:shelf-*ID*:cardSlot-*ID*:card:daughterCardSlot-*ID*:daughterCard:port-*ID*:*DS3E3Channel-ID*	network:10.1.1.218:shelf-1:cardSlot-3:card:daughterCardSlot-1:daughterCard:port-1:ds3e3-1
DS0 Channel Group (tdmequipment.DS0ChannelGroup)	network:*siteID*:shelf-*ID*:cardSlot-*ID*:card:daughterCardSlot-*ID*:daughterCard:port-*ID*:*DS1E1Channel-ID*:*DS0ChannelGroup-ID*	network:10.1.1.218:shelf-1:cardSlot-3:card:daughterCardSlot-1:daughterCard:port-1:ds1e1-3:ds0Grp-3

(3 of 3)

Note 1 — There is no documented list of object full name formats. The samples in Table 9-4 are not guaranteed to remain valid from release to release of the 5620 SAM. It is recommended that 5620 SAM-O request XML script writers find FDNs to use in their requests by performing a <find> request for the class of the object they are looking for. See section 14.1 for more information about the <find> method.

For classes with no inventory present, the find will not return a result where you can view an object full name. Create the object using the 5620 SAM, assuming the physical network element is configured to allow the creation of the object. You can then perform the find request that allows you to view the object full name for the created object.

You can also monitor the 5620 SAM logs for the FDN when you create or modify an object using the 5620 SAM GUI. See section 10.1 for more information about log files.

Note 2 — Do not compose or parse the <objectFullName> element. You can only use the <objectFullName> element in a complete format as a unique identifier or key. The element identifies or retrieves a specific 5620 SAM object.

10 — XML message structure

10.1 XML message structure overview *10-2*

10.2 CLI commands within XML methods *10-17*

10.3 Mapping XML methods to GUI operations *10-18*

10.4 XML samples *10-20*

10.1 XML message structure overview

The 5620 SAM-O uses request, response, and fault messages to handle the XML message data related to network objects. Messages are sent to the 5620 SAM-O using an RPC pattern. The messages are constructed and wrapped in a SOAP envelope. The 5620 SAM-O then returns a success response message or a failure fault message.

The SOAP encoding transports the XML data. Three types of SOAP messages are used:

- requests
- responses for a successful or unsuccessful execution
- faults for SOAP messages that are not well-formed

Message structure

Figure 10-1 shows the structure of the three SOAP message types.

Figure 10-1 SOAP message types

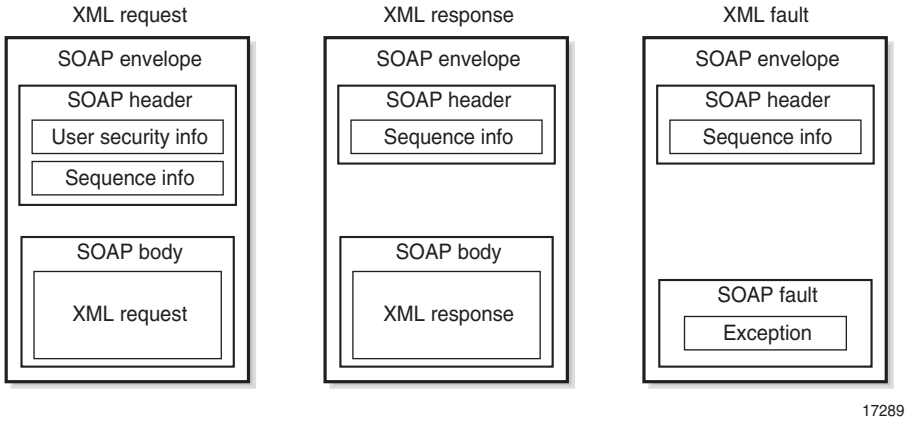


Table 10-1 describes the SOAP message types.

Table 10-1 SOAP message type details

SOAP message type	Contents	Details
Request	SOAP envelope	Contains all message data

	SOAP header	The header contains:
		• user security details:
		  • user IDs in plain text format
		  • passwords in plain text or an MD5-hashed format. A utility to convert plain text passwords to an MD5-hashed format is available. See section 3.4 for more information.
		• sequence information in the form of a requestID to associate requests with responses and faults. You can use a string for the requestID. The requestID must be unique per HTTP request.
		Do not use the following formats:
		  • packageName-packageName.ObjectName-[instanceFullNameIfApplicable].methodName.
		  For example:
		  netw-netw.Topology-[].configure
		  • Areq*ObjectNameMethodName*-CLIENT-*userId-sequenceNumber*
		  For example:
		  AreqVirtualInterfaceConfigureIpAddresses-CLIENT-admin-2
		  • Sreq*ObjectNameMethodName*-CLIENT-*userId-sequenceNumber*
		  For example:
		  SreqTopologyConfigure-CLIENT-operator-12

	SOAP body	Contains the XML request in the general format: <package.object.method> or <internal_method> <inParam1>value</inParam1> <inParamX>value</inParamX> </package.object.method> or </internal_method>
Response	SOAP envelope	Contains all message data
	SOAP header	Contains sequence information in the form of the request ID of the original request
	SOAP body	Contains the XML response in the general format: <package.object.methodResponse> or <package.object.methodException> or <XMLException> or <IMException> <outParam1>value</outParam1> <outParamX>value</outParamX>
Fault	SOAP envelope	Contains all message data
	SOAP header	Contains sequence information in the form of the request ID of the original request
	SOAP fault	Contains: • SOAP fault information • exception details See "Faults and exceptions" for more information about the fault details.

Note — The 5620 SAM-O supports streaming SOAP messages.

Requests

The 5620 SAM-O supports the following request types:

- synchronous
- asynchronous (default)

There are three scenarios for a message request:

- The OSS requests a database interaction that causes network deployments, for example, provisioning a service. See Figures 10-2 and 10-3.
- The OSS requests a database interaction that does not cause network deployments. See Figure 10-4.
- The OSS requests read information, for example, inventory information or an alarm feed. See Figure 10-5.

Note — The 5620 SAM maps the object FDNs to a corresponding database index. Alcatel-Lucent recommends using FDN properties in queries to maximize the processing speed of the query. The 5620 SAM-O queries should also use concrete classes rather than an abstract class to optimize performance.

Figure 10-2 Database interaction with synchronous network deployment

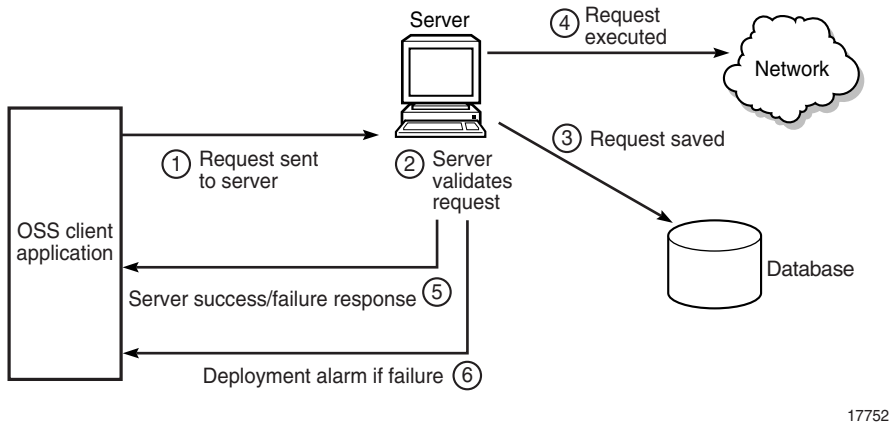


Figure 10-3 Database interaction with asynchronous network deployment

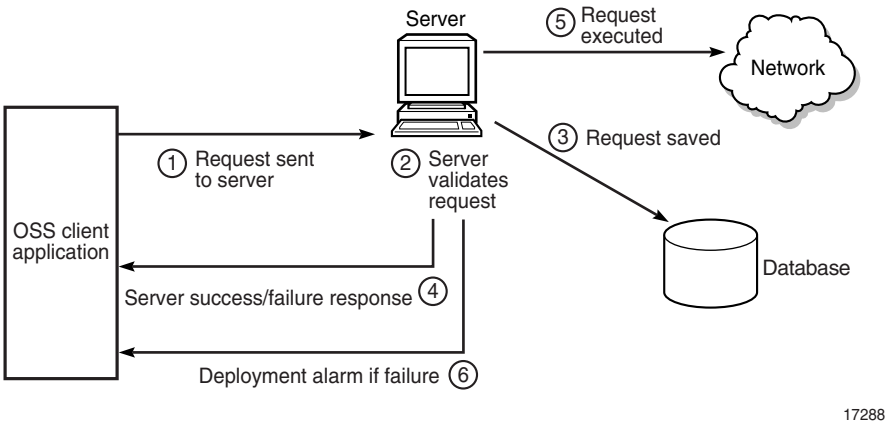


Figure 10-4 Database interaction without network deployment

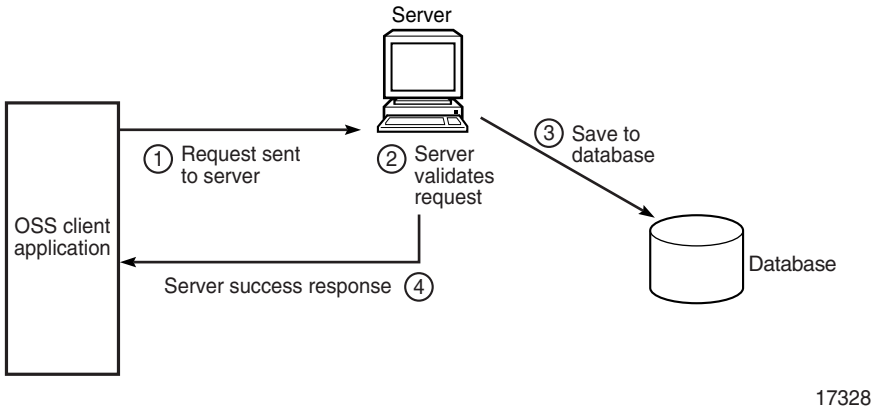
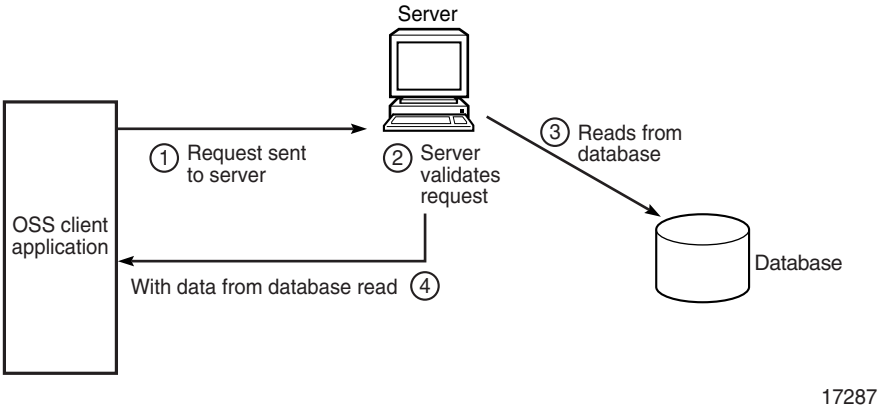


Figure 10-5 Read from database interaction



Network interactions are performed using deployers. When a request fails before being committed to the database, a fault message is sent to the OSSI client. For a synchronous request, the XML response indicates the success or failure of deployments. A single request may result in multiple deployers. The success or failure of multiple deployers is returned. OSS clients must be designed to accept more than one deployer in a response.

See "Faults and exceptions" in this section for more information about deployment failure recovery. See section 16.3 for more information about deployer failure JMS alarms and deployer request configurations.

Sample request

The XML/SOAP sample requests in this guide are intended as a base on which to build your own requests. The sample requests may contain:

- Methods that are deprecated by the 5620 SAM-O OSS
- Configuration information that is not applicable to your network architecture

Ensure that you test your request before network deployment.

Code 10-1 shows a sample request. The request shows the sequence of events and interactions associated with a request. In this sample, the OSS application changes the configuration of port 1/1/3 on node 10.1.202.93 to be access, dot1q, and administratively up.

Caution — The following sample message is an example of the request format. Use the sample as a base to build your request. Ensure that you test your request before network deployment.

Code 10-1: Sample request

```
<SOAP:Body>
        <generic.GenericObject.configureInstanceWithResult
xmlns="xmlapi_1.0">
            <deployer>immediate</deployer>
<distinguishedName>network:10.1.202.93:shelf-1:cardSlot-2:card:daughterCardS
lot-1:daughterCard:port-3</distinguishedName>
            <includeChildren>true</includeChildren>
            <configInfo>
                <equipment.PhysicalPort>
                   <actionMask><bit>modify</bit></actionMask>
                   <administrativeState>portInService</administrativeState>
                   <mode>access</mode>
                   <encapType>qEncap</encapType>
                   <children-Set/>
                </equipment.PhysicalPort>
            </configInfo>
        </generic.GenericObject.configureInstanceWithResult>
    </SOAP:Body>
```

Code 10-1 lists the following details for the request:

- The generic.GenericObject.configureInstanceWithResult identifies:
  - the use of the generic package
  - the executed method configureInstanceWithResult is defined in the GenericObject class
- The standards-based schema and version.

- The distinguishedName tag defines the unique Distinguished Full Name of the object in 5620 SAM.
- The configInfo object is a generic object that can contain various objects such as equipment.PhysicalPort

The configureInstanceWithResult method, as indicated in the request, is defined in the genericMethods.xsd. The equipmentTypes.xsd schema file defines the valid parameters available for configuration on equipment.PhysicalPort. The request is executed by the server according to the parameters specified in the configureInstanceWithResult method.

XML request grouping

You can concatenate multiple message requests in the body of a single SOAP message. The response contains the result of each request.

Code 10-2 shows an example of a SOAP message with multiple message requests.

Code 10-2: XML request grouping

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope
  xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP:Header>
        <header xmlns="xmlapi_1.0">
            <security>
                <user>user name</user>
                <password>MD5-hashed password</password>
            </security>
            <requestID>clientName@requestId</requestID>        </header>
    </SOAP:Header>
    <SOAP:Body>
        <find xmlns="xmlapi_1.0">
            <fullClassName>equipment.DaughterCard</fullClassName>
            <filter>
                <equal name="siteName" value="sim202_93"/>
            </filter>
            <resultFilter>
                <attribute>objectFullName</attribute>
                <attribute>operationalState</attribute>
                <attribute>administrativeState</attribute>
                <attribute>specificType</attribute>
    <children></children>
            </resultFilter>
        </find>
        <find xmlns="xmlapi_1.0">
            <fullClassName>equipment.PhysicalPort</fullClassName>
            <filter>
                <equal name="siteName" value="sim202_93"/>
            </filter>
            <resultFilter>
                <attribute>objectFullName</attribute>
                <attribute>operationalState</attribute>
                <attribute>administrativeState</attribute>
                <attribute>mode</attribute>
    <children></children>
            </resultFilter>
        </find>
    </SOAP:Body>
```

Password format

The 5620 SAM server accepts XML requests with plaintext or MD5-hashed passwords. See section 3.4 for more information.

Action on failure

The <continueOnFailure> and <onFailure> options help manage failed requests.

```
<continueOnFailure>false</continueOnFailure>
```

When the <continueOnFailure> option is set to false and a request in a concatenated message fails, the 5620 SAM responds as follows:

- ignores all remaining requests in the concatenated message
- returns responses for all successful requests up to the failed request
- generates an exception message for the failure

If the <continueOnFailure> option is set to true, the execution of the requests continues even if a failure has occurred. Responses and exceptions are returned for each success or failure, respectively.

The <execute> and <onFailure> options allow requests to recover from a failure. Multiple message requests can be placed within <execute> and <onFailure> blocks. The execution begins with the first request in the <execute> block. If there is a failure, the execution branches to the start of the <onFailure> block. Returned results contain the results of all the requests up until the failed request in the <execute> block, and the results of all the requests up until the failed request, if any, in the <onFailure> block. Only one level of nesting is allowed inside an <onFailure> block.

Each <execute> block can have a timeout parameter. For example,

```
<execute timeout="500000" xmlns="xmlapi_1.0">

…

</execute>

<onFailure>

…

</onFailure>
```

Elapsed time is verified between requests in the <execute> block. If the timeout is exceeded, the execution is terminated and a request timed out failure is returned in the result.

Note — It is more difficult to troubleshoot problems with multiple message requests in comparison to single message requests. When failures occur, a corrective configuration action may need to be taken to address certain issues, such as a partial configuration.

Alcatel-Lucent recommends that you consider the risks associated with support and maintenance if you choose to use multiple message requests.

Code 10-3 shows a request to create a VPLS site.

Code 10-3: Sample request to create a VPLS site

```
<soapenv:Body>
      <generic.GenericObject.configureChildInstance
xmlns="xmlapi_1.0">
   <distinguishedName>svc-mgr:service-67</distinguishedName>
   <childConfigInfo>
    <vpls.Site>
   <actionMask>
  <bit>create</bit>
 <bit>modify</bit>
 </actionMask>
    <siteId>10.1.241.69</siteId>
    <displayedName>Alcatel_Lab:VPLS:100020:sim241_2</displayedName>
  <administrativeState>1</administrativeState>
  <mtu>0</mtu>
    </vpls.Site>
  </childConfigInfo>
    </generic.GenericObject.configureChildInstance>
  </soapenv:Body>
```

Code 10-4 shows a request that can be used after a failure occurs when creating the VPLS site.

Code 10-4: Sample request after a failure during VPLS site creation

```
<soapenv:Body>
    <generic.GenericObject.deleteInstance xmlns="xmlapi_1.0">
  <distinguishedName>
svc-mgr:service-67:10.1.241.69
                      </distinguishedName>
     </generic.GenericObject.deleteInstance>
</soapenv:Body>
```

You can combine both requests in codes 10-3 and 10-4 with the execute and onFailure blocks, as shown in code 10-5.

Code 10-5: Sample request with execute and onFailure blocks

```
<soapenv:Body>
  <execute>
     <generic.GenericObject.configureChildInstance
xmlns="xmlapi_1.0">
   <distinguishedName>svc-mgr:service-67</distinguishedName>
   <childConfigInfo>
    <vpls.Site>
   <actionMask>
  <bit>create</bit>
  <bit>modify</bit>
 </actionMask>
    <siteId>10.1.241.69</siteId>
    <displayedName>Alcatel_Lab:VPLS:100020:sim241_2</displayedName>
  <administrativeState>1</administrativeState>
  <mtu>0</mtu>
    </vpls.Site>
  </childConfigInfo>
 </generic.GenericObject.configureChildInstance>
  </execute>
```

```
  <onFailure>
<generic.GenericObject.deleteInstance xmlns="xmlapi_1.0">
  <distinguishedName>
svc-mgr:service-67:10.1.241.69
                    </distinguishedName>
  </generic.GenericObject.deleteInstance>
</onFailure>
</soapenv:Body>
```

Responses

Responses identify the execution of a method.

Note — Alcatel-Lucent recommends that you construct requests for one object at a time. For example, to configure individual cards or interfaces.

Before you create services, configure the related objects. Each step to create the service should be sent as an individual request rather than as a complex query. Smaller requests can simplify the debugging of 5620 SAM-O application interactions.

Code 10-6 shows a successful response to the request to configure a network interface.

Code 10-6: Sample response to a successful request to configure a network interface

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP:Header>
      <header xmlns="xmlapi_1.0">
         <requestID>clientName@requestId</requestID>      </header>
   </SOAP:Header>
   <SOAP:Body>
      <rtr.RoutingInstanceSite.configureResponse xmlns="xmlapi_1.0">

<objectFullName>network:10.1.202.95:router-1:ip-interface-24</objectFullName
>
      </rtr.RoutingInstanceSite.configureResponse>
   </SOAP:Body>
</SOAP:Envelope>
```

You can also include the content type information in the HTTP response header, using the <xmlapi> element in the nms-server.xml file. The xmlContentType attribute defines the content; for example, using the default attribute value of text/xml; charset=ISO-8859-1. You can use any valid string to configure the attribute.

Code 10-7 shows a response that includes the content type.

Code 10-7: Sample response that includes content type

```
HTTP/1.1 200 OK
  Request Version: HTTP/1.1
  Response Code: 200
Server: Apache-Coyote/1.1
X-Powered-By: Servlet 2.4; JBoss-4.0.2 (build: CVSTag=JBoss_4_0_2
date=200505022023)/Tomcat-5.5
Content-Type: text/xml;charset=ISO-8859-1
Transfer-Encoding: chunked
Date: Fri, 12 May 2006 16:55:02 GMT
```

HTTP response codes

Table 10-2 defines the HTTP response codes based on the RFC 2616 standard. The 5620 SAM-O does not use all responses defined by the RFC 2616 standard. The HTTP response codes are associated with the successful or unsuccessful transmission of the XML response.

Table 10-2 HTTP response codes

Code Range	Definition
1xx [(1)]	Informational - Request received, continuing process.
2xx	Success - The action was successfully received, understood, and accepted.
3xx [(1)]	Redirection - Further action must be taken in order to complete the request.
4xx	Client Error - The request contains bad syntax or cannot be fulfilled.
5xx	Server Error - The server failed to fulfill an apparently valid request.

Note
(1) The XML API does not send 1xx and 3xx messages.

The XML API client must check the HTTP response code before it attempts to parse the XML API results. An HTTP response code between 200 and 299 identifies successful execution of the request. The HTTP body for a 2xx response contains a valid XML SOAP response. All other response codes may not contain well-formed XML.

Faults and exceptions

The 5620 SAM-O produces a SOAP fault message when there is a problem processing a SOAP header or empty SOAP body. The subsequent errors result in an exception within the SOAP body.

SOAP exception messages can contain the following fault-related details:

- <faultcode> provides some standard SOAP information, including:
  - version mismatch between SOAP versions
  - server problems
  - client problems, for example, sending a request to execute a method that does not exist on the server
- <faultstring> error message
- <faultactor> is the URL of the server if the server is the problem

The 5620 SAM-O uses the following format for the <faultstring> element:

```
<faultstring>[error_string] message</faultstring>
```

Table 10-3 lists the error types that are associated with the <faultstring> element.

Table 10-3 <faultstring> error types for SOAP fault messages

[error_string]	Details
license	Not licensed to use the XML API
soap	Incomplete or incorrect SOAP construction
xml-header	XML API header is missing, or is missing necessary information, such as the user ID or the MD5-hashed password
security	Authentication error, for example, invalid user ID

Table 10-4 describes the faults and exceptions that can appear in an XML response.

Table 10-4 Summary of XML response faults and exceptions

Fault or exception	Format	Description

SOAPFault	`<SOAP:Fault>`
	`<faultcode>value</faultcode>`
	`<faultstring>value</faultstring>`
	`<faultactor>value</faultactor><detail>`
	`<requestID>value</requestID></detail>`
	`</SOAP:Fault>`
		Errors in a SOAP header.
		The <faultstring> in a <SOAP:Fault> response provides details on the fault.
		The strings may change over different releases of the 5620 SAM and therefore should not be parsed by the OSS application.

XMLException	`<XMLException xmlns="xmlapi_1.0">`
	`<description>value</description>`
	`<line>value</line>`
	`<column>value</column>`
	`</XMLException>`
		Errors in the XML syntax or because of a non-existent method or attribute in the 5620 SAM-O.
		The XMLException <description> field provides details on the XML errors.
		The strings may change over different releases of the 5620 SAM and therefore should not be parsed by the OSS application.

IMException	`<IMException>`
	`<cause>value</cause>`
	`<description>value</description>`
	`</IMException`
		Errors in the object model. For example, binding a SAP to a non-existent tunnel.
		The <description> field provides details about the error.
		The strings may change over different releases of the 5620 SAM and therefore should not be parsed by the OSS application.

BaseException	`<BaseException xmlns="xmlapi_1.0">`
	`<description>value</description>`
	`</BaseException>`
		Operation-related exceptions, such as configurationException and creationException. The BaseException appears in the XML response in these exceptions.
		The <description> field in the XML response provides details about the error.
		The strings may change over different releases of the 5620 SAM and therefore should not be parsed by the OSS application.

Table 10-5 lists sample faults and exceptions that can appear in an XML response, as described in table 10-4.

Table 10-5 Sample faults and exceptions

Fault or exception	Example fault or exception

SOAPFault	`<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">`
	`  <SOAP:Header>`
	`  <header xmlns="xmlapi_1.0">`
	`  <requestID>XmlApiClient:0</requestID>`
	`  </header>`
	`  </SOAP:Header>`
	`  <SOAP:Fault>`
	`  <faultcode>SOAP:Client</faultcode>`
	`  <faultstring>[security] Users require OSS Management privileges to use SAM-O</faultstring>`
	`  <faultactor>XmlApi</faultactor>`
	`  <detail>`
	`  <requestID>XmlApiClient:0</requestID>`
	`  </detail>`
	`  </SOAP:Fault>`
	`</SOAP:Envelope>`

XMLException	`<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">`
	`  <SOAP:Header>`
	`  <header xmlns="xmlapi_1.0">`
	`  <requestID>XmlApiClient:0</requestID>`
	`  </header>`
	`  </SOAP:Header>`
	`  <SOAP:Body>`
	`  <XMLException xmlns="xmlapi_1.0">`
	`  <description>Command 'fm.FaultManager.findFault2s' is not defined</description>`
	`  <line>4</line>`
	`  <column>57</column>`
	`  </XMLException>`
	`  </SOAP:Body>`
	`</SOAP:Envelope>`

IMException	`<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">`
	`  <SOAP:Header>`
	`  <header xmlns="xmlapi_1.0">`
	`  <requestID>client1:0</requestID>`
	`  </header>`
	`  </SOAP:Header>`
	`  <SOAP:Body>`
	`  <IMException xmlns="xmlapi_1.0">`
	`  <cause>java.lang.SecurityException</cause>`
	`  <description>SecurityManager:Security Breach: No such user: 'oss_client'</description>`
	`  </IMException>`
	`  </SOAP:Body>`

(1 of 2)

Fault or exception	Example fault or exception

BaseException	`<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">`
	`  <SOAP:Header>`
	`  <header xmlns="xmlapi_1.0">`
	`  <requestID>XmlApiClient:5</requestID>`
	`  </header>`
	`  </SOAP:Header>`
	`  <SOAP:Body>`
	`  <script.ScriptManager.configureException xmlns="xmlapi_1.0">`
	`  <description>[app: script] [class: script.Script] [instance: -unknown-]`
	`[descr: the NE type is invalide type,{neType=craig}] </description>`
	`  </script.ScriptManager.configureException>`
	`  </SOAP:Body>`
	`</SOAP:Envelope>`

(2 of 2)

Code 10-8 shows an exception message generated when a user does not have the required OSS management privileges to use SAM-O. The sample exception provides the following details:

- <faultcode> element indicates that the server could not execute the request because of a fault of the client
- <faultstring> element provides the security error string, which indicates that the Client user specified does not have the required OSS management privileges to use SAM-O
- <faultactor> element identifies the XmlApi as the fault factor

Code 10-8: Sample exception message for failed router configuration

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP:Header>
      <header xmlns="xmlapi_1.0">
         <requestID>clientName@requestId</requestID>
         <requestTime>Jan 2, 2007 1:18:53 PM</requestTime>
         <responseTime>Jan 2, 2007 1:18:53 PM</responseTime>
      </header>
   </SOAP:Header>
   <SOAP:Fault>
      <faultcode>SOAP:Client</faultcode>
      <faultstring>[security] Users require OSS Management privileges to use
SAM-O</faultstring>
      <faultactor>XmlApi</faultactor>
      <detail>
         <requestID>clientName@requestId</requestID>
      </detail>
   </SOAP:Fault>
</SOAP:Envelope>
```

Note — The <responseTime> tag in a 5620 SAM-O header is the time at which the response stream is opened.

The <detail> element in the SOAP exception message identifies the command that caused the request failure.

The 5620 SAM-O validates the XML request when it is sent to the server. If the validation fails, 5620 SAM-O generates and sends an exception message as shown in Code 10-8. If the request requires changes to the managed network, deployers are created and queued to send the changes to the routers. If the deployment fails, 5620 SAM-O raises an alarm which includes details for the particular deployer ID. The network can be in an indeterminate state because another configuration request may have succeeded. See section 16.3 for more information about creating and viewing deployer requests and alarms.

To recover from request errors, the OSS application may need to check the affected objects in the deployer and issue appropriate requests to undo the action that caused the error. Another option is to manually maintain a list of failures for resolution at a later date.

Note — See "Workflow to handle deployer failures" in chapter 16 for more information about how to handle deployment errors that may cause the 5620 SAM database to be out of synchronization with the managed network.

Logging exception messages

Additional troubleshooting information about exceptions is available in the server EmsServer.log file on the 5620 SAM server. To obtain the log file, go to the 5620 SAM server installation directory and navigate to the /nms/log/server directory. You can then open the log file using a text editing tool. You can view the backup files in the same directory as the original file. See section 10.3 for more information.

Log file for XML requests, responses, and exceptions

You can enable the logging of 5620 SAM-O XML request, response, and exception messages on the active or standby 5620 SAM servers. The entries in the message logs can assist you in the following tasks:

- identifying the user associated with an XML request
- debugging the OSS requests from a user

You can enable XML message logging for an individual user or multiple users. See Procedure A-6 for more information. The 5620 SAM creates log files for each HTTP request and response associated with the 5620 SAM-O user defined by the <systemOssiLog> element in the nms-server.xml file. An HTTP request and response can contain multiple XML requests and responses.

The 5620 SAM stores the log file in the directory defined by the <systemOssiLog> element in the nms-server.xml file, typically .../log/. The 5620 SAM uses the following naming convention for log files:

- ossi<user>Request<n+>.log
- ossi<user>Response<n+>.log

  where <user> is the 5620 SAM user name and <n+> is the incremental request number

The request log contains the body of the SOAP message. The response log contains the entire SOAP envelope of the response.

Caution — The prolonged use of XML message logging in a busy network environment may have network management performance impacts. Use the log functionality for a limited time to debug a specific problem.

Code 10-9 shows a sample request for an XML message log file.

Code 10-9: Sample request from the ossiuserRequest1.log XML message log file

```
<?xml version='1.0'?>
<Request user="user" requestId="clientName@requestId">
  <find>
    <fullClassName>...</fullClassName>
    <filter>
      ...
    </filter>
  </find>
</Request>
```

Code 10-10 shows sample response to the request for an XML message log file.

Code 10-10: Sample response from the ossiuserResponse1.log XML message log file

```
<?xml version='1.0'?>
<Response user="user" requestId="clientName@requestId">
    <SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
        <SOAP:Header>
            <header xmlns="xmlapi_1.0">
                <requestID>clientName@requestId</requestID>
                <requestTime>Jan 2, 2007 3:10:41 PM</requestTime>
                <responseTime>Jan 2, 2007 3:10:41 PM</responseTime>
            </header>
        </SOAP:Header>
        <SOAP:Body>
          <findResponse xmlns="xmlapi_1.0">
            <result>
              ...
            </result>
          </findResponse>
        </SOAP:Body>
    </SOAP:Envelope>
</Response>
```

Note — The <responseTime> tag in a 5620 SAM-O header is the time at which the response stream is opened.

The system administrator configures the XML message log using the nms-server.xml file in the install directory. The system administrator is responsible for the maintenance and backup of log files.

Caution — Modifying the nms-server.xml file can seriously affect the network management and performance of the 5620 SAM.

As part of debug logging policy configuration, the system administrator can configure the maximum disk space for storing log messages. The 5620 SAM disables the log option and raises an alarm if the size of the log file exceeds the storage allocation. The default log file retention period is 24 hours. See Procedure A-6 for information on how to modify the log file retention period.

In a redundant network configuration, the 5620 SAM logs the activities for the active server.

See Procedure A-6 for information on how to enable the logging of XML message transactions on the 5620 SAM server. You can also use the 5620 SAM GUI client to view activity for a 5620 SAM-O user. The 5620 SAM GUI client tracks the network management activity for GUI and OSS users. See the *5620 SAM User Guide* for more information.

10.2 CLI commands within XML methods

The 5620 SAM client can send CLI commands to a node using methods in the network element object. The client receives the corresponding response for the executed CLI commands. Table 10-6 describes the methods in the network element object.

Table 10-6 CLI methods in the network element object

Method	Description
executeCli	Sends a CLI command to the node. See Code 10-11 for an XML example.
executeMultiCli	Sends multiple CLI commands to the node. See Code 10-12 for an XML example.

Code 10-11: Sample single CLI command

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope>

    <SOAP:Header>

        <header xmlns="xmlapi_1.0">
            <security>
                <user>user</user>
                <password>MD5-hashed user password</password>
            </security>
            <requestID>clientName@requestId</requestID>        </header>
    </SOAP:Header>
    <SOAP:Body>
        <netw.NetworkElement.executeCli
          xmlns="xmlapi_1.0">

            <instanceFullName>network:10.1.186.183</instanceFullName>
 <command>ping 138.120.186.185</command>
        </netw.NetworkElement.executeCli>
    </SOAP:Body>
</SOAP:Envelope>
```

Code 10-12: Sample multiple CLI commands

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope>
```

```
    <SOAP:Header>

        <header xmlns="xmlapi_1.0">
            <security>
                <user>user</user>
                <password>MD5-hashed user password</password>
            </security>
            <requestID>clientName@requestId</requestID>        </header>
    </SOAP:Header>
    <SOAP:Body>
        <netw.NetworkElement.executeMultiCli
          xmlns="xmlapi_1.0">

            <instanceFullName>network:10.1.186.183</instanceFullName>
<commands>
<string>ping 138.120.186.185</string>
<string>ping 138.120.186.184</string>
</commands>
        </netw.NetworkElement.executeMultiCli>
    </SOAP:Body>
</SOAP:Envelope>
```

10.3 Mapping XML methods to GUI operations

In development environments, you can configure the server to record the methods related to 5620 SAM GUI operations in the server log. The additional log information helps:

- developers understand which methods should be used to integrate an OSS application. You can use the server log file to trace the classes and methods called when an action is performed on the GUI.
- Alcatel-Lucent support staff troubleshoot the operation of the 5620 SAM

Procedure 10-1 To enable logging of GUI operations on the 5620 SAM server

Caution — The logging of GUI operations should only be enabled on servers in a development environment. When you enable the server log on a server that is running in live network environment, server performance may be affected.

1 Open the nms-server.xml file on the 5620 SAM server. The file is located in the *install directory*/nms/config directory, where *install directory* is the installation directory.

2 Find the <Filter> element within the <systemLog> element.

3 Add the following code directly after the line that starts with <!-- Don't change this section END -->.

```
<include severity="debug" class=".*IFGBase"
method="printDebug.*"/>
```

Caution — Do not modify other nms-server.xml parameters. Modifying the file can seriously affect the network management and performance of the 5620 SAM.

4 Restart the 5620 SAM server. All actions executed on the GUI and the XML OSS interface are now logged to the server log file.

5 Open the EmsServer.log debug log file. The log file is located in the *install directory*/nms/log/server directory, where *install directory* is the installation directory.

Procedure 10-2 To view the GUI operation in the server log

1 Complete Procedure 10-1.

2 Open the server log file.

3 Perform an action on the GUI.

4 View the logged information for that action in the server log file to determine the class and method used.

Output format of a GUI operation in the server log file

Code 10-13 shows the output format of a GUI operation in the server log file after you delete an IP interface on the GUI.

Code 10-13: Sample server log output

```
<server.core.IFGBase.printDebug>
IFG[generic]:([generic.GenericObject].deleteInstance() : BEGIN
<server.core.NullDeployer.issueDatabaseLog> From RIWorker
[8](1301414107317),DatabaseLogRuleInvokationCapsule User Id:
securityManager:user-admin Request Id:
AreqGenericObjectDeleteInstance-CLIENT-admin-5620SAM@6-181 Target Class:
rtr.DirectInterfaceCtp Target Object Id:
network:35.121.20.59:router-1:ip-interface-2:TP Operation: ObjectDeletion
Result: true
<server.core.IFGBase.printDebug>
IFG[generic]:([generic.GenericObject].deleteInstance() : END ::: successfuly
executed : true
```

Code 10-13 contains the following information:

- the class and method that corresponds to the GUI action, which in this example is generic.GenericObject.deleteInstance. You can use the identified class and method to create an OSS application that performs the same task.
- the specific instance of the action that was on the managed router IP interface with the unique ID network:35.121.20.59:router-1:ip-interface-2:TP. The unique ID identifies that the router has an IP address of 35.121.20.59 and a name of router-1.
- acknowledgement of successful completion of the configuration action

Note 1 — In some cases, a GUI action may use custom methods that are not directly applicable for use by an OSS. For example, when configuring objects using the GUI, the server log output will log the object.configure method. For those cases, an OSS should use the generic methods. See section 16.2 for more information on using generic methods for configuring network objects.

Note 2 — In some cases, no specific instance is returned, for example, when you perform a find action.

10.4 XML samples

Contact your Alcatel-Lucent OIPS technical support representative to obtain XML samples from the 5620 SAM-O SDK library. See Appendix C for more information.

5620 SAM-O common tools and methods

11 — 5620 SAM-O core methods

12 — Script management

11 — 5620 SAM-O core methods

11.1 Core methods overview *11-2*

11.1 Core methods overview

Table 11-1 describes the core methods that a 5620 SAM-O client can use to perform specific actions.

Table 11-1 Core XML methods

Method	Description	Input parameters	Output parameters
version	To return the version of the 5620 SAM server that is currently being used.	—	• **version** The version of the server (string) • **baseVersion** The major.minor version of the server (string) • **build** The build of the server (string) • **patch** The patch level of the server (string)
ping	To check the connectivity of the 5620 SAM server.	—	—
timeStamp	To view the current Java time on the 5620 SAM server, in milliseconds. The execution time of a command or group of commands can be determined by surrounding them with timeStamp commands and taking the delta of their return values.	• **includeFormatted (optional)** Include formatted timestamp in the response (Boolean)	• **millis** The difference in milliseconds between the current time and midnight, Jan 1, 1970 UTC (long) • **timeStamp** The formatted time stamp (string)
find	To return the set of objects of the specified type which match the given filter criteria.	• **fullClassName** Package qualified class name in dot-separated format • **filter (strongly recommended)** Filter which objects are returned based on the properties of the class specified by fullClassName • **timeout (optional)** Specifies the time, in milliseconds, after which the find operation times out. The request is aborted and an exception is returned. • **resultFilter (strongly recommended)** Filter which attributes are returned for each object	• **result** (generic.CommonManagedEntityInformation) The list of objects which match the given filter criteria.

(1 of 3)

Method	Description	Input parameters	Output parameters
findToFile	To find the set of objects of the specified type which match the given filter criteria, and places the results in the specified file on the server. Upon completion of the command, a fileAvailable event is generated.	• **fullClassName** Package qualified class name in dot-separated format • **fileName** The relative file in which to store the results of this find operation. The format is: [s]ftp://[*user:password@*]host[*:port*]/directory/file The following rules must be applied to the format: • When using a remote client, the directory must be relative to the default FTP directory for the remote client. • The characters '/' and ';' are reserved and must be encoded. For example, to specify a file in /tmp directory on the remote client, the path must be encoded as ftp://name@password/%2Ftmp/filename.xml. The path ftp://name@password/tmp/filename.xml specifies a file in a /tmp directory under the default FTP directory of the user. See *RFC 1738 Uniform Resource Locators (URL), Section 3.2.2* for more information on the FTP URL path requirements. • **timeStamp (optional)** Includes the timestamp into the filename of the saved file • **timeout (optional)** Specifies the time, in milliseconds, after which the findToFile operation times out. The request is aborted and an exception is returned. The file may be partially written and may not be valid. An exception indicating a timeout has occurred will be present in the JMS FileAvailableEvent also. • **filter (strongly recommended)** Filter on properties of the class corresponding to fullClassName • **resultFilter (strongly recommended)** Filter for narrowing down the information returned per object • **synchronous (optional)** Specifies that the method is run synchronously and that the result is returned only when the find is complete. The default is true.	—

(2 of 3)

Method	Description	Input parameters	Output parameters
deregisterLogToFile	To stop the creation of accounting stats files.	• **fullClassName (optional)** Package qualified class name in dot separated format. If this parameter is not specific then it deregister for all type of accounting stats (string) • **jmsClientId** The JMS Client ID (string)	—
registerLogToFile	To create accounting stats file based on the specified class type. Accounting stats Files are created and are placed in the specified directory on the server. Each time a file is created, a LogFileAvailableEvent event is generated. If there is no JMS client subscribed within a given time, the 5620 SAM server degregisters the request.	• **fullClassName** Package qualified class name in dot separated format. A comma separated list of accounting statistics classes (string) • **dirName** The relative path to the subdirectory where the files are to be saved. The path is relative to the OSS XML export directory/accountingStats. The use of separate directories for different applications that export statistics is recommended. • **compress** Specifies whether export files should be compressed when they are created (optional) • **jmsClientId** The JMS Client ID that is notified when a file is created (string) • **resultFilter** Specifies the attributes that are to be included in the exported data records. You can use the resultFilter parameter to reduce the size of the export file (optional) <attribute>*attribute_name*</attribute>	—

(3 of 3)

12 — Script management

12.1 Script management *12-2*

12.2 Workflow to execute a script *12-2*

12.3 Sample script management requests *12-3*

12.1 Script management

You can manage scripts of CLI commands and XML API commands that are used to configure selected network devices and to view configuration information using an OSS client. The OSS client can use the 5620 SAM script manager for the following:

- testing XML scripts
- XML script and OSS software troubleshooting

See the *5620 SAM Scripts and Templates Developer Guide* for more information about the script manager.

Warning — CLI and XML API scripts that are not correctly applied or created can cause serious damage to the network. Alcatel-Lucent recommends that system administrators clearly define user responsibilities for script usage, and ensure that scripts are verified and validated before they are executed on devices in a live network.

See the appropriate device hardware documentation for the applicable CLI commands, structure, and usage.

12.2 Workflow to execute a script

1. Ensure that the mediation policy for the target device(s) is configured with the correct user name and password for CLI communication.
2. For secure scripting, ensure that SSH2 is properly configured on SSH2-capable devices. Ensure that the mediation policy for the target device(s) is configured with SSH2 for CLI communication.
3. To use the script manager with generic NEs, ensure that the generic network element profile is configured with the correct CLI read-write access login prompts, and connection information.

Note — All script property values, including default values, must be included when you define the parameters for a script.

4. Create and execute the script.
   - **i** Create the script. See “Sample request to create a CLI script” and “Sample request to create an XML API script”.
   - **ii** Create the script version. See “Sample request to create a CLI script version” and “Sample request to create a XML API script version to configure a mediation policy”.
   - **iii** Associate the target NEs with the CLI script version and specify any input parameter values within the script content. See “Sample request to create a script target for a CLI script” and “Sample request to create a script instance for an XML API script”.

iv Execute the selected set of script targets. See "Sample request to execute a script target".

v Retrieve the scripts results using FTP or SOAP/XML. See "Sample request to retrieve script results".

You can synchronously or asynchronously execute the scripts. For synchronous execution, a response attribute identifies the success or failure of script execution. For asynchronous execution, the OSS client can either wait for a JMS notification or poll for status on the execution of the script.

The script results are stored in a file for the OSS client to collect. For scalability reasons, the file is removed after 10 minutes. When the OSS client retrieves the results using the SOAP interface, the result is stripped of non-printable characters embedded in the response.

12.3 Sample script management requests

The following are sample requests to create, execute and retrieve results from CLI and XML API scripts.

Code 12-1: Sample request to create a CLI script

```
 <SOAP:Body>
    <script.ScriptManager.configure xmlns="xmlapi_1.0">
      <deployer>immediate</deployer>
      <synchronousDeploy>true</synchronousDeploy>
      <configInfo>
        <script.Script>
          <actionMask>create</actionMask>
          <contentType>CLI</contentType>
          <neType>Alcatel-SR-7750</neType>
<mtosi_description>create-CLI-Script-description</mtosi_description>
          <mtosi_id>21</mtosi_id>
          <mtosi_type>create-CLI-Scrip-type</mtosi_type>
          <scriptName>create-CLI-Script-1</scriptName>
          <useLatestVersion>true</useLatestVersion>
        </script.Script>
      </configInfo>
    </script.ScriptManager.configure>
  </SOAP:Body>
```

Code 12-2: Sample request to create an XML API script

```
<SOAP:Body>
    <script.ScriptManager.configure xmlns="xmlapi_1.0">
      <deployer>immediate</deployer>
      <synchronousDeploy>true</synchronousDeploy>
      <configInfo>
        <script.XmlApiScript>
          <actionMask>create</actionMask>
<mtosi_description>create-velocity-XmlApiScript-description</mtosi_d
escription>
          <mtosi_id>23</mtosi_id>
          <mtosi_type>create-velocity-XmlApiScript-type</mtosi_type>
          <scriptName>create-velocity-XmlApiScript-23</scriptName>
```

```
          <useLatestVersion>true</useLatestVersion>
        </script.XmlApiScript>
      </configInfo>
    </script.ScriptManager.configure>
  </SOAP:Body>
```

Code 12-3: Sample request to create a CLI script version

```
 <SOAP:Body>
   <script.AbstractScript.createScriptVersion  xmlns="xmlapi_1.0">
     <deployer>immediate</deployer>
     <synchronousDeploy>true</synchronousDeploy>
     <instanceFullName>script-manager:script-21</instanceFullName>
     <versionConfigInfo>
       <script.Version>
         <actionMask>create</actionMask>
         <mtosi_id>0</mtosi_id>
         <comment>create-cliScript-version</comment>
       </script.Version>
     </versionConfigInfo>
     <bodyScript><string>exit all</string>
       <string>show service id $serviceId$ $option$(default
sap)</string>
       <string>exit all</string>
     </bodyScript>
   </script.AbstractScript.createScriptVersion>
 </SOAP:Body>
```

Code 12-4: Sample request to create a XML API script version to configure a mediation policy

```
 <SOAP:Body>
   <script.AbstractScript.createScriptVersion  xmlns="xmlapi_1.0">
     <deployer>immediate</deployer>
     <synchronousDeploy>true</synchronousDeploy>
<instanceFullName>script-manager:xmlApiScript-23</instanceFullName>
     <versionConfigInfo>
       <script.XmlApiVersion>
         <actionMask>create</actionMask>
         <mtosi_id>0</mtosi_id>
         <comment>test</comment>
         </script.XmlApiVersion>
       </versionConfigInfo>
<bodyScript>
<string>
<![CDATA[
<xmlapiRequest xmlns="xmlapi_1.0">
#*
<velocityProperties>
 <tab><name>General</name><tooltip>The general tab</tooltip>
   <group><name>General</name><tooltip>The general group</tooltip>
     <property>
       <name>name</name>
       <uiName>Profile Name:</uiName>
       <type>String</type>
       <default>private</default>
       <tooltip>The unique name of the profile</tooltip>
```

```
        <required>true</required>
      </property>
      <property>
        <name>username</name>
        <uiName>User Name:</uiName>
        <tooltip>The user name</tooltip>
        <type>String</type>
        <default>admin</default>
        <required>true</required>
      </property>
      <property>
        <name>password</name>
        <uiName>Password:</uiName>
        <tooltip>The password</tooltip>
        <type>String</type>
        <default>admin</default>
        <required>true</required>
        <min>3</min>
        <max>10</max>
      </property>
      <property>
        <name>id</name>
        <uiName>Unique ID:</uiName>
        <tooltip>The unique id</tooltip>
        <type>Range</type>
        <default>1001</default>
        <min>3</min>
        <max>10000</max>
      </property>
    </group>
  </tab>
</velocityProperties>
*#
<generic.GenericObject.configureInstanceWithResult
xmlns="xmlapi_1.0">
   <distinguishedName>pollerManager</distinguishedName>
   <includeChildren>false</includeChildren>
   <deployer>immediate</deployer>
   <configInfo>
       <security.MediationPolicy>
           <actionMask><bit>create</bit></actionMask>
           <displayedName>$name</displayedName>
           <cliPassword>$password</cliPassword>
           <cliUserName>$username</cliUserName>
           <community>private</community>
           <id>$id</id>
           <securityModel>snmpv2c</securityModel>
           <snmpRetry>1</snmpRetry>
           <snmpTimeout>10000</snmpTimeout>
       </security.MediationPolicy>
  </configInfo>
</generic.GenericObject.configureInstanceWithResult>
</xmlapiRequest>
]]>
</string>
</bodyScript>
     </script.AbstractScript.createScriptVersion>
```

```
   </SOAP:Body>
```

Code 12-5: Sample request to create a script target for a CLI script

```
<SOAP:Body>
   <script.AbstractScript.createTarget xmlns="xmlapi_1.0">
     <deployer>immediate</deployer>
     <synchronousDeploy>true</synchronousDeploy>
     <instanceFullName>script-manager:script-21</instanceFullName>
     <configInfo>
       <script.TargetScript>
         <actionMask>create</actionMask>
         <mtosi_id>0</mtosi_id>
         <mtosi_version>0</mtosi_version>
<scriptDisplayName>create-cli-Script-target</scriptDisplayName>
         <targetName>create-cli-Script-target</targetName>
<targetObjectFullName>network:10.1.1.48</targetObjectFullName>
         <script.TargetParameter-Set>
           <script.TargetParameter>
             <actionMask>create</actionMask>
             <parameterName>serviceId</parameterName>
             <value>7</value>
           </script.TargetParameter>
         </script.TargetParameter-Set>
       </script.TargetScript>
     </configInfo>
   </script.AbstractScript.createTarget>
   </SOAP:Body>
```

Code 12-6: Sample request to create a script instance for an XML API script

```
 <SOAP:Body>
   <script.AbstractScript.createTarget xmlns="xmlapi_1.0">
     <deployer>immediate</deployer>
     <synchronousDeploy>true</synchronousDeploy>
<instanceFullName>script-manager:xmlApiScript-23</instanceFullName>
     <configInfo>
       <script.TargetScript>
         <actionMask>create</actionMask>
         <mtosi_id>0</mtosi_id>
         <mtosi_version>0</mtosi_version>
<scriptDisplayName>create-XmlApiScript-target</scriptDisplayName>
         <targetName>create-XmlApiScript-target</targetName>
<targetObjectFullName>network:10.1.1.48</targetObjectFullName>
         <script.TargetParameter-Set>
           <script.TargetParameter>
             <actionMask>create</actionMask>
             <parameterName>name</parameterName>
             <value>secondParameterName</value>
           </script.TargetParameter>
           <script.TargetParameter>
             <actionMask>create</actionMask>
             <parameterName>username</parameterName>
             <value>myOtherName</value>
           </script.TargetParameter>
           <script.TargetParameter>
             <actionMask>create</actionMask>
```

```
            <parameterName>password</parameterName>
            <value>myOtherPassword</value>
          </script.TargetParameter>
          <script.TargetParameter>
            <actionMask>create</actionMask>
            <parameterName>id</parameterName>
            <value>122</value>
          </script.TargetParameter>
        </script.TargetParameter-Set>
      </script.TargetScript>
    </configInfo>
  </script.AbstractScript.createTarget>
  </SOAP:Body>
```

Code 12-7: Sample request to execute a script target

```
<SOAP:Body>
    <script.TargetScript.executeScript xmlns="xmlapi_1.0">
     <isSynchronized>true</isSynchronized>
<instanceFullName>script-manager:script-21:target-script-6</instance
FullName>
     <clientId>oips-test@21</clientId>
    </script.TargetScript.executeScript>
    </SOAP:Body>
```

Code 12-8: Sample request to retrieve script results

```
<SOAP:Body>
      <script.ScriptManager.getBodyOfTargetScriptResult
xmlns="xmlapi_1.0">
<targetScriptResultObjectFullName>script-manager:script-21:target-sc
ript-6:target-script-result-1170360659965</targetScriptResultObjectF
ullName>
      </script.ScriptManager.getBodyOfTargetScriptResult>
    </SOAP:Body>
```

OSS domains

13 — Fault management

14 — Inventory management

15 — Accounting and performance monitoring

13 — Fault management

13.1 Fault management overview *13-2*

13.2 Workflow to configure alarm and event management *13-5*

13.3 OAM overview *13-8*

13.1 Fault management overview

The 5620 SAM converts SNMP traps from network routers to events and alarms. The 5620 SAM correlates the events and alarms against the managed equipment and configured services and policies. The 5620 SAM applies the alarms against the appropriate equipment and services.

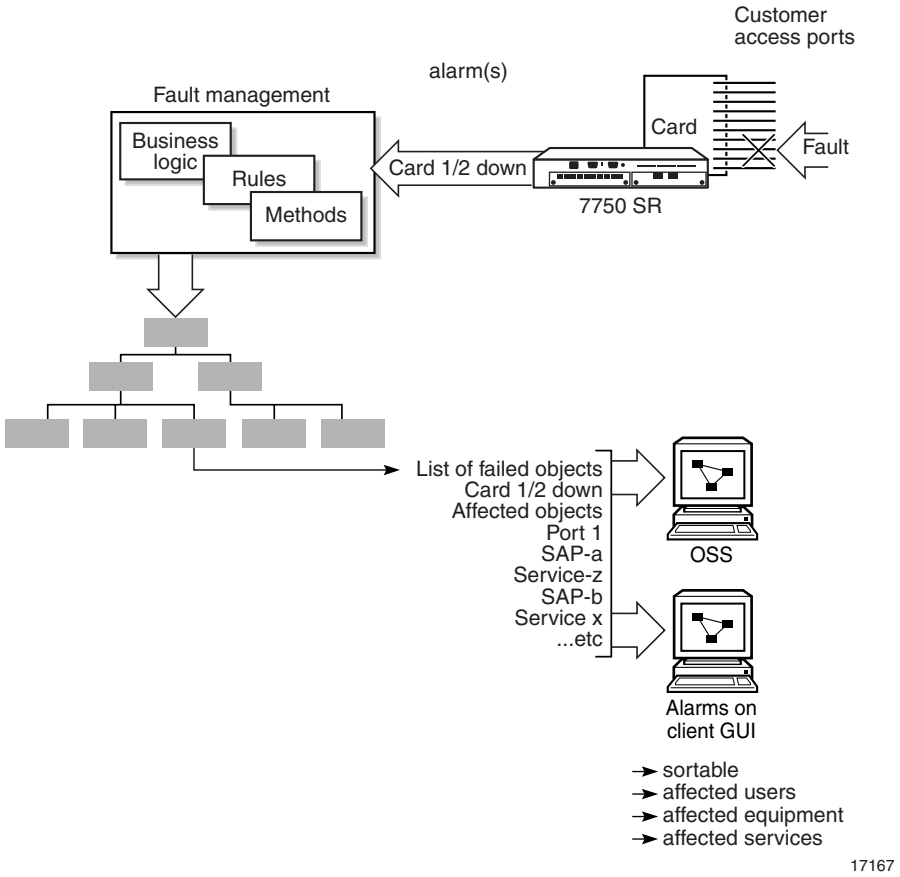
The alarm and event management system provides:

- impact analysis and correlation of alarms to equipment and service-affecting faults
- updated operational status of equipment, services, and interfaces in near-real-time from the network router resources
- alarm policy control by network administrators so that the administrators can determine how to handle individual incoming alarms, and how alarm logs are created and stored
- operator notes and acknowledgements to track the solution for the problem that caused the alarm
- correlated data in a historical alarm database to provide trend analysis and records

A correlated alarm refers to an alarm that causes fault conditions for many objects. For example, if an alarm occurs because a port goes down, all services that use the port receive notification of the alarm. You can view the alarm from the service configuration form or from the subscriber information form that lists the affected service. All object information forms contain a faults tab, which lists the alarms that affect the object. All alarms appear in the alarm list of the 5620 SAM GUI.

Figure 13-1 shows the method used by the alarm management system to handle alarms.

Figure 13-1 Alarm handling



Note 1 — See the *5620 SAM Troubleshooting Guide* for more information about the alarm elements and types that are associated with the 5620 SAM-O fault management package.

Note 2 — You can find specific alarm descriptions in the following locations:

- *Installation_directory*/nms/User_Documentation/5620_SAM-O_documentation/XML_Reference/alarmDetails.xml, where *Installation_directory* is the client installation folder or directory
- User_Documentation/5620_SAM-O_documentation/XML_Reference/alarmDetails.xml directory on the product DVD-ROM
- http://*server address*:8085/5620 SAM-O documentation/XML Reference/alarmDetails.xml, where *server address* is the address of the server

Alarm management requests

The OSS application uses a JMS and/or an XML/SOAP interface to communicate with the server. For alarm management:

- JMS receives network event notifications and current alarms from the 5620 SAM
- XML/SOAP retrieves historical alarms from the 5620 SAM

See chapter 4 for more information about JMS near-real-time events and alarms.

OSS client alarm testing

You can use the fm.FaultManager.testAlarm method to trigger an alarm generated by the 5620 SAM and sent to the OSS client using JMS. The fm.FaultManager.testAlarm method allows you to generate alarms using the 5620 SAM and facilitates alarm testing using an OSS client. The fm.FaultManager.testAlarm method allows you to manually generate alarms instead of using equipment traps to trigger the alarm condition.

Note 1 — You must ensure that you clear the alarms associated with the fm.FaultManager.testAlarm method after you complete your alarm testing session.

Note 2 — The fm.FaultManager.testAlarm method does not perform a validation to ensure that you are raising an alarm with the correct parameters against the correct object class. Additionally, the raised alarm will not have any correlation, aggregation, or association relationships with other objects or alarms.

To create an alarm that simulates a real alarm, you must include the following alarm attributes when you use the fm.FaultManager.testAlarm method:

- objectInstanceName
- alarmNameId
- alarmTypeId
- probableCauseId
- severity
- alarmClassTag

If the correct alarm attributes are used, the 5620 SAM generates alarms similar to real alarms that are triggered by network events. See the *5620 SAM Troubleshooting Guide* for more information about alarm attributes.

The fm.FaultManager.testAlarm method can be used in a development environment to test certain types of alarms that require real NEs, where real NEs are not accessible. It is an effective means to generate alarms and to verify the alarm response output received by an OSS client.

Code 13-1 shows an example of how to generate a service site down alarm.

Code 13-1: Sample request to generate a service site down alarm

```
<SOAP:Body>
   <fm.FaultManager.testAlarm xmlns="xmlapi_1.0">
     <deployer>immediate</deployer>
```

```
      <objectInstanceName>svc-mgr:service-3:10.1.1.90</objectInstanceName>
      <alarmNameId>97</alarmNameId>
      <alarmTypeId>17</alarmTypeId>
      <probableCauseId>84</probableCauseId>
      <severity>critical</severity>
      <alarmClassTag>svc.ServiceSiteDown</alarmClassTag>
      <namingComponent/>
      <additionalText>my additional text</additionalText>
    </fm.FaultManager.testAlarm>
  </SOAP:Body>
```

Code 13-2 shows an example of the response to the request to generate a service site down alarm.

Code 13-2: Sample response to the request to generate a service site down alarm

```
<SOAP:Body>
     <fm.FaultManager.testAlarmResponse xmlns="xmlapi_1.0"/>
  </SOAP:Body>
```

13.2 Workflow to configure alarm and event management

Use the following steps to generate requests for alarm and event management information.

1. Retrieve a snapshot of the network elements for which you want to generate alarm and event management requests. See chapter 14 for more information about inventory retrieval.

2. Obtain a current alarm list. This can be done by:

   a. Using the generic OSS <find> method, as defined in the genericMethods.xsd file. The generic OSS <find> method does not retrieve <relatedObjects>, <affectingObjects>, or <aggregatedObjects>.

   b. Using the fm.FaultManager.findFaults method. See Appendix C for information on how to obtain the following sample alarm query requests:

      - request to return a list of alarms
      - request to return a list of alarms based on a filter

Note — To retrieve <relatedObjects> and <affectingObjects> on an object, use the fm.FaultManager.findObjectsAffectingOfn or fm.FaultManager.findObjectsAffectingOfn method respectively.

To request a list of alarms by category on an object, use the fm.FaultManager.findAlarmsForOfn method and filter the list as desired.

To obtain the alarm and aggregated alarm status on an object, use the fm.FaultManager.findAlarmsStatusesForOfn method.

See the XML schema fmMethods.xsd for more information about these methods.

3. Configure alarm policies, if required. Code 13-3 shows a sample request to configure an alarm policy.

Code 13-3: Sample request to configure the equipment.LinkDown alarm policy

```
<?xml version="1.0" encoding="UTF-8"?>
  <SOAP:Envelope  xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP:Header>
      <header xmlns="xmlapi_1.0">
        <requestID>clientName@requestId</requestID>
      </header>
    </SOAP:Header>
    <SOAP:Body>
      <generic.GenericObject.configureInstanceWithResult xmlns="xmlapi_1.0">
        <deployer>immediate</deployer>
        <distinguishedName>faultManager:globalPolicy:equipment.LinkDown</dis
tinguishedName>
        <includeChildren>true</includeChildren>
        <configInfo>
          <fm.SpecificPolicy>
            <actionMask>
              <bit>modify</bit>
            </actionMask>
            <autoAssignedSeverity>critical</autoAssignedSeverity>
          </fm.SpecificPolicy>
        </configInfo>
      </generic.GenericObject.configureInstanceWithResult>
    </SOAP:Body>
  </SOAP:Envelope>
```

Note — There is one global alarm policy and a default list of specific alarm policies. Specific alarm policies can be modified but cannot be created or deleted.

4 Establish a JMS connection.

Note — The 5620 SAM limits the number of concurrent JMS connections from OSS applications to 10 connections.

i Compile a valid JMS client *.java file. See Procedure 4-3 for more information.

The received component of a Java classpath is samOss.jar.

ii Run the compiled program. The components required for running the file and receiving the near-real-time event stream are:

- samOss.jar
- the IP address of the server
- the port number of the server
- the user name and password of the JMS client

Warning 1 — The OSS client must use the correct the samOss.jar that corresponds with the version of the 5620 SAM that you are using. Using the incorrect samOss.jar can cause the 5620 SAM server or client to become unstable. The 5620 SAM release information is in the JAR manifest file.

Warning 2 — The OSS client must use the standard JMS 1.1 API interfaces (javax.jms.*), and not JBoss implementations, to avoid compatibility issues.

Note 1 — You can find the samOss.jar on the 5620 SAM installation DVD-ROM in the /integration/SAM_O directory. The samOss.jar is installed on the 5620 SAM server in the /opt/5620sam/server/nms/integration/SAM_O directory. Contact your Alcatel-Lucent OIPS technical support representative for more information.

Note 2 — Since the jar files are compiled with Java 1.6, the OSS client must run a minimum of Java 1.6 JVM.

5 Process the incoming stream of network events and alarms for your network purposes.

6 Acknowledge, clear, or remove the alarms, as required. Specify a valid method for the request based on those listed and defined in the fmMethods.xsd file.

See Appendix C for information on how to obtain the following sample requests that use the objects and methods associated with the fm package:

- request to acknowledge a fault on an existing alarm
- request to clear faults
- request to remove faults

7 In the event of a connection failure, repeat steps 1 to 6.

13.3 OAM overview

The 5620 SAM-O uses the sas package to group various OAM tests into test suites for network troubleshooting and to verify compliance with SLAs. You can schedule the periodic execution of a test suite to provide continual performance feedback, or run a test suite on demand to investigate service issues.

The grouping of tests into a test suite enables a 5620 SAM-O operator to use one schedule for the periodic execution of multiple OAM diagnostics against an object in the network, for example, a service, device, or network transport element. A 5620 SAM-O operator can choose to include existing tests or use the 5620 SAM-O to generate the tests that comprise a test suite.

The sas package contains methods to invoke service assurance tests. Specific test definitions are defined in the package of the tested objects. The following is a list of test types and the associated packages:

- LSP Ping, LSP Trace, and LDP Tree Trace are defined in the mpls package
- ATM Ping is defined in the atm package
- Service Site Ping, MAC Ping, MAC Trace, MAC Populate, MAC Purge. MEF MAC Ping, and CPE Ping are defined in the service package
- VPRN Ping and VPRN Trace are defined in the vprn package
- Tunnel Ping and MTU Ping are defined in the svt package
- ICMP Ping, ICMP trace, and DNS Ping are defined in the icmp package
- Multicast router information, multicast trace, and multicast FIB Ping are defined in the multicast package
- ANCP Loopback is defined in the ancp package
- CFM Continuity Check, CFM Loopback, and CFM Link Trace are defined in the ethernetoam package

See section 8.1 for more information about the XML package types. See the *5620 SAM User Guide* for more information about the individual test types.

Note — The 5620 SAM-O user must be assigned the appropriate scope of command roles, including OSS Management and Service Test Management, to create and run service tests. In addition, 5620 SAM-O users are assigned to a user group that is configured with span of control profiles that assign access rights to configured 5620 SAM test objects. See the *5620 SAM User Guide* for more information about managing security on 5620 SAM user groups and the associated scope of command roles, permissions, and span of control.

Test policies

To enable the automatic generation of tests for a test suite, the 5620 SAM-O requires a test policy that contains a set of test definitions and pre- and post-processing rules. A test policy also specifies the order of execution for the generated tests. A test policy applies to only one test suite, and a test suite can have only one test policy.

A test policy is specific to one type of entity, for example, a VLL service or a service tunnel. The test definitions in the policy are restricted to the tests that apply to the entity type specified in the policy. A test policy is applied to a test suite during test suite creation.

Test suites

A test suite contains three types of test groups:

- first-run tests
- generated tests
- last-run tests

First-run tests are preliminary tests that are executed before tests in the other two groups. They typically include high-level diagnostic tests, for example, a service site ping, VPRN ping, or VPRN trace.

Generated tests are created by the 5620 SAM-O for use with a particular service, based on the topology of the service and the information specified in the associated test policy. For example, a policy that specifies a service site ping as a test definition is associated with a test suite for a VPRN service that has three sites. The 5620 SAM-O generates three tests—one service site ping for each site in the service.

Last-run tests are final tests that are executed after the first-run and generated tests. They typically include transport layer diagnostics, for example, an LSP trace or a tunnel ping.

OAM service validation

You can generate OAM validation tests to verify the operational status of a service or service tunnel. The operational status of a service depends on the operational states of its service sites or instances. It is possible for a service instance to be operationally up when communication between sites is not operational. For example, a VRF can be operationally up but the routes to its peers might not be populated because of the routing policy, route target, or ACL configuration. The operational state cause of the service or service tunnel indicates the success or failure of the OAM validation test, and therefore, service connectivity.

Note — OAM service validation is not supported for HVPLS.

Set the sasValidator flag to true when you create or modify the sas.TestSuite to specify whether the test suite is used as a validator of its tested entities. A service can be bound to only one test suite with the sasValidator flag set to true. The sasValidator flag is equivalent to the Tested Entity Status Affected by Results parameter on the 5620 SAM GUI. See the *5620 SAM User Guide* for more information about configuring OAM validation test suites.

To execute the validation, use the sas.TestManager.executeValidator method with a specified list of sas.TestEntity; only tested entities with the sasValidator flag set to true are validated.

The results of the OAM validation are indicated by the oamValidationFailed bit of the operationalFlags attribute of the associated service.Service or svt.Tunnel class.

14 — Inventory management

14.1 Inventory retrieval *14-2*

14.2 5620 SAM object hierarchy *14-10*

14.3 Filters *14-20*

14.1 Inventory retrieval

The 5620 SAM-O root package contains types and methods that allow you to gather large amounts of data from the 5620 SAM server or database. Figure 14-1 shows an OSS request for read information from the database.

Figure 14-1 Read from database interaction

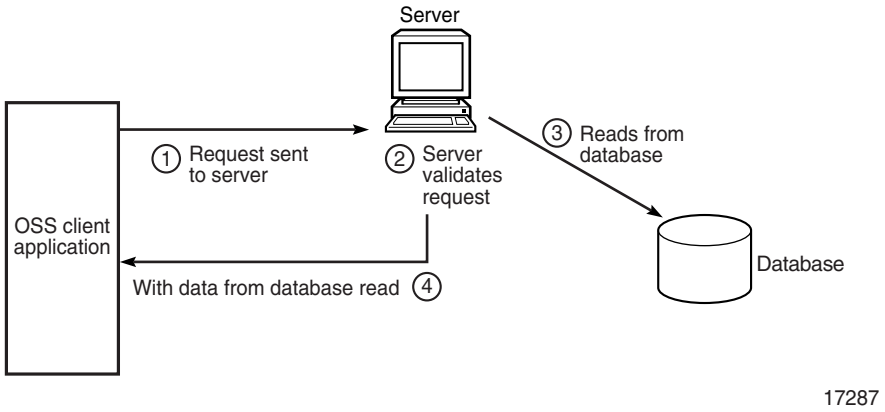


Table 14-1 identifies the XML methods that an OSS client can use to retrieve the objects.

Table 14-1 XML methods for object retrieval

Method	Input parameters	
find	fullClassName	Package qualified class name in dot-separated format
	filter (strongly recommended)	Filter on properties of the class corresponding to fullClassName
	timeout (optional)	Specifies the time, in milliseconds, after which the find operation times out. The request is aborted and an exception is returned.
	resultFilter (strongly recommended)	Filter for narrowing down the information returned per object
findToFile	fullClassName	Package qualified class name in dot-separated format
	fileName	The relative file in which to store the results of this find operation. The format is: [s]ftp://[*user:password@*]host[:*port*]/directory/file The following rules must be applied to the format: • When using a remote client, the directory must be relative to the default FTP directory for the remote client. • The characters '/' and ';' are reserved and must be encoded. For example, to specify a file in /tmp directory on the remote client, the path must be encoded as ftp://name@password/%2Ftmp/filename.xml. The path ftp://name@password/tmp/filename.xml specifies a file in a /tmp directory under the default FTP directory of the user. See *RFC 1738 Uniform Resource Locators (URL), Section 3.2.2* for more information on the FTP URL path requirements.
	timeStamp (optional)	Includes the timestamp into the filename of the saved file.
	timeout (optional)	Specifies the time, in milliseconds, after which the findToFile operation times out.
	filter (strongly recommended)	Filter on properties of the class corresponding to fullClassName
	resultFilter (strongly recommended)	Filter for narrowing down the information returned per object
	synchronous (optional)	Specifies that the method is run synchronously and that the result is returned only when the find is complete. The default is true.

Note — You should retrieve inventory with the 5620 SAM-O using the <find> and <findToFile> methods defined in the root package. Some methods exist within the network and equipment packages to retrieve specific object classes. These methods are not recommended because they are not optimized.

<find> method

The <find> method returns a set of objects of the type specified in the <fullClassName> element and that match the filter criteria. The response for the method is in the form of a streamed XML result.

The <find> method is beneficial for queries of any size because of the compact system resources required to return query results.

Caution — Network performance can be adversely affected by numerous queries that gather large amounts of data.

For example, to gather the containment hierarchy of 50 000 services that have three sites, one access interface for each site, and three circuits requires more than 800 Mbytes of space to store the report, and can take many hours to complete.

Code 14-1 shows a sample request to find ACL IP filters.

Code 14-1: Sample request to find access control list IP filters

```
<SOAP:Body>
  <find xmlns="xmlapi_1.0">
    <fullClassName>aclfilter.IpFilter</fullClassName>
    <filter>
      <equal name="objectFullName" value="IP Filter:1000"/>
    </filter>
  </find>
</SOAP:Body>
```

<findToFile> method

The <findToFile> method returns a set of objects of the type specified in the <fullClassName> element and that match the filter criteria. You can use the following options to save the results:

- record the results to a file on the 5620 SAM server
- transfer the results by FTP to a remote server

The 5620 SAM server broadcasts a JMS message to all listeners of the topic when the <findToFile> results are complete for the request. The JMS message includes the URI of the result and the request ID. The JMS message is sent out through the JMS topic that publishes as XML, not the JMS topic that publishes an object. The JMS message header contains the client ID of the client that made the request. See section 4.2 for more information about JMS filters for client ID and global messages.

The <findToFile> method is beneficial for large queries because of the compact system resources required to return query results.

Code 14-2 shows a sample request to retrieve historical statistics from a router.

Code 14-2: Sample request to retrieve historical statistics from a router

```
 <SOAP:Body>
        <findToFile xmlns="xmlapi_1.0">
            <fullClassName>equipment.InterfaceAdditionalStatsLogRecord</full
ClassName>
            <filter>
                <and>
```

```
                    <equal name="monitoredObjectPointer"
value="network:10.1.202.93:shelf-1:cardSlot-1:card:daughterCardSlot-1:daught
erCard:port-3"/>
                    <between name="timeCaptured" first="1127142900000"
second="1127143800000"/>
                </and>
            </filter>
            <fileName>Equipment.InterfaceAdditionalStatsLogRecord.xml</fileN
ame>
        </findToFile>
    </SOAP:Body>
```

Local file storage of query results

The URL used to record the <findToFile> method responses is a relative path on the 5620 SAM main server. The path root is specified during 5620 SAM server installation.

Note — You can manage the volume of files that are stored locally by configuring the following attributes in the xmalapi section of the nms/config/nms-server.xml file on the 5620 SAM main server:

- xmlTimeToKeepFile—Specifies how long, in m, to keep each file created by findToFile requests. The range is 5 to 1440. The default is 15.
- xmlCheckDiskInterval—Specifies how often, in m, to check the file system for file purging. The range is 1 to 60. The default is 5.

Code 14-3 shows an example storage configuration in nms-server.xml.

Code 14-3: Sample file storage configuration

```
 <xmlapi
....
....
xmlTimeToKeepFile="1440"
xmlCheckDiskInterval="60" />
```

Code 14-4 shows an example of how to record the results to a file.

Code 14-4: Sample configuration of local file storage

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope
  xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">

    <SOAP:Header>
        <header xmlns="xmlapi_1.0">
            <security>
                <user>user name</user>
                <password>MD5-hashed password</password>
            </security>
            <requestID>clientName@requestId</requestID>
        </header>
    </SOAP:Header>

    <SOAP:Body>
        <findToFile xmlns="xmlapi_1.0">
            <fullClassName>equipment.PhysicalPort</fullClassName>
            <filter>
<and>
                    <equal name="siteId" value="10.1.202.93"/>
    <equal name="shelfId" value="1"/>
```

```
    <equal name="cardSlotId" value="1"/>
    <equal name="daughterCardSlotId" value="1"/>
    <equal name="mode" value="access"/>
</and>
            </filter>
    <fileName>equipmentPhysicalPort.xml</fileName>
        </findToFile>
    </SOAP:Body>
</SOAP:Envelope>
```

Remote file storage of query results

You can configure the <findToFile> method to record the query results to a file on a remote server using FTP. The fileName parameter for the <findToFile> method allows you to configure the user, server, and file information. The user and password are optional configuration components. The 5620 SAM-O uses the following substitute information if you do not supply a user or password for the FTP session:

- anonymous if you do not specify an FTP username
- an e-mail address if you do not specify the FTP password

Code 14-5 shows the configuration of the remote storage option for the <findToFile> method.

Code 14-5: Sample configuration of remote file storage

```
<findToFile xmlns="xmlapi_1.0">
    <synchronous>false</synchronous>
    <fullClassName><class name></fullClassName>
    <fileName>
    [s]ftp://[user[:password]@]host[:port]/directory/file
    </fileName>
    <resultFilter>
.
.
.
    </resultFilter>
</findToFile>
```

Synchronous and asynchronous requests

By default, the <findToFile> method runs synchronously from the invocation of the request. Any errors during the query or file creation results in a JMS notification. The 5620 SAM-O returns the following notifications for problems related the <findToFile> requests:

- JMS notification for an error that occur during the query or file creation
- error message returned to the issuer of the request if the number of <findToFile> requests exceeds the limit configured in the nms-server.xml file for the pool or queue values

The <synchronous> element in the <findToFile> request specifies the execution type for the method. When synchronous requests are used, the request blocks further requests until the <findToFile> request is complete. You can set the <synchronous> element to false to run asynchronous requests.

To avoid blocking other HTTP(s) concurrent requests, up to the HTTP maximum limit, the maximum number of concurrent <find> or <findToFile> requests is 5. If 5 concurrent <find> or <findToFile> requests is reached, subsequent <find> or <findToFile> requests are rejected with an XMLException or IMException response respectively.

FTP configuration

Files created using the <findToFile> method have the file permissions of the 5620 SAM-O process. OSS clients that FTP to the 5620 SAM server to obtain the files must ensure that they possess read/write permissions for the directory that contains the files.

Note — The <findToFile> method returns the relative path for the request. You must make sure that the root directory of your FTP application aligns with the root directory used by the 5620 SAM server to store results of the <findToFile> method. The xmlFindRoot property in the nms-server.xml file defines the root directory. The default value is C:\5620sam\server\xml_output.

File formats

The file format that contains the data requested by the OSS application when performing a <findToFile> method depends on the result filters that are specified in the request. See section 14.3 for more information about result filters.

You specify the directory to store the XML during the installation of the database. Subdirectories are not supported.

See Appendix C for more information on how to obtain sample XML requests and responses to gather large amounts of data from the 5620 SAM server or database.

Scheduled <findToFile>

The <findToFile> method supports the scheduled export of data using the existing 5620 SAM scheduler framework. All the options that are supported by the scheduler framework are available for scheduling a <findToFile> task. Table 14-2 describes how to schedule a data export.

Table 14-2 Creating a scheduled data export

Task	Example

1. Create a schedule or choose an existing schedule.	`<generic.GenericObject.configureChildInstance xmlns="xmlapi_1.0">`
	`<deployer>immediate</deployer>`
	`<distinguishedName>scheduleManager</distinguishedName>`
	`<childConfigInfo>`
	`<schedule.SamSchedule>`
	`<actionMask>`
	`<bit>create</bit>`
	`</actionMask>`
	`<id>10</id>`
	`<displayedName>samSchedule-10</displayedName>`
	`<description>samSchedule-10</description>`
	`<frequency>minute</frequency>`
	`<onGoing>true</onGoing>`
	`<runEveryMinute>5</runEveryMinute>`
	`</schedule.SamSchedule>`
	`</childConfigInfo>`
	`</generic.GenericObject.configureChildInstance>`

2. Create a findToFileTask or reference an existing task.	`<generic.GenericObject.configureChildInstance xmlns="xmlapi_1.0">`
	`<deployer>immediate</deployer>`
	`<distinguishedName>scheduleManager</distinguishedName>`
	`<childConfigInfo>`
	`<schedule.OssTask>`
	`<actionMask>`
	`<bit>create</bit>`
	`</actionMask>`
	`<id>10</id>`
	`<displayedName>FindToFile</displayedName>`
	`<description>FindToFile</description>`
	`<commandName>FindToFile</commandName>`
	`<command>`
	`<findToFile xmlns="xmlapi_1.0">`
	`<fullClassName>netw.NetworkElement</fullClassName>`
	`<fileName>netw.xml</fileName>`
	`<resultFilter>`
	`<children>`
	`<resultFilter class="equipment.Equipment">`
	`</resultFilter>`
	`</children>`
	`</resultFilter>`
	`</findToFile>`
	`</command>`
	`</schedule.OssTask>`
	`</childConfigInfo>`

(1 of 2)

Task	Example

3. Create a samScheduledTask, which points to a schedule, and a findToFile task.	`<generic.GenericObject.configureChildInstance xmlns="xmlapi_1.0">`
	`<deployer>immediate</deployer>`
	`<distinguishedName>scheduleManager</distinguishedName>`
	`<childConfigInfo>`
	`<schedule.SamScheduledTask>`
	`<actionMask>`
	`<bit>create</bit>`
	`</actionMask>`
	`<id>15</id>`
	`<administrativeState>2</administrativeState>`
	`<displayedName>FindToFileTask-15</displayedName>`
	`<description>FindToFileTask15</description>`
	`<scheduledObjectPointer>scheduleManager:OssTask-111</scheduledObjectPointer>`
	`<schedulePointer>scheduleManager:samSchedule-10</schedulePointer>`
	`</schedule.SamScheduledTask>`
	`</childConfigInfo>`
	`</generic.GenericObject.configureChildInstance>`

4. Turn up, shut down, start or stop the samScheduledTask.	—

(2 of 2)

Procedure 14-1 To configure permission for the <findToFile> directory

1 Create file storage directories for each client application accessing the 5620 SAM-O. For example:

```
.../5620SAM-O/clientapp1

.../5620SAM-O/clientapp2

.../5620SAM-O/clientapp3
```

2 Configure read/write permissions for the FTP directory associated with each client application.

3 Configure an FTP account for the 5620 SAM-O client accessing the directory.

4 Configure the FTP account home directories to point to the directory associated with the <findToFile> method. For example:

```
.../5620SAM-O
```

OSS clients can save files using the <findToFile> method by specifying the appropriate client directory in the path. For example:

```
clientapp1/<filename>
```

14.2 5620 SAM object hierarchy

Objects in the 5620 SAM object model are organized in an object hierarchy. The specific parent-child relationships in the hierarchy are captured by the Children-Set property for a given parent object. See the 5620 SAM-O XML Reference for more information about the Children-Set property.

Physical object hierarchy

The <objectFullName> tag displays the hierarchy of how physical objects are organized and used when constructing XML requests. The 5620 SAM physical objects include:

- sites
- cards
- ports
- physical links

The 5620 SAM-O uses the following format for the <objectFullName> tag. This is a sample format associated with a port:

```
network:<siteID>:shelf-<ID>:cardSlot-<ID>:card:daughterCardSlot-<ID>
:daughterCard:port-<ID>
```

The format is:

- The network is the top level.
- The site ID of the managed router.
- Each router has one shelf, which can have multiple slots.
- Up to 12 card slots, where cards are inserted, is indicated by the slot ID.
- The card, which is also called an IOM, contains one or two daughter card (MDA) slots.
- The daughter cards plug into daughter card slots and contain the physical ports.
- The port ID, if applicable.

See Appendix C for more information on how to obtain samples to generate inventories of equipment using the generic package.

Table 14-3 lists the containment hierarchy tree that is associated with equipment types.

Table 14-3 Equipment containment hierarchy

Shelf/slot/card

```
(1) netw.NetworkElement
        (2) equipment.Shelf
                (3) aps.ApsGroup
                        (4) aps.ApsChannel
                        (4) sonetequipment.SonetPortMonitorSpecifics
                        (4) sonetequipment.SonetPortOverheadSpecifics
                        (4) sonetequipment.SonetPortSpecifics
                        (4) sonetequipment.Sts3Channel
                        (4) sonetequipment.Sts12Channel
                        (4) sonetequipment.Sts48Channel
                        (4) sonetequipment.Sts192Channel
                (3) equipment.CCM
```

(1 of 2)

```
(3) equipment.FanTray
(3) equipment.LEDPane
(3) equipment.PowerSupplyTray
(3) equipment.SystemStatsHolder
(3) equipment.CardSlot
        (4) equipment.BaseCard
                (5) sw.IOCardSoftware
                (5) equipment.HwEnvironment
                (5) equipment.PhysicalPort
                (5) equipment.DaughterCardSlot
                        (6) slope.QosPool
                        (6) equipment.MCMCardSlot
                        (6) equipment.MCMCard
                        (6) equipment.HwEnvironment
                        (6) sw.IOCardSoftware
                        (6) equipment.DaughterCard
                                (7) sw.IOCardSoftware
                                (7) bundle.Interface
                                        (8) bundle.PortTermination
                                        (8) netw.PhysicalInterfaceCtp
                                (7) circem.CeDaughterCardSpecifics
                                (7) circem.CesChannelGroup
                                        (8) circem.StructuredInterface
                                (7) equipment.HwEnvironment
                                (7) equipment.PhysicalPort
        (4) equipment.ProcessorCard
                (5) equipment.ControlProcessor
                        (6) equipment.HwEnvironment
                        (6) sw.ControlProcessorSoftware
                (5) equipment.FlashMemory
                (5) equipment.ManagementPort
                        (6) equipment.MediaAdaptor
                        (6) ethernetequipment.EthernetPortSpecifics
                (5) equipment.SwitchFabricProcessor
```

Port

```
(1) equipment.PhysicalPort
        (2) circem.UnstructuredInterface
        (2) equipment.MediaAdaptor
        (2) ethernetequipment.EthernetPortSpecifics
        (2) netw.PhysicalInterfaceCtp
        (2) pae802_1x.PaePortAuthenticatorSpecifics
        (2) pae802_1x.PaePortSpecifics
        (2) slope.QosPool
        (2) sonetequipment.SonetPortMonitorSpecifics
        (2) sonetequipment.SonetPortOverheadSpecifics
        (2) sonetequipment.SonetPortSpecifics
        (2) sonetequipment.Sts1Channel
        (2) sonetequipment.Sts3Channel
        (2) sonetequipment.Sts12Channel
        (2) sonetequipment.Sts48Channel
        (2) sonetequipment.Sts192Channel
        (2) tdmequipment.DS1E1Channel
        (2) tdmequipment.DS1E1PortSpecifics
        (2) tdmequipment.DS3E3Channel
        (2) tdmequipment.DS3E3PortSpecifics
        (2) vpls.VlanPort
                (3) vpls.VlanPortL2AccessBinding
```

(2 of 2)

Logical object hierarchy

Table 14-4 lists the hierarchy of logical objects, such as services, and the children objects that belong to them.

Note — See the DVD-ROM for the 5620 SAM application for the containment and inheritance hierarchy that is associated with logical objects. You can access the information using the index.html file in the user documentation directory. See section 7.1 for more information.

Table 14-4 Logical object hierarchy

Services:
Composite services
(1) service.CompositeServiceManager (comp-svc-mgr)
    (2) service.CompositeService
    (2) service.CrossConnect
    (2) service.ScpConnector
    (2) service.SpokeConnector

IES
(1) service.ServiceManager (svc-mgr)
    (2) ies.Ies
        (3) ies.L3AccessInterface
            (4) atm.SapAtmConfiguration
            (4) rtr.DhcpRelayConfigurationy
            (4) rtr.StaticArp
            (4) rtr.VirtualInterfaceArpConfiguration
            (4) rtr.VirtualInterfaceIcmpConfiguration
            (4) rtr.VirtualInterfaceNtpConfiguration
            (4) rtr.DhcpRelayV6Configuration
            (4) rtr.DhcpRelayV6PrefixDelegation
            (4) rtr.DirectInterfaceCtp
            (4) rtr.NeighborDiscovery
            (4) rtr.ProxyNeighborDiscovery
            (4) rtr.RouterAdvertisement
            (4) rtr.UnnumberedInterfaceCtp
            (4) rtr.VirtualInterfaceIcmp6Configuration
            (4) rtr.VirtualRouterIpAddress
            (4) service.EgressAccessPolicyQueueOverride
            (4) service.EgressSchedulerPolicyEntryOverride
            (4) service.IngressAccessPolicyQueueOverride
            (4) service.IngressSchedulerPolicyEntryOverride
            (4) ressubscr.SapSubMgmtCfg
            (4) vrrp.Instance (See Protocol section)
        (3) ies.SubscriberInterface
            (4) rtr.SubItfDhcpRelayCfg
            (4) ies.GroupInterface
                (5) rtr.GrpItfDhcpRelayCfg
                (5) rtr.VirtualInterfaceArpConfiguration
                (5) rtr.VirtualInterfaceIcmpConfiguration
                (5) ies.ServiceAccessPoint
                (6) antispoof.AntiSpoofingStaticHosts
                (6) ressubscr.SapSubMgmtCfg
                (5) rtr.DirectInterfaceCtp
            (4) rtr.DirectInterfaceCtp
            (4) rtr.VirtualRouterIpAddress
        (3) svt.SpokeSdpBinding (See Topology section)

VLL — L2 Access Interface
(1) vll.L2AccessInterface
    (2) atm.SapAtmConfiguration
    (2)service.EgressAccessPolicyQueueOverride
    (2) service.EgressSchedulerPolicyEntryOverride
    (2) service.IngressAccessPolicyQueueOverride
    (2) service.IngressSchedulerPolicyEntryOverride

VLL — Apipe
(1) service.ServiceManager (svc-mgr)
    (2) apipe.Apipe
        (3) apipe.Site
            (4) svt.SpokeSdpBinding (See Topology section)
            (4) vll.Endpoint
            (4) vll.L2AccessInterface
                (5) atm.SapAtmConfiguration

VLL — Epipe
(1) service.ServiceManager (svc-mgr)
    (2) epipe.Epipe
        (3) epipe.Site
            (4) svt.SpokeSdpBinding (See Topology section)
            (4) vll.Endpoint
            (4) vll.L2AccessInterface

VLL — Fpipe
(1) service.ServiceManager (svc-mgr)
    (2) fpipe.Fpipe

(1 of 8)

```
                (3) fpipe.Site
                        (4) svt.SpokeSdpBinding (See Topology section)
                        (4) vll.Endpoint
                        (4) vll.L2AccessInterface
```

VLL — Ipipe

```
(1) service.ServiceManager (svc-mgr)
        (2) ipipe.Ipipe
                (3) ipipe.Site
                        (4) svt.SpokeSdpBinding (See Topology section)
                        (4) vll.Endpoint
                        (4) ipipe.L2AccessInterface
                        (Children sets inherited from vll.L2AccessInterface)
```

VLAN

```
(1) service.ServiceManager (svc-mgr)
        (2) vlan.vlan
                (3) vlan.MvrConfiguration
                (3) vlan.site
                        (4) vlan.L2AccessInterface
                                (5) layer2.MvrInterfaceConfigure
                                (5) layer2.StdVlanInterfaceConfigure
                                (5) layer2.SuperVlanInterfaceConfigure
                                (5) layer2.TlsInterfaceConfigure
                        (4) vlan.SuperVlanSite
                (3) vlan.SuperVlanConfiguration
                (3) vlan.TlsConfiguration
```

VPLS

```
(1) service.ServiceManager (svc-mgr)
        (2) vpls.Vpls
                (3) l2fwd.ServiceMacProtection
                (3) vpls.Site
                        (4) l3fwd.ServiceSite
                        (4) l3fwd.ServiceSiteExportPolicy
                        (4) l3fwd.ServiceSiteImportPolicy
                        (4) rtr.SourceAddress
                        (4) rtr.StaticRoute
                        (4) svt.Cloud
                        (4) svt.SpokeSdpBinding (See Topology section)
                        (4) vpls.L2AccessInterface
                                (5) l2fwd.AccessInterfaceStp
```

MVPLS

```
(1) service.ServiceManager (svc-mgr)
        (2) mvpls.Mvpls
                (3) mvpls.Site
                        (4) l2fwd.SiteMstp
                                (5) l2fwd.MstInstance
                                                (6) l2fwd.MstiVlanRange
                        (4) mvpls.L2AccessInterface
                        (4) rtr.SourceAddress
                        (4) rtr.StaticRoute
                        (4) svt.Cloud
                        (4) svt.SpokeSdpBinding (See Topology section)
                        (4) vpls.L2AccessInterface
                                (5) l2fwd.RedundantVlanRange
                        (4) shg.Site
```

VPRN

```
(1) service.ServiceManager (svc-mgr)
        (2) vprn.Vprn
                (3) vprn.Site
                        (4) l3fwd.ServiceSite
                        (4) l3fwd.ServiceSiteExportPolicy
                        (4) l3fwd.ServiceSiteImportPolicy
                        (4) rtr.SourceAddress
                        (4) rtr.StaticRoute
                        (4) svt.Cloud
                        (4) svt.SpokeSdpBinding (See Topology section)
                        (4) vprn.L3AccessInterface
                                (5) atm.SapAtmConfiguration
                                (5) antispoof.L3AntiSpoofing
                                                (6)Antispoof.AntiSpoofingStaticHosts
                                (5) rtr.DhcpRelayConfiguration
                                (5) rtr.StaticArp
```

(2 of 8)

(5) rtr.VirtualInterfaceArpConfiguration
(5) rtr.VirtualInterfaceIcmpConfiguration
(5) rtr.VirtualInterfaceNtpConfiguration
(5) rtr.DirectInterfaceCtp
(5) rtr.UnnumberedInterfaceCtp
(5) rtr.VirtualRouterIpAddress
(5) vrrp.Instance (See Protocol section)
(5) service.EgressAccessPolicyQueueOverride
(5) service.EgressSchedulerPolicyEntryOverride
(5) service.IngressAccessPolicyQueueOverride
(5) service.IngressSchedulerPolicyEntryOverride
(4) vprn.RoutingInstanceSite
(5) bgp.Site (See Protocol section)
(5) igmp.Site (See Protocol section)
(5) ospf.Site (See Protocol section)
(5) pim.Site (See Protocol section)
(5) rip.Site (See Protocol section)
(4) vprn.SubscriberInterface
(5) rtr.DirectInterfaceCtp
(5) rtr.VirtualRouterIpAddress
(5) vprn.GroupInterface
(6) rtr.GrpItfDhcpRelayCfg
(6) rtr.VirtualInterfaceArpConfiguration
(6) rtr.VirtualInterfaceIcmpConfiguration
(6) rtr.DirectInterfaceCtp
(6) vprn.ServiceAccessPoint
(7)ressubscr.SapSubMgmtCfg

Cross Connect Aggregation Group
(1) ccag.CrossConnectAggregationGroup
(2) ccag.CcagMdaMember
(2) ccag.CcagPath
(3) ccag.CcagPathCc<NetSap, SapNet, SapSap>
(4) netw.PhysicalInterfaceCtp

Mirror
(1) service.ServiceManager (svc-mgr)
(2) mirror.Mirror
(3) mirror.Site
(4) mirror.L2AccessInterface
(4) mirror.RemoteSource
(4) mirror.SourceIngressLabel
(4) mirror.SourceInterface
(4) mirror.SourceIpFilter
(4) mirror.SourceMacFilter
(4) mirror.SourcePort
(4) svt.MirrorSdpBinding

Subscribers:
Subscriber
(1) subscr.Manager (subscriber)
(2) subscr.Subscriber
(3) subscr.Site
(4) svq.AggregationScheduler

Residential Subscribers:
SLA Profile
(1) slaprofile.Policy
(2) slaprofile.EgressAccessPolicyQueueOverride
(2) slaprofile.IngressAccessPolicyQueueOverride

Subscriber Identification Policy
(1) subscrident.Policy
(2) subscrident.SLAProfileEntry
(2) subscrident.SubscrProfileEntry

Subscriber Profile
(1) subscrprofile.Policy
(2) subscrprofile.EgressSchedulerPolicyEntryOverride
(2) subscrprofile.IngressSchedulerPolicyEntryOverride
(2) subscrprofile.SLAProfileEntry

Topology:
Network – Topology Discovery
(1) netw.Topology (topology)
(2) netw.NodeDiscoveryControl

(3 of 8)

(2) netw.TopologyDiscoveryRule
(3) netw.DiscoveredNode
(3) netw.TopologyDiscoveryRuleElement

Generic – Generic NE Profile
(1)genericne.GenericNeProfile

Virtual Router
(1) netw.NetworkElement
(2) rtr.VirtualRouter
(3) bgp.Confederation
(3) bgp.Site (See Protocol section)
(3) igmp.Site (See Protocol section)
(3) isis.Site (See Protocol section)
(3) l3fwd.DefaultRoutingSite
(3) ldp.Site (See Protocol section)
(3) mpls.Site (See Protocol section)
(3) msdp.Site (See Protocol section)
(3) ospf.Site (See Protocol section)
(3) pim.Site (See Protocol section)
(3) rip.Site (See Protocol section)
(3) rp.PolicyManager
(3) rsvp.Site (See Protocol section)
(3) rtr.Aggregation
(3) rtr.NetworkInterface
(3) rtr.ProtocolSite
(3) rtr.ServiceAddressRange
(3) rtr.SourceAddress
(3) rtr.StaticRoute

Router – Network Interface
(1) rtr.NetworkInterface
(2) rtr.DhcpRelayConfiguration
(2) rtr.DirectInterfaceCtp
(2) rtr.NeighborDiscovery
(2) rtr.ProxyNeighborDiscovery
(2) rtr.RouterAdvertisement
(3) rtr.RouterAdvertisementPrefix
(2) rtr.StaticArp
(2) rtr.UnnumberedInterfaceCtp
(2) rtr.VirtualInterfaceArpConfiguration
(2) rtr.VirtualInterfaceIcmp6Configuration
(2) rtr.VirtualInterfaceIcmpConfiguration
(2) rtr.VirtualInterfaceNtpConfiguration
(2) rtr.VirtualRouterIpAddress
(2) vrrp.Instance (See Protocol section)

ATM – Network Interface
(1) atm.Interface
(2) atm.IfConnection
(2) atm.IlmiLink
(2) atm.PvcConnection
(2) atm.VPConnection
(2) atm.VTConnection

OSPF Area
(1) ospf.Ospf
(2) ospf.Area
(3) opsf.AreaMemberSite

Service Tunnel
(1) svt.Manager
(2) svt.Tunnel
(3) svt.TransportConnection
(3) svt.TunnelInstanceSpecific

Service Tunnel Spoke SDP Binding
(1) svt.SpokeSdpBinding
(2) svt.SdpBindingApipe
(2) svt.SdpBindingIpipe
(2) svt.SdpBindingIgmpSnpgCfg
(3) svt.SdpBindingIgmpSnpgStaticMcastGrp
(2) l2fwd.CircuitFib
(2) l2fwd.CircuitStp
(2) l2fwd.RedundantCircuitConfiguration

(4 of 8)

(2) vpls.CircuitDhcpRelayCfg
(2) vpls.SpokeSdpBindingCfg

MPLS Dynamic LSP Path
(1) mpls.DynamicLsp
(2) mpls.LspExtension
(2) mpls.LspPath

MPLS Static LSP Path
(1) mpls.StaticLsp
(2) mpls.StaticHop

MPLS Provisioned Path
(1) mpls.ProvisionedPath
(2) mpls.PathPostOperator
(2) mpls.ProvisionedHop

MPLS Actual Path
(1) mpls.ActualPath
(2) mpls.ActualHop

MPLS CSPF Path
(1) mpls.CSPFPath
(2) mpls.CSPFHop

Protocols:
BGP
(1) bgp.Site (router)
(2) bgp.Md5Key
(2) bgp.PeerGroup
(3) bgp.GroupExportPolicy
(3) bgp.GroupImportPolicy
(3) bgp.Md5Key
(3) bgp.Peer
(4) bgp.PeerExportPolicy
(4) bgp.Md5Key
(4) bgp.PeerExportPolicy
(2) bgp.SiteExportPolicy
(2) bgp.SiteImportPolicy

IGMP
(1) igmp.Site
(2) igmp.Interface
(3) igmp.IfGroup
(4) igmp.IfGrpSrc
(3) igmp.McastCacLagPortDown
(3) igmp.McastCacLevel
(3) igmp.StaticGrpSrc
(2) igmp.SSMTranslate
(2) igmp.SourceGroup

OSPF
(1) ospf.Site
(2) ospf.AreaSite
(3) ospf.AdvertisedRange
(3) ospf.Interface
(4) ospf.Md5Key
(4) ospf.Neighbor
(3) ospf.LsaCounter
(3) ospf.VirtualLink
(4) ospf.Md5Key
(4) ospf.VirtualNeighbor
(2) ospf.ExportPolicy

ISIS
(1) isis.Site
(2) isis.AreaSite
(2) isis.AuthenticationKey
(2) isis.Interface
(3) isis.Adjacency
(4) isis.AdjacencyArea
(3) isis.AuthenticationKey
(3) isis.InterfaceLevelConfig
(3) isis.InterfaceLevelOneConfig
(4) isis.AuthenticationKey
(3) isis.InterfaceLevelTwoConfig

(5 of 8)

```
                    (4) isis.AuthenticationKey
        (2) isis.RouteSummarization
        (2) isis.SiteExportPolicy
        (2) isis.SiteLevelConfig
        (2) isis.SiteLevelOneConfig
                (3) isis.AuthenticationKey
        (2) isis.SiteLevelTwoConfig
                (3) isis.AuthenticationKey
        (2) isis.SpfLog
```

RIP

```
(1) rip.Site
        (2) rip.AuthenticationKey
        (2) rip.Group
                (3) rip.AuthenticationKey
                (3) rip.GroupExportPolicy
                (3) rip.GroupImportPolicy
                (3) rip.Interface
                        (4) rip.AuthenticationKey
                        (4) rip.InterfaceExportPolicy
                        (4) rip.InterfaceImportPolicy
        (2) rip.SiteExportPolicy
        (2) rip.SiteImportPolicy
```

LDP

```
(1) ldp.Site
        (2) ldp.Interface
        (2) ldp.Peer
                (3) ldp.Md5Key
        (2) ldp.Session
        (2) ldp.SiteExportPolicy
        (2) ldp.SiteImportPolicy
        (2) ldp.StaticFec
        (2) ldp.TargetedPeer
        (2) ldp.TargetedSessionExportPolicy
        (2) ldp.TargetedSessionImportPolicy
```

RSVP

```
(1) rsvp.Site
        (2) rsvp.Interface
                (3) rsvp.Neighbor
        (2) rsvp.Session
```

MPLS

```
(1) mpls.Site
        (2) mpls.Crossconnect
        (2) mpls.Interface
                (3) mpls.Crossconnect
                (3) mpls.InSegment
                (3) mpls.OutSegment
                (3) mpls.StaticLabelMap
        (2) mpls.OutSegment
```

MSDP

```
(1) msdp.Site
        (2) msdp.Peer
                (3) msdp.PeerExportPolicy
                (3) msdp.PeerImportPolicy
                (3) msdp.PeerMd5Key
        (2) msdp.PeerGroup
                (3) msdp.GroupPeer
                        (4) msdp.GroupPeerMd5Ke
                        (4) msdp.GrpPeerExportPolicy
                        (4) msdp.GrpPeerImportPolicy
                (3) msdp.GrpExportPolicy
                (3) msdp.GrpImportPolicy
        (2) msdp.SiteExportPolicy
        (2) msdp.SiteImportPolicy
        (2) msdp.Source
        (2) msdp.SourceActive
```

PIM

```
(1)  pim.Site
        (2) pim.AnyCastRP
                (3) pim.AnyCastPeer
        (2) pim.DataMdtThreshold
```

(6 of 8)

(2) pim.DataMtInterface
(2) pim.ExportBootstrapPolicy
(2) pim.GroupPrefix
(2) pim.Groups
(2) pim.ImportBootstrapPolicy
(2) pim.ImportJoinPrunePolicy
(2) pim.ImportRegisterPolicy
(2) pim.Interface
(3) pim.McastCacLagPortDown
(3) pim.McastCacLevel
(3) pim.Neighbor
(2) pim.RPSet
(2) pim.SSMGroup
(2) pim.SptSwitchOvrThreshold
(2) pim.StaticRP
(3) pim.StaticGroupToRP

VRRP Instance
(1) rtr.NetworkInterface
(2) vrrp.Instance
(3) vrrp.AuthenticationKey
(3) vrrp.BackupAddress

VRRP Policy
(1) vrrp.Policy
(2) vrrp.HostUnreachableEvent
(2) vrrp.LagPortDownEvent
(3) vrrp.NumberDown
(2) vrrp.PortDownEvent
(2) vrrp.RouteUnknown
(3) vrrp.NextHop

Policy:
Poller
(1) snmp.PollerManager (pollerManager)
(2) mediation.ManagementPingPolicy
(3) mediation.ManagementPingDestination
(2) security.MediationPolicy
(2) snmp.StatsPollerPolicy
(3) snmp.ProductPolicy
(2) snmp.ProductPolicy
(3) snmp.VersionPolicy
(4) snmp.PollerPolicy

Accounting
(1) policy.Manager
(2) accounting.Policy

File
(1) policy.Manager
(2) file.Policy

Access Egress
(1) aengr.Policy
(2) aengr.ForwardingClass
(2) aengr.Queue

Access ingress
(1) aingr.Policy
(2) aingr.Dot1p
(2) aingr.Dscp
(2) aingr.ForwardingClass
(3) aingr.ForwardingSubClass
(2) aingr.IpMatch
(2) aingr.Ipv6Match
(2) aingr.MacMatch
(2) aingr.Precedence
(2) aingr.Queue

ACL Filter – IP Filter
(1) aclfilter.IPFilter
(2) aclfilter.IpFilterEntry

ACL Filter – MAC Filter
(1) aclfilter.MACFilter
(2) aclfilter.MacFilterEntry

(7 of 8)

Aggregation Scheduler
(1) svq.AggregationScheduler
    (2) svq.EgressAggregationSchedulerOverride
    (2) svq.IngressAggregationSchedulerOverride

Network Policy
(1) niegr.Policy
    (2) niegr.Dot1p
    (2) niegr.Dscp
    (2) niegr.ForwardingClass
    (2) niegr.LspExp

Network Queue
(1) nqueue.Policy
    (2) nqueue.Entry
    (2) nqueue.ForwardingClass

RADIUS Server
(1) policy.Manager
    (2) pae802_1x.RadiusServerPolicy
        (3) pae802_1x.RadiusServerItem

Router
(1) rp.PolicyManager
    (2) rp.ASPath
    (2) rp.Community
        (3) rp.CommunityMember
    (2) rp.Damping
    (2) rp.PolicyStatement
        (3) rp.DefaultAction
        (3) rp.PolicyStatementEntry
            (4) rp.Action
            (4) rp.FromCriteria
            (4) rp.ToCriteria
    (2) rp.PrefixList
        (3) rp.PrefixListMember
    (2) rp.TransactionObject

Virtual Scheduler
(1) vs.Policy
    (2) vs.Entry

Slope
(1) slope.Policy
    (2) slope.HighSlope
    (2) slope.LowSlope

Other:
Script Management
(1) script.ScriptManager (script-manager)
    (2) script.Script
        (3) script.TargetScript
            (4) script.TargetParameter
            (4) script.TargetScriptResult
        (3) script.Version
            (4) script.Parameter
            (4) script.ParameterTab
                (5) script.ParameterGroup
                (6) script.FilterParameter
                (6) script.FormatParameter
                (6) script.ListParameter
                (7) script.ListItemParameter
                (6) script.Parameter
    (2) script.VelocityTemplate
    (2) script.XmlApiScript
        (3) script.TargetScript
            (4) script.TargetParameter
            (4) script.TargetScriptResult
        (3) script.XmlApiVersion
            (4) script.ParameterTab
                (5) script.ParameterGroup
                (6) script.FilterParameter
                (6) script.FormatParameter
                (6) script.ListParameter
                (7) script.ListItemParameter

(8 of 8)

14.3 Filters

You can define a filter for any property within an object. Use the filters to limit or refine the:

- information returned to the OSS application after a request is made to the server
- configuration requests sent to the server

The filter syntax is part of an XML OSS request. In code 14-6, the filter specifies that the server executes the method when the monitoredObjectSiteId element has an IP address of 1.11.111.111. The server executes the request and returns information from a NE with an IP address of 1.11.111.111.

Code 14-6: Sample filter syntax

```
<filter>
<equal name="monitoredObjectSiteId" value="1.11.111.111" />
 </filter>
```

Filter construction requires that you know the class to which you want to apply the filter. The class can be defined by the class attribute in the top-level <filter> element. Code 14-7 shows an example of using the class attribute to define the filter class.

Note — The use of filters when retrieving information about deployers is handled slightly differently compared to using <find> or <findToFile> methods described above. See section 16.3 for more information about deployer methods.

Code 14-7: Sample filter using class attribute

```
<filter>
 <equal class="netw.NetworkElement".../>
</filter>
```

Children filters

The <find> and <findToFile> methods support the use of nested children filters for an object. Code 14-8 shows the use of a filter that returns VPLS sites with the following criteria:

- displayed name that starts with *MySites*
- contain children with a displayed name that starts with *MyInterfaces*
- contains *ressubscr.SapSubMgmtCfg* with disabled administrative status

The attribute *childClass* is required if the attributes used in the children filter are specific to a child class. You can set the *withChildrenOnly* attribute to true to specify an object is returned only if it has children that match the child filter. The *withChildrenOnly* attribute is optional and is set to false by default.

Code 14-8: Children filters

```
<filter class="vpls.Site">
           <wildcard name="displayedName"
                  value="MySites%"/>
           <children>
                   <filter class="vpls.Site" withChildrenOnly="true">
```

```
                        <wildcard name="displayedName"
value="MyInterfaces%"/>
                        <children>
                             <filter class="vpls.L2AccessInterface"
childClass="ressubscr.SapSubMgmtCfg" withChildrenOnly="true">
                                <equal name="adminStatus" value="disabled"/>
                             </filter>
                        </children>
                    </filter>
             </children>
        </filter>
```

Filter element types

The 5620 SAM-O supports the following filter element types:

- composite
- leaf
- result

Composite filter

Composite filter elements have associated child elements. Table 14-5 describes the composite filter element types.

Table 14-5 Composite filter element types

Filter element	Description	Example
and	AND the results of two containing filter elements	X and Y
or	OR the results of two containing filter elements	X or Y
not	NOT the results of one containing filter elements	NOT X

Code 14-9 shows a sample filter to yield results that are the same as the given value. You must specify the bit mask value using a decimal integer format.

Code 14-9: Sample filter using equal and bitmask

```
<filter>
   <and>
      <equal name="siteId" value="10.1.1.52"/>
      <and>
         <anyBit name="family" value="64"/>
         <not>
            <equal name="localAS" value="20"/>
          </not>
       </and>
   </and>
</filter>
```

Leaf filter

Leaf filter elements do not have associated child elements. Table 14-6 describes the leaf filter element types.

Table 14-6 Leaf filter element types

Filter element	Description	Example
equal	Return true if the value is the same as the given value.	<equal name=”localAS” value=”20”/>
notEqual	Return true if the value is not the same as the given value.	<notEqual name="localAS" value="20"/>
greater	Return true if the numeric value is greater than the given value.	<greater name="localAS" value="20"/>
greaterOrEqual	Return true if the numeric value is greater than or equal to the given value.	<greaterOREqual name="localAS" value="20"/>
less	Return true if the value is less than the given value. This only applies to numeric values.	<less name="localAS" value="20"/>
lessOrEqual	Return true if the value is less than or equal to the given value. This only applies to numeric values.	<lessOREqual name="localAS" value="20"/>
anyBit	Return true if the value bit ANDed with the given value does not return 0. The filter is true if any of the “1” bits in the specified value are enabled in the property value. This only applies to non-negative numeric types and bitmask types.	<anyBit name=”parameter” value=”3”/>
allBits	Return true if the value bit ANDed with the given value returns the given value. The filter is true if all of the “1” bits in the specified value are enabled in the property value. This only applies to non-negative numeric types and bitmask types.	<allBits name=”parameter” value=”5”/>
wildcard	Perform equality using wildcards. The character % signifies 0 or more characters of any type, and the character _ signifies one character of any type.	<wildcard name=”SiteId value=”1.%.%.%.%”/>
between	Return true if the numeric value is between the first and second given values	<between name="preference" first="150" second="200" />

Code 14-10 shows an example of the numeric value that is greater than or equal to the given value. You can specify the value of an enumeration using a numeric or string format.

Code 14-10: Sample filter example - usage of greater, between, and equal

```
<filter>
    <and>
       <greater name="localAS" value="0"/>
       <and>
          <equal name="domain" value="network"/>
          <between name="preference" first="150" second="200" />
       </and>
    </and>
</filter>
```

Composite and leaf filters can be used together for not only the top level object, but also for children. For children filters, you must specify the class of the child object being filtered. For example, <filter class="service.Site">.

Code 14-11 shows an example of composite and leaf filters used together for a top level vprn.Site object and children.

Code 14-11: Sample filter example - composite and leaf filters used together for a top level object and children

```
<find xmlns="xmlapi_1.0">
 <fullClassName>service.Service</fullClassName>
  <filter>
     <wildcard  name="displayedName"
               value="%Epipe%"/>
     <children>
       <filter class="service.Service">
         <wildcard  name="displayedName"
                    value="%epipe"/>
         <children>
           <filter class="service.Site">
            <wildcard  name="description"
                       value="%Epipe%"/>
           </filter>
         </children>
       </filter>
     </children>
  </filter>
   <resultFilter>
     …
  </resultFilter>
</find>
```

Result filter

Table 14-7 describes the result filter element types.

Table 14-7 Result filter element types

Children values	Description	Example
<resultFilter>	You can use the <resultFilter> element to define the attributes and children returned for 5620 SAM-O queries. The 5620 SAM-O returns all attributes and children if you do not specify the types in the <resultFilter> element. A nested <resultFilter> element can contain the class attribute that specifies the following filtering types: • Package-qualified class name of the returned child; for example, <resultFilter class="*equipment.Port*"> • Generic base class name for all managed objects; for example, <resultFilter [class="{*child class name*|*ManagedObject*}"]> A result filter applies only to the returned objects. Some objects have their children suppressed by default; for example, <childrenSummaryEnabled>.	—

(1 of 2)

Children values	Description	Example
<attribute>	Attributes on the managed objects for which you are searching; for example, <attribute>*objectFullName*</attribute>	—
<children>	Signifies a list of children and their associated attributes defined by the sub-result filters. An empty tag, <children/>, signifies the return of no children past this level. No <children> tag returns all children. The recursive attribute allows you to filter the children using the same rules as the parent; for example, <children recursive="yes"/>. You do not need to use nested <resultFilter> elements if you enable the recursive attribute.	—

(2 of 2)

Code 14-12 shows an example of a result filter with contained children. There is no limit to the depth of the contained result filters and children tags.

Code 14-12: Result filter with contained children

```
<resultFilter>
  <attribute>attribute</attribute>
  <atribute>status</attribute>
.
.
.
  <children [recursive="{yes|no}"]>
     <resultFilter [class="{child class name|ManagedObject}"]>
        <attribute>attribute</attribute>
     </resultFilter>
.
.
.
  </children>
</resultFilter>
```

Code 14-13 shows an example configuration of a result filter that returns the administrative status and the port speed for the network equipment.

Code 14-13: Result filter for equipment

```
 <SOAP:Body>
        <find xmlns="xmlapi_1.0">
    <fullClassName>equipment.PhysicalPort</fullClassName>
    <filter>
        <equal name="siteName" value="sim202_93"/>
    </filter>
    <resultFilter>
        <attribute>administrativeState</attribute>
                <attribute>speed</attribute>
                <children/>
            </resultFilter>
        </find>
    </SOAP:Body>
```

Code 14-14 shows an example configuration of a result filter in a request that uses the generic.GenericObject.configureInstanceWithResult method. See section 16.2 for more information on the generic methods that are applicable to all objects.

Code 14-14: Result filter in a request that uses generic methods

```
 <SOAP:Body>
        <generic.GenericObject.configureInstanceWithResult
xmlns="xmlapi_1.0">
            <deployer>immediate</deployer>
```

```
            <!-- Network Interface -->
            <distinguishedName>network:10.1.1.223:router-1:ip-interface-5</d
istinguishedName>
            <includeChildren>false</includeChildren>
            <configInfo>
                <rtr.NetworkInterface>
                    <actionMask><bit>modify</bit></actionMask>
                    <!-- vRtrIfUp/vRtrIfDown -->
                    <administrativeState>vRtrIfUp</administrativeState>
                </rtr.NetworkInterface>
            </configInfo>
  <resultFilter>
                <attribute>objectFullName</attribute>
<attribute>displayedName</attribute>
                <children recursive="yes" />
            </resultFilter>
          </generic.GenericObject.configureInstanceWithResult>
    </SOAP:Body>
```

15 — Accounting and performance monitoring

15.1 Accounting and performance monitoring overview *15-2*

15.2 Statistic objects *15-2*

15.3 Workflow to retrieve statistics *15-3*

15.4 JMS XML statistics topic *15-4*

15.5 Scheduled and on-demand statistics *15-5*

15.6 Statistics retrieval process *15-6*

15.7 Sample request to retrieve historical statistics from a router *15-17*

15.8 Sample request to collect interface statistics *15-17*

15.1 Accounting and performance monitoring overview

You can use the 5620 SAM, 5620 SAM-O, and service- and equipment-related statistics counters to collect statistics from the managed devices. The 5620 SAM supports the following types of statistics:

- performance statistics for network objects
- accounting statistics for service access points, network ports and other relevant objects

See the *5620 SAM Statistics Management Guide* for more information about managing statistics collection and to view a list of supported MIB counters that can be collected by the 5620 SAM.

15.2 Statistic objects

The 5620 SAM-O uses the following types of objects to track statistics:

- current data record for the most recent performance statistics poll
- log records for historical polls of accounting and performance statistics

Log records are maintained for both scheduled and non-scheduled statistics. The 5620 SAM-O organizes statistic objects on a package basis. For example, the bgp package defines the BGP statistics. You must use the corresponding log record object to retrieve statistics for an object. For example, the BGP peer statistics are tracked by the PeerStatsLogRecord class in the bgp package. The log records contain the following information:

- time captured
- monitored object pointer
- relevant statistics

You can retrieve statistics from 5620 SAM in the following formats:

- log record objects
- XML-formatted file of the log record objects

A policy is associated with each type of log record. You can use the log policy to modify the thresholds and maximum log size that are associated with a log record. For example, you can modify the PeerStatsLogPolicy class to customize the log records collected for the BGP Peer Stats. The default settings generally provide appropriate values for the thresholds and log size.

Note — The 5620 SAM system performance can be adversely affected by increasing the number of historical statistics entries recorded by the 5620 SAM. Network performance degradations include increased time to list log records from the GUI and OSS clients, increased size of Oracle tablespaces, and increased time to backup the database.

Alcatel-Lucent recommends the efficient management of the number of historical statistics recorded by the 5620 SAM to optimize network performance.

accounting package

The accounting package contains methods that you can use to define accounting policies. The accounting policies collect accounting statistics. There can be an active accounting policy for each of the eight access port statistics types and for the five network port statistics types. The 5620 SAM-O uses an accounting and file policy to manage the collection of statistics.

file package

The file package contains methods that you can use to define file policies. File policies define the storage location for statistics that are generated and collected on the managed devices. File policies are also used to define the rollover period of the stored statistics logs.

log package

The log package contains methods that you can use to define how to retrieve targeted statistics.

Note — Alcatel-Lucent recommends that you do not use the log package to retrieve large quantities of data.

root package

The root package contains methods that you can use to retrieve large quantities of statistics. You can retrieve the statistics using the <findToFile> method. See chapter 14 for more information.

15.3 Workflow to retrieve statistics

Use the 5620 SAM-O and the registerLogToFile method or the findToFile method to transfer the statistics log record from the 5620 SAM database to your OSS client application.

1. Specify a filter to limit the data stored to the file.
2. Construct a valid SOAP request.

3 Perform one of the following steps.

   a Use the findToFile method to retrieve specific accounting or performance statistics.

      i Specify the log record name.

      ii Specify an XML export format.

      iii Specify the name of the data storage file.

   b Use the registerLogToFile method to create accounting statistics export files.

      i Specify the accounting statistics class or classes.

      ii Specify the location of the export files.

      iii Specify whether the files are compressed.

      iv Specify the attributes that are to be provided with the exported data records.

4 Send the request.

5 Receive a response or exception to the request.

   See the following samples for more information about configuring the collection of statistics:

   - "Sample request to retrieve historical statistics from a router"
   - "Sample request to collect interface statistics"
   - "Statistics retrieval example using the <registerLogToFile> method"

15.4 JMS XML statistics topic

The 5620 SAM supports the publishing of polled statistics measurements to OSS clients using JMS on the XML statistics topic.

Note — This section describes a deprecated method of publishing polled statistics measurements to OSS clients and is not recommended by Alcatel-Lucent.

If the configuration of the topics specifies statistics on the same topic as other message types, you can filter the client statistics and accounting data using the filter on a subscription. The filter is based on the category of the message, for example, statistics or accounting data. The default setting for the 5620 SAM is to send statistics and accounting messages to all clients.

A client can stop listening to the messages by setting one or more filters on a subscription identifying that the category of message should not include statistics or accounting data. A maximum of ten 5620 SAM-O clients can listen to the messages.

See section 4.2 for more information about subscribing to JMS topics.

JMS XML statistics notifications are generated whenever scheduled statistics of the corresponding statistics policy are modified. You can disable this functionality by setting the Events Enabled field for the statistics policy to false.

15.5 Scheduled and on-demand statistics

You can use the 5620 SAM-O to collect statistical data using the following methods:

- scheduled polls
- on-demand collection

Both scheduled and on-demand performance statistics have a current data record. History is maintained for the scheduled polls using the LogRecord type. No history is maintained for on-demand statistics. The data from the last scheduled poll is also stored using the LogRecord type. For example, equipment.InterfaceAdditionalStats are collected in a 5-minute polling interval. The scheduled polling creates a current data record and a log record.

Note — The current data records do not apply to accounting statistics.

The <objectFullName> of the scheduled current data starts with: logger:scheduled. The on-demand current data <objectFullName> starts with: logger:real-time. A historical log record is also created for the scheduled current data. See "Object full name" in section 9.1 for more information about the unique identifier of the instance of the object.

If you initiate a non-scheduled collection using the generic.GenericObject.triggerCollect method, a new on-demand current data object is brought into the system. The 5620 SAM-O does not modify the scheduled current data object or create a new historical log record. On-demand statistics are propagated to the history as non-scheduled statistics and do not affect any scheduled statistic retrievals.

You can list several instanceNames when you use the generic.GenericObject.triggerCollect method for a non-scheduled collection, but you must specify the current data class for each of the instances that contain the statistics that you want to collect. The current data classes must be listed in currentDataClasses. Code 15-1 shows a sample non-scheduled collection.

Code 15-1: Sample non-scheduled statistics collection

```
<generic.GenericObject.triggerCollect xmlns="xmlapi_1.0">
<instanceNames><string>network:IP_address:shelf-1:cardSlot-1:card:da
ughterCardSlot-1:daughterCard:port-4</string></instanceNames>
<currentDataClasses><string>equipment.InterfaceAdditionalStats</stri
ng></currentDataClasses>
</generic.GenericObject.triggerCollect>
```

For the equivalent behavior of the Collect button on the 5620 SAM GUI, you can specify one instance in the list with the corresponding data class in the second list. For the equivalent behavior of the Collect All button on the 5620 SAM GUI, you can specify all relevant instances in the list with all corresponding data classes in the second list. See the *5620 SAM User Guide* for more information about how to use the Collect and Collect All buttons.

15.6 Statistics retrieval process

You can use the 5620 SAM-O to collect statistical data using the following methods:

- <registerLogToFile> to maintain up-to-date accounting statistics export files
- <findToFile> to retrieve specific accounting and performance statistics

Note — The 5620 SAM allows you to configure queue filters for accounting statistics. See the *5620 SAM Statistics Management Guide* for more information.

Accounting statistics retrieval using registerLogToFile

You can use the 5620 SAM-O and the <registerLogToFile> method to create accounting statistics export files that are based on specific class types. Accounting statistics data is exported to the file after the data is read from the NE. The file is saved to a specified directory on the 5620 SAM server. A LogFileAvailableEvent is generated each time a file is created. See section "LogFileAvailableEvent" in chapter 4 for more information.

You can specify multiple statistics classes in the registration request. The statistics export files are created for each 5620 SAM-O client that registers to receive notifications.

Note 1 — Alcatel-Lucent recommends using the registerLogToFile method to maintain an up-to-date view of accounting statistics. Use the findToFile method for infrequent queries on accounting or performance statistics. The findToFile method is not recommended in large networks with high statistics collection rates. See section "Accounting and performance statistics retrieval using findToFile" for more information about the findToFile method.

Note 2 — The use of the file compression option with registerLogToFile method requires additional CPU resources on the 5620 SAM server or auxiliary statistics collector. The additional load can result in the 5620 SAM not being able to collect the maximum number of accounting records documented in the *5620 SAM Planning Guide*.

Note 3 — The 5620 SAM server verifies the connectivity of the 5620 SAM-O client that is specified in the registerLogtoFile method using the jmsClientId input parameter. The server deregisters the LogToFile request and the 5620 SAM-O client if the JMS session is disconnected.

Alcatel-Lucent recommends that you run the registerLogToFile method every time the 5620 SAM-O client establishes a JMS session.

Table 15-1 describes the input parameters of the <registerLogToFile> method.

Table 15-1 <registerLogToFile> method input parameters

Input parameter	Description	Values
fullClassName	Package qualified class name in dot separated format.	A comma-separated list of accounting statistics classes
dirName	The relative path to the subdirectory where the files are to be saved. The path is relative to the OSS XML export directory/accountingStats. The use of separate directories for different applications that export statistics is recommended.	—
compress	(Optional) Specifies whether export files should be compressed when they are created. The use of registerLogToFile with the compress option set to true impacts the statistics collection capability of the 5620 SAM.	The options are: • true Saves files with gzip compression and a filename extension of gz. • false (default) Files are not compressed when they are created.

(1 of 2)

Input parameter	Description	Values
jmsClientId	The JMS client ID that is notified when a new export file is created for the specified parameters. A notification message is sent to each client that registers for statistics every time an export file is created that is related to that client.	See chapter 4 for more information about JMS client ID formats.
resultFilter	(Optional) Specifies the attributes that are to be included in the exported data records. You can use the resultFilter parameter to reduce the size of the export file.	<attribute>*attribute_name*</attribute>

(2 of 2)

The statistics export files can be created on the 5620 SAM server or auxiliary servers. When statistics are collected on one or more auxiliary servers, export files are created on the auxiliary server or servers and not on the primary 5620 SAM server. Export file names are unique and are created under the directory that you specify in the registration request. For example:

/opt/5620sam/server/nms/export/accountingStats/*client_directory*/138.120.111.112_1184686976235.xml

The 5620 SAM-O client can retrieve the export files using FTP. Alcatel-Lucent recommends that you administer the FTP accounts in a common way over the main 5620 SAM and auxiliary servers to ensure that OSS applications have access to files in the same way on all servers. For example, assign the same FTP user and password on both servers. The FTP home directory for each user can be configured relative to the location where the 5620 SAM server stores accounting and other generated files.

Note — The accounting statistics file export can generate a large number of files that consume disk space. By default, files are deleted every 15 minutes. OSS developers can manage the size of the files that are stored locally or on auxiliary servers by configuring the following attribute located in the <logToFile section of the nms/config/nms-server.xml file, located on the main 5620 SAM server.

- timeToKeepFile — Configures how long to keep the file created by the registerLogToFile request on the disk before purging. The maximum is 600 minutes. The default is 15 minutes.

  Code 15-2 shows an example of attribute assignment entries to manage the size of files that are stored locally.

Code 15-2: Sample attribute assignment entries

```
 <logToFile
....
....
timeToKeepFile="15"
.... />
```

Figure 15-1 the process associated with event driven export of statistics from a main 5620 SAM server. The 5620 SAM-O client must first register for notifications using the <registerLogToFile> method. Files are generated as accounting files are read from the network elements.

Figure 15-1 Event driven file export to 5620 SAM main server using LogToFile method

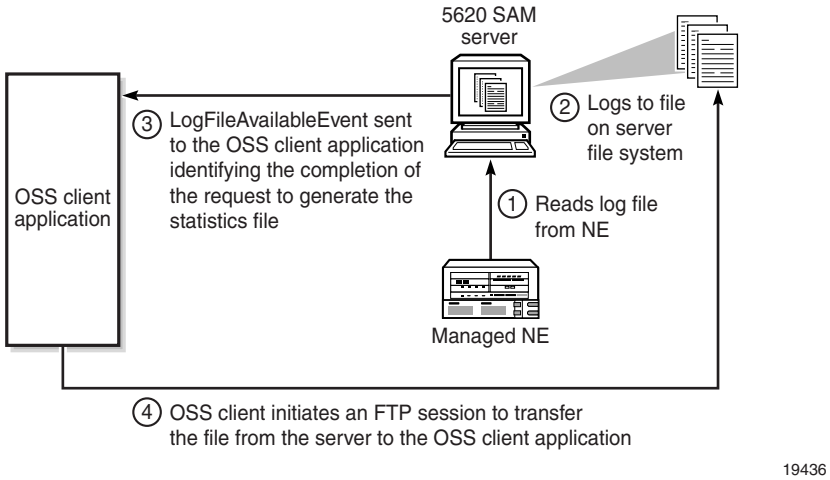
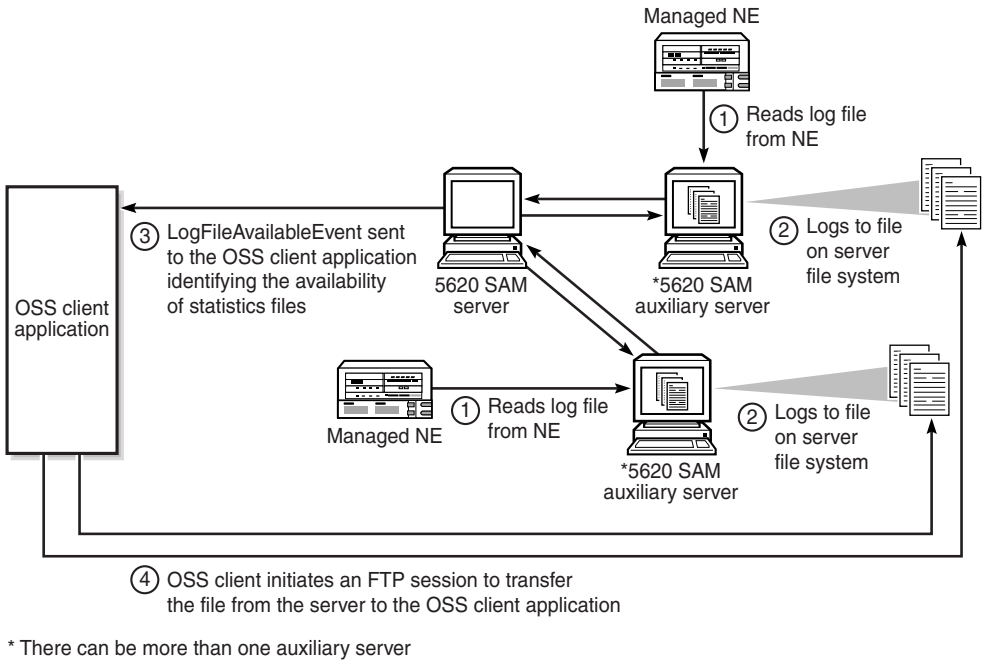


Figure 15-2 the process associated with event driven export of statistics from 5620 SAM auxiliary servers. The 5620 SAM-O client must first register for notifications using the <registerLogToFile> method. Files are generated as accounting files are read from the network elements.

Figure 15-2 Event driven file export to 5620 SAM auxiliary servers using LogToFile method



A 5620 SAM-O client can deregister from receiving notifications for accounting statistics and stop the export file creation by sending a deregisterLogToFile request, as shown in Code 15-3.

Code 15-3: <deregisterLogToFile> request

```
<deregisterLogToFile xmlns="xmlapi_1.0">
<jmsClientId>JMS_client_ID<jmsClientId/>
</deregisterLogToFile>
```

A 5620 SAM-O client is also deregistered if the JMS session is closed and must register again to receive notifications.

Statistics retrieval example using the <registerLogToFile> method

The <fullClassName> element specifies the statistic type for the XML request to create the statistics export file. The service.CompleteSubscriberIngressPacketOctets and service.CompleteSubscriberEgressPacketOctets statistic are the objects shown in Code 15-4.

Code 15-4 shows a sample request to export filtered accounting statistics to a compressed XML file using the <registerLogToFile> method.

Code 15-4: Sample request to export accounting statistics to a compressed XML file using a filter

```
<registerLogToFile xmlns="xmlapi_1.0">
<fullClassName>service.CompleteSubscriberIngressPacketOctets,service.Complet
eSubscriberEgressPacketOctets</fullClassName>
      <dirName>client_directory</dirName>
<compress>true</compress> <jmsClientId>JMS_Client_ID</jmsClientId>
      <resultFilter>
```

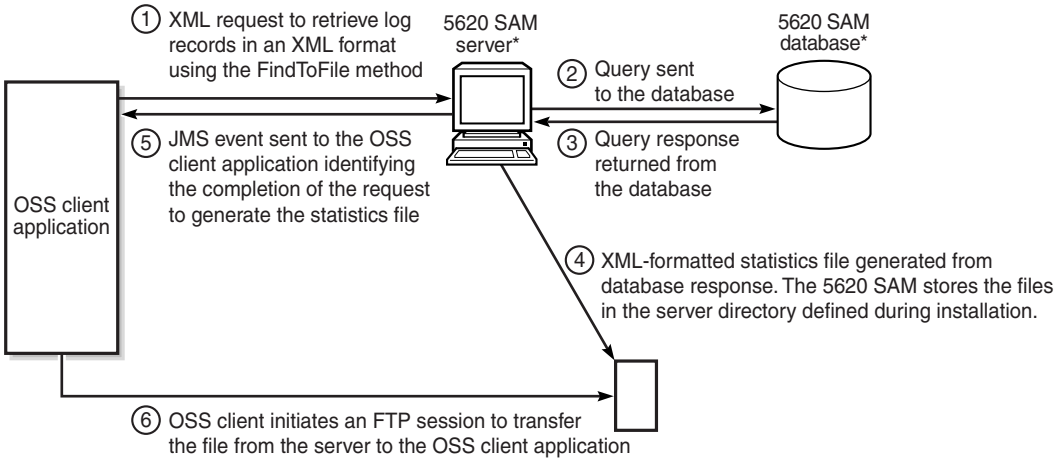
```
    <attribute>monitoredObjectPointer</attribute>
    <attribute>inProfileOctetsForwarded</attribute>
    <attribute>outOfProfileOctetsForwarded</attribute>
    <attribute>queueId</attribute>
    <attribute>timeRecorded</attribute>
  </resultFilter>
</registerLogToFile>
```

Accounting and performance statistics retrieval using findToFile

You can use the 5620 SAM-O and the <findToFile> method to transfer the accounting and performance statistics log records from the 5620 SAM database to your OSS client application. The <findToFile> method can save statistics for multiple NEs to one file. The JMS event FileAvailableEvent is generated each time a statistics file becomes available. See section "FileAvailableEvent" in chapter 4 for more information.

Figure 15-3 shows the process associated with the retrieval of statistics from the 5620 SAM database.

Figure 15-3 Synchronous statistics collection process



* The 5620 SAM server and database can be installed on the same or separate stations

18007

Note 1 — In addition to the illustrated scenario, step 4 of figure 15-3 can result in the following:

- if a file name is specified, a write file is created
- if an FTP address is specified, the 5620 SAM server sends the file to a URL without creating an intermediate file
- if the file location is on a remote disk that is read directly by the client application, the application may read or copy the file.

The OSS application is responsible for managing files and related disk space associated with these requests.

Note 2 — The 5620 SAM maps the object FDNs to a corresponding database index. Alcatel-Lucent recommends using FDN properties in queries to maximize the processing speed of the query. The 5620 SAM-O XML API queries should also use concrete classes rather than an abstract class to optimize performance.

Note 3 — There are two methods to export statistics from 5620 SAM— logtoFile and findToFile. The logToFile method is recommended to minimize collection latency and to reduce system load. For situations where less than 400 000 statistics records are retrieved in 15 minutes and where longer collection latency is tolerable, the findToFile method can be used.

See chapter 14 for more information about the <findToFile> method.

Statistics retrieval examples using the <findToFile> method

The <fullClassName> element specifies the statistic type for the XML request to retrieve statistics. The equipment.InterfaceAdditionalStats statistic is the object shown in Code 15-5 and Code 15-6.

Code 15-5 shows a sample request to export performance statistics to an XML file using the <findToFile> method.

Code 15-5: Sample request to export performance statistics to an XML file

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope
  xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP:Header>
        <header xmlns="xmlapi_1.0">
            <security>
                <user>user name</user>
                <password>MD5-hashed password</password>
            </security>
            <requestID>clientName@requestId</requestID>
        </header>
    </SOAP:Header>

    <SOAP:Body>
        <findToFile xmlns="xmlapi_1.0">
            <fullClassName>equipment.InterfaceAdditionalStats</fullClassName
>
            <filter>
            </filter>
    <fileName>ExportStatsToXmlFile_dump.xml</fileName>
    <resultFilter>
<children/>
            </resultFilter>
        </findToFile>
    </SOAP:Body>
</SOAP:Envelope>
```

Code 15-6 shows a sample request to export a filtered set of performance statistics using the <find> method.

Code 15-6: Sample request to export performance statistics using the XML API and a filter

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope
  xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
    <SOAP:Header>
        <header xmlns="xmlapi_1.0">
            <security>
                <user>user name</user>
                <password>MD5-hashed password</password>
            </security>
            <requestID>clientName@requestId</requestID>
        </header>
    </SOAP:Header>
    <SOAP:Body>
        <find xmlns="xmlapi_1.0">
    <fullClassName>equipment.InterfaceAdditionalStats</fullClassName>
    <filter>
        <or>
            <equal name="monitoredObjectSiteId" value="10.1.202.95"/>
            <equal name="monitoredObjectSiteId" value="10.1.202.94"/>
        </or>
    </filter>
        </find>
    </SOAP:Body>
</SOAP:Envelope>
```

Recovery of accounting statistics where OSS is using registerLogToFile

You can use the 5620 SAM-O and the <registerLogToFile> method to recover the accounting statistics log records from the 5620 SAM server to your OSS client application.

OSS connection lost

If an OSS loses connectivity to the 5620 SAM server or auxiliary server, the OSS needs to retrieve the logToFile export files on reconnect. These files will still be available in the export file directory for retrieval and processing by the OSS providing the configured time to keep these files was not exceeded. By default, the time allowed is 15 minutes. Contact your Alcatel-Lucent OIPS technical support representative for information about how to configure the 5620 SAM to extend the time to keep statistics files in the export directory.

The files should be retrieved from all of the auxiliary servers. For networks that do not have auxiliary servers, the OSS should retrieve the files from both the active and standby 5620 SAM server.

OSS connection lost and JMS client unsubscribed

If the OSS JMS subscription for an OSS client becomes unsubscribed, the 5620 SAM server deregisters that logToFile session and stops creating export files after a configured time period. By default, the time period is 20 minutes. Contact your Alcatel-Lucent OIPS technical support representative for information about how to configure the 5620 SAM to extend the time to continue creating export files after the OSS JMS subscription has been removed.

Note — Alcatel-Lucent does not recommend the use of findToFile to retrieve missed accounting statistics log records if the logToFile is the method used for retrieving accounting statistics data. Such findToFile requests may introduce excessive processing on the 5620 SAM server, which may hinder 5620 SAM functionality.

Recovery of accounting and performance statistics where OSS is using findToFile

You can use the 5620 SAM-O and the <findToFile> method to recover the accounting statistics log records from the 5620 SAM database to your OSS client application.

OSS connection lost

If an OSS loses connectivity to the 5620 SAM server and statistics record are not being retrieved, the OSS needs to keep track of the last time statistics records were received for relevant objects. When connectivity is restored the OSS should send multiple findToFile requests for specific objects and time periods which have missed statistics data. The OSS must be aware of the different Statistics Policy retention times so that it does not request older log records that have been purged from the 5620 SAM database.

Missing performance statistics from 5620 SAM

It is possible that in between 5620 SAM performance statistics collection intervals, there are no statistics detected by the OSS. This means that performance statistics log records are missing from the 5620 SAM database. If these statistics records are missing, the 5620 SAM server does not attempt to retry collection from the network elements. Therefore the OSS should not attempt to retrieve the statistics log records for that time, and instead continues to process statistics for the next collection interval.

XML statistics output file

Statistics retrieved from the 5620 SAM database in an XML format contain a set of elements and attributes that define the statistic class.

Code 15-7 shows an extract from the XML output file for the statistic counter receivedTotalOctets (ifHCInOctets from IF-MIB) in the equipment.InterfaceAdditionalStats statistic object.

Code 15-7: XML output example for statistics

```
<findToFileResponse xmlapi_1.0">
    <equipment.InterfaceAdditionalStatsLogRecord>
        <monitoredObjectClass>equipment.PhysicalPort</monitoredObjectClass>
        <monitoredObjectPointer>network:10.1.202.93:shelf-1:cardSlot-1:card:daughterCardSlot-
1:daughterCard:port-3</monitoredObjectPointer><displayedName>Port
1/1/3</displayedName><monitoredObjectSiteId>10.1.202.93</monitoredObjectSiteId>
        <monitoredObjectSiteName>sim202_93</monitoredObjectSiteName>
        <timeCaptured>1127878285113</timeCaptured>
        <periodicTime>938610</periodicTime>
<suspect>false</suspect>
        <objectFullName>equipment.InterfaceAdditionalStatsLogRecord.equipment.InterfaceAdditi
onalStats30-346</objectFullName>
        <name>equipment.InterfaceAdditionalStats30-346</name>
        <createdOnPollType>ScheduledFullNodeResync</createdOnPollType>
        <updatedOnPollType>ScheduledFullNodeResync</updatedOnPollType>
        <recordId>346</recordId>
        <bucketId>30</bucketId>
        <deploymentState>0</deploymentState>
        <receivedTotalOctets>0</receivedTotalOctets>
        <receivedTotalOctetsPeriodic>0</receivedTotalOctetsPeriodic>
        <receivedUnicastPackets>0</receivedUnicastPackets>
        <receivedUnicastPacketsPeriodic>0</receivedUnicastPacketsPeriodic>
        <receivedMulticastPackets>0</receivedMulticastPackets>
        <receivedMulticastPacketsPeriodic>0</receivedMulticastPacketsPeriodic>
        <receivedBroadcastPackets>0</receivedBroadcastPackets>
        <receivedBroadcastPacketsPeriodic>0</receivedBroadcastPacketsPeriodic>
        <transmittedTotalOctets>0</transmittedTotalOctets>
        <transmittedTotalOctetsPeriodic>0</transmittedTotalOctetsPeriodic>
        <transmittedUnicastPackets>0</transmittedUnicastPackets>
        <transmittedUnicastPacketsPeriodic>0</transmittedUnicastPacketsPeriodic>
        <transmittedMulticastPackets>0</transmittedMulticastPackets>
        <transmittedMulticastPacketsPeriodic>0</transmittedMulticastPacketsPeriodic>
        <transmittedBroadcastPackets>0</transmittedBroadcastPackets>
        <transmittedBroadcastPacketsPeriodic>0</transmittedBroadcastPacketsPeriodic>
        <children-Set></children-Set>
    </equipment.InterfaceAdditionalStatsLogRecord>
</findToFileResponse>
```

Table 15-2 provides a description of the XML elements and attributes shown in Code 15-7.

Table 15-2 Description of the statistic XML element attributes

Element	Attribute	Description
<class>	name	Package and class name for the statistic class being retrieved from the database
	xmlTag	Alias of the class name if a conflict exists between two class names
	displayed	Name of the class displayed on the 5620 SAM GUI
	cli	CLI command to view the statistics on the node
<property>	name	Name of the statistic counter
	type	Type of statistic
	default	Default value of the statistic counter, if applicable
	loggableType	Method for storing the counter on the node. The counter can be stored as raw data, an absolute counter, or as a rate. In the case of an absolute counter, a Periodic Statistic Counter is also stored on the 5620 SAM and calculates the delta between the values retrieved for the current and previous intervals
	snmpCounter	MIB information for the counter
	dependsOn	Parent counter for the statistic. Applicable for periodic statistics
<Description>	—	Description of the statistic counter function

JMS performance statistics

The following 5620 SAM server JMS-specific performance statistics are available for collection:

- **JMS Publisher Events**

  Class info:
  - statistics.JmsPublisherQueueStats—holds details for the Publisher Queue (Main server)
- **JMS Server Events**

  Class info:
  - statistics.JmsSubscriberSessionStats—contains details of JMS subscriber Session count
  - statistics.JmsSubscriberTopicStats—contains details of JMS subscriber count for a given XML topic

Because these statistics can have a performance impact on the server, real-time collection is not supported.

15.7 Sample request to retrieve historical statistics from a router

Code 15-8 shows a sample request to retrieve historical statistics from a router using the <findToFile> method. An exception-free response indicates that the 5620 SAM-O received and validated the request.

Code 15-8: Sample request to retrieve historical statistics from a router

```
 <SOAP:Body>
        <findToFile xmlns="xmlapi_1.0">
            <fullClassName>equipment.InterfaceAdditionalStatsLogRecord</fullClassName>
            <filter>
                <and>
                    <equal name="monitoredObjectPointer"
value="network:10.1.202.93:shelf-1:cardSlot-1:card:daughterCardSlot-1:daughterCard:port-3"/>
                    <between name="timeCaptured" first="1127142900000"
second="1127143800000"/>
                </and>
            </filter>
            <fileName>Equipment.InterfaceAdditionalStatsLogRecord.xml</fileName>
        </findToFile>
    </SOAP:Body>
```

15.8 Sample request to collect interface statistics

Code 15-9 shows a sample request to perform the following tasks using the <findToFile> method:

- collect interface statistics
- format the collected statistics in a XML file for post-processing

To return the data using the XML API, you can create a similar set of filters and use the <find> method.

The following conditions apply to the sample shown in Code 15-9:

- uses the <findToFile> method, which can use any class from any other package
- specifies the <propertyFormat>, which indicates the number of parameters (elements) to retrieve, either all or configOnly
- exports the collected data to an XML file, with a filename of stats_file.xml

When you create export files using the <findToFile> method, you should be aware of the following:

- the export files are placed in a directory specified during installation
- no FTP server is installed during the 5620 SAM database installation, so to export files from the specified directory there can be an FTP server installed on the machine where the database is installed, if necessary. If these files are located on a remote disk, FTP may not be necessary to access them.

An exception-free response indicates that the 5620 SAM-O received and validated the request.

Code 15-9: Sample request to collect interface statistics

```
 <SOAP:Body>
        <findToFile xmlns="xmlapi_1.0">
            <fullClassName>equipment.InterfaceAdditionalStatsLogRecord</fullClassName>
            <filter>
                <and>
                    <equal name="monitoredObjectPointer"
value="network:10.1.202.93:shelf-1:cardSlot-1:card:daughterCardSlot-1:daughterCard:port-3"/>
                    <between name="timeCaptured" first="1127142900000"
second="1127143800000"/>
                </and>
            </filter>
            <fileName>Equipment.InterfaceAdditionalStatsLogRecord.xml</fileName>
        </findToFile>
    </SOAP:Body>
```

Configuration management

16 — Configuration management overview

17 — Equipment configuration management

18 — Routing protocol configuration management

19 — Customer and residential subscriber configuration management

20 — Policy configuration management

21 — Service configuration management

22 — XML API service template configuration

16 — Configuration management overview

16.1 Configuration management overview *16-2*

16.2 Generic methods *16-2*

16.3 Deployers *16-5*

16.4 Workflow to handle deployer failures *16-13*

16.1 Configuration management overview

You can use the 5620 SAM-O to configure objects based on the XML schema. The user privileges associated with the OSS user define the ability to configure 5620 SAM objects. See section 14.2 for more information about the 5620 SAM equipment and logical object hierarchies.

When you send XML API requests (with dependent relationships) between the objects being modified and where the changes cause side effects on those objects, Alcatel-Lucent recommends that you use a configurable sleep timer between the requests. The sleep timer configuration depends on several factors, including:

- network latency
- SNMP trap incoming rate on the 5620 SAM server
- processing load on the 5620 SAM server
- potential SNMP trap loss rate

See the 5620 SAM-O XML Reference for information about object relationships.

Alcatel-Lucent recommends that you configure and delete network objects one object at a time, to facilitate troubleshooting of potential problems, such as deployer failures.

16.2 Generic methods

The 5620 SAM-O uses the generic package to define methods that are applicable to all objects. The XML schema contains package-specific methods to configure objects; for example, equipment.DaughterCardSlot.configureDaughterCard. Support for the package-specific configuration methods will be removed in a future release of the 5620 SAM-O.

Note — Alcatel-Lucent strongly recommends using the generic package for the configuration of all network objects.

Table 16-1 lists some of the method that you can use with the generic package.

Table 16-1 Object configuration using the generic package

Method	Description
generic.GenericObject.configureInstance	Modifies an existing object and associated children. Not recommended for OSS clients as there are no return parameters.
generic.GenericObject.configureInstanceWithResult	Modifies an existing object and associated children. The method returns the newly configured parameters on the object and child, including parameters set due to associated conditions. See the 5620 SAM-O XML Reference for input parameters. See Code 16-1 for an example of the method.

(1 of 2)

Method	Description
generic.GenericObject.configureChildInstance	Creates a child of the object with the desired configuration values for the child or children objects, and optional grandchildren. See the 5620 SAM-O XML Reference for input parameters. See Code 16-2 for an example of the method.
generic.GenericObject.configureChildInstanceWithResult	Creates a child of an existing object. The method returns the newly configured parameters on the child object and grandchildren, including parameters set due to associated conditions. See the 5620 SAM-O XML Reference for input parameters. See Code 16-3 for an example of the method.
generic.GenericObject.deleteInstance	Delete an existing object. See Code 16-4 for an example of the method.

(2 of 2)

The following section provides sample requests using the objects and methods associated with the configuration and management of equipment operations.

Caution — The messages in this section are examples of the SOAP XML request format. Use the sample as a base to build your request. Ensure that you test your request before network deployment.

Sample request to create a channel on a SONET port

Code 16-1 shows a sample request to create a channel on a SONET port. The response to the request contains the newly created objects and the associated parameters.

Code 16-1: Sample request to create a channel on a SONET port

```
 <SOAP:Body>
        <generic.GenericObject.configureInstanceWithResult
xmlns="xmlapi_1.0">
            <deployer>immediate</deployer>

<distinguishedName>network:10.1.202.93:shelf-1:cardSlot-2:card:daughterCardS
lot-2:daughterCard:port-1</distinguishedName>
            <includeChildren>true</includeChildren>
            <configInfo>
                <sonetequipment.Sts48Channel>
                    <actionMask><bit>create</bit></actionMask>
                    <administrativeState>portInService</administrativeState>
                    <portId>1</portId>
                    <mode>access</mode>
                    <encapType>bcpDot1qEncap</encapType>
                    <localChannelId>0</localChannelId>
                    <displayedLocalChannelId>1</displayedLocalChannelId>
                    <portChannelType>sonetSts48</portChannelType>
                    <children-Set>
                        <sonetequipment.SonetChannelOverheadSpecifics>
                            <actionMask><bit>modify</bit></actionMask>

<isPayloadScramblingEnabled>true</isPayloadScramblingEnabled>
                            <children-Set/>
                        </sonetequipment.SonetChannelOverheadSpecifics>
                    </children-Set>
                </sonetequipment.Sts48Channel>
            </configInfo>
```

```
        </generic.GenericObject.configureInstanceWithResult>
    </SOAP:Body>
```

Sample request to configure a SONET port

Code 16-2 shows a sample request to configure a SONET port. The response to the request contains the FDNs of the newly created objects.

Code 16-2: Sample request to configure a SONET port

```
 <SOAP:Body>
        <generic.GenericObject.configureChildInstance xmlns="xmlapi_1.0">
            <deployer>immediate</deployer>

<distinguishedName>network:10.1.202.93:shelf-1:cardSlot-2:card:daughterCardS
lot-2:daughterCard:port-1</distinguishedName>
            <childConfigInfo>
                <sonetequipment.SonetPortSpecifics>

<objectFullName>network:10.1.202.93:shelf-1:cardSlot-2:card:daughterCardSlot
-2:daughterCard:port-1:sonetPort</objectFullName>
                    <actionMask><bit>modify</bit></actionMask>
                    <clockSource>nodeTimed</clockSource>
                    <children-Set/>
                </sonetequipment.SonetPortSpecifics>
            </childConfigInfo>
        </generic.GenericObject.configureChildInstance>
    </SOAP:Body>
```

Sample request to create a network interface

Code 16-3 shows a sample request to create a network interface using the rtr.RoutingInstanceSite.configure method. The network interface consists of an IP address, a subnet mask, and any policy settings that define the QoS properties of the interface. (You can also define parameters for ACL, ICMP, ARP, and NTP Broadcast.) You must define the network interface on the port that is physically connected and share routing information and IP traffic.

The response to the request contains the FDN of the managed object.

Code 16-3: Sample request to create a network interface

```
 <SOAP:Body>
        <generic.GenericObject.configureChildInstanceWithResult
xmlns="xmlapi_1.0">
            <deployer>immediate</deployer>
            <synchronousDeploy>true</synchronousDeploy>
            <deployRetries>1</deployRetries>
            <!-- in milliseconds -->
            <deployRetryInterval>10000</deployRetryInterval>
            <distinguishedName>network:10.1.202.225:router-1</distinguishedN
ame>
            <childConfigInfo>
                <rtr.NetworkInterface>
                    <actionMask><bit>create</bit></actionMask>
                    <displayedName>Network-Interface-Example</displayedName>
                    <id>24</id>
                    <portPointer>network:10.1.202.225:shelf-1:cardSlot-1:car
d:daughterCardSlot-1:daughterCard:port-7</portPointer>
                    <physicalAddress>00-00-00-00-00-00</physicalAddress>
                    <networkPolicyObjectPointer>Network:1</networkPolicyObje
ctPointer>
                    <children-Set>
                        <rtr.VirtualRouterIpAddress>
```

```
                            <index>1</index>
                            <actionMask><bit>create</bit></actionMask>
                            <ipAddress>12.12.16.1</ipAddress>
                        </rtr.VirtualRouterIpAddress>
                    </children-Set>
                </rtr.NetworkInterface>
            </childConfigInfo>
            <resultFilter>
                <attribute>objectFullName</attribute>
                <attribute>displayedName</attribute>
                <children recursive="yes" />
            </resultFilter>
        </generic.GenericObject.configureChildInstanceWithResult>
    </SOAP:Body>
```

Sample request to remove a daughter card slot

Code 16-4 shows a sample request to remove a daughter card slot. An exception-free response indicates that the 5620 SAM-O received and validated the request.

Code 16-4: Sample request to remove a daughter card slot

```
 <SOAP:Body>
        <generic.GenericObject.deleteInstance xmlns="xmlapi_1.0">
        <deployer>immediate</deployer>

<distinguishedName>network:10.1.202.93:shelf-1:cardSlot-2:card:daughterCardS
lot-2</distinguishedName>
        </generic.GenericObject.deleteInstance>
    </SOAP:Body>
```

16.3 Deployers

Deployers model the communication requests made from the management domain to the network. Deployers track changed objects and parameters (elements) that need to be sent to the network. When the deployer performs the network change, all data is cleared from the deployer. If the deployer fails, it attempts to redeploy the request based on the deployer policy configuration.

Caution — Network performance can be adversely affected by numerous queries that gather large amounts of data.

For example, to gather the containment hierarchy of 50 000 services that have three sites, one access interface for each site, and three circuits requires more than 800 Mbytes of space to store the report, and can take many hours to complete.

When changes must be sent to the network, deployers are created, queued, and dispatched to managed routers. If the deployment is successful, the 5620 SAM notifies the OSS that the deployment is completed for the request which triggered the deployment. If the deployment fails, an alarm creation or change event is generated and sent using JMS. See the following sections for more information:

- "Deployer failures" in this section for more information about managing deployer failures
- "Deployer alarms" in this section for more information about the deployer alarm format
- chapter 13 for more information about JMS events

Deployment types <deployer>

All configuration methods require a specified deployer type, as specified in the <deployer> tag. The deployer type is specified in the xmlApiTypes.xsd file. Deployers specify how and when deployment occurs. The valid values (with numeric equivalents) are:

- immediate (-1) to deploy to the network immediately
- followTransaction (-2) reserved for future use
- postponed (-3) reserved for future use
- ignored (-4) to not deploy to the network, so changes are only added to the database

Synchronous and asynchronous requests

There are two types of deployments you can configure:

- asynchronous (default)
- synchronous

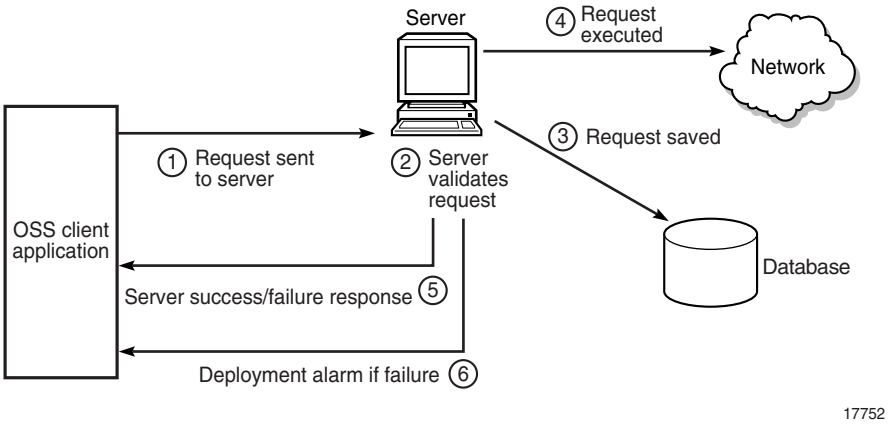
Users can specify synchronous network requests using the <synchronousDeploy> and other tags.

Synchronous requests

Synchronous requests are validated by the 5620 SAM-O and saved to the database. When the deployers have successfully performed the deployment across the network, or when all the retries specified in the request are completed, the synchronous request result is returned.

Figure 16-1 shows a synchronous network deployment.

Figure 16-1 Database interaction with synchronous network deployment



The time required to complete each deployment request depends on the SNMP deployment parameters and complexity of the request. If a network element is not reachable through SNMP, it may take much longer for the request to complete.

Multiple requests can be sent in one SOAP body using streamed HTTP connections. The successful or failed results are streamed back to the OSS application in the order processed. Request processing stops for the SOAP body when the first failed request is encountered or when all requests are successful. When a failure occurs, subsequent requests in the same multi-part request are not processed.

Code 16-5 shows a synchronous configuration request with the following details:

- attempt to synchronously deploy the request to the network twice as indicated in the <deployRetries> tag
- retry the deployment in 1000 s intervals as indicated in the <deployRetryInterval>

Caution — The following sample message is an example of the request format. Use the sample as a base to build your request. Ensure that you test your request before network deployment.

Alcatel-Lucent recommends that you do not change the default value (0) of the <deployRetries> parameter.

Code 16-5: Sample synchronous configuration request

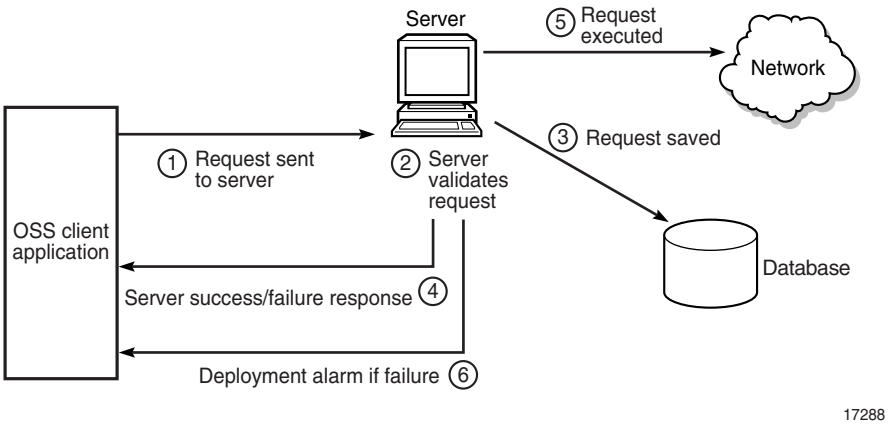
```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope
xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP:Header>
<header xmlns="xmlapi_1.0">
<security>
<user>user name</user>
<password>MD5-hashed password</password>
</security>
<requestID>clientName@requestId</requestID>
</header>
</SOAP:Header>
<SOAP:Body>
<generic.GenericObject.deleteInstance xmlns="xmlapi_1.0">
<deployer>immediate</deployer> <synchronousDeploy>true</synchronousDeploy>
<deployRetries>2</deployRetries>
<deployRetryInterval>1000</deployRetryInterval>
<distinguishedName>network:10.1.202.93:shelf-1:cardSlot-2:card</distinguishe
dName>
</generic.GenericObject.deleteInstance>
</SOAP:Body>
</SOAP:Envelope>
```

Asynchronous requests

Asynchronous requests are validated, saved to the database, a response is sent back for the request, and the request is queued for deployment to the network. The default setting is that requests are made asynchronously across the network.

Figure 16-2 shows an asynchronous network deployment.

Figure 16-2 Database interaction with asynchronous network deployment



The sample synchronous request shown in Code 16-5 can be changed to an asynchronous request by changing <synchronousDeploy> to false:

```
<synchronousDeploy>false</synchronousDeploy>
```

Deployer failures

The 5620 SAM-O returns control to the OSS following a deployer failure and the OSS is notified of the deployer failure. The OSS must subsequently perform procedures for error-recovery.

If the OSS connection is lost or times out before the 5620 SAM-O returns a result to the OSS application, the OSS may not be aware of any subsequent success or failure notifications of the request. Steps must be taken at a later time, either manually or through the OSS, to verify whether the request was successful. If the request was unsuccessful, for example, because of a failed deployer, the OSS must perform procedures for error-recovery.

Note 1 — It is recommended that the OSS application record configuration failures for future reference in error recovery or troubleshooting.

Note 2 — When failures occur, deployers are left in an undeployed state and must be cleared. Uncleared deployments consume resources on the 5620 SAM server and can block further synchronous OSS requests.

Alcatel-Lucent recommends the following strategy to minimize the use of resources and the blocking of synchronous OSS requests:

- Set the <clearOnDeployFailure> parameter to true to automatically clear deployers when a failure occurs.
- Do not change the default value (0) of the <deployRetries> and <deployRetryInterval> parameters.

Contact your Alcatel-Lucent OIPS technical support representative for information about how to optimize deployer performance so that resource consumption and the blocking of synchronous OSS requests to network elements is minimized.

Table 16-2 describes generic methods that can be used in error recovery.

Table 16-2 Generic methods for error recovery

Generic method	Description	Dependencies
generic.GenericObject.getDeployers	To find deployer IDs for deployers with the failed deployment states set in the generic.GenericObject.deploymentState parameter. See $core.DeploymentState in the *5620 SAM-O XML Reference* to see all the valid failure values.	—
generic.GenericObject.getDeployer	To find all parameter values for a specific deployer ID	—
generic.GenericObject.clearDeployer	To clear the deployer with the specified deployer ID	—
generic.GenericObject.triggerResync	To perform a non-scheduled resynchronization of an object	—
generic.GenericObject.getDeployers Shallow	To find matched deployers without children hierarchy	—

Filtering can be applied when retrieving information about deployers using the following methods:

- generic.GenericObject.getDeployers
- generic.GenericObject.getDeployer
- generic.GenericObject.getDeployersShallow

Code 16-6 shows an example of using the class attribute to define the filter class using the genericObject.

Code 16-6: Sample of defining the filter class using the genericObject

```
<filter>
    <and class="generic.DeployerInfo">
    <notEqual value="16" name="state" />
    <wildcard value="Default.DeployerBank:depl-%" name="objectFullName" />
   </and>
 </filter>
```

Code 16-7 shows a failed response for a deployer request. When all retries fail, the failure response indicates the failed deployer ID. An alarm is not raised until all retries are attempted. See Appendix C for information on how to obtain a sample request to retrieve deployer information.

Code 16-7: Sample failed response for a deployer request

```
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP:Header>
 <header xmlns="xmlapi_1.0">
    <requestID>clientName@requestId</requestID>
    <requestTime>11-Dec-2006 10:53:48 AM</requestTime>
    <responseTime>11-Dec-2006 10:53:49 AM</responseTime>
 </header>
 </SOAP:Header>

<SOAP:Body>
    <generic.GenericObject.configureChildInstanceException
xmlns="xmlapi_1.0">
    <description>
[ app: generic ] [ class: generic.GenericObject ] [ instance: N/A ] [ descr:
Operation Failed During Deployment ]
    </description>
   <deployers>
       <deployer>Default.DeployerBank:depl-482</deployer>
       <deployer>Default.DeployerBank:depl-480</deployer>
   </deployers>
 </generic.GenericObject.configureChildInstanceException>
 </SOAP:Body>
</SOAP:Envelope>
```

Note — The <responseTime> tag in a 5620 SAM-O header is the time at which the response stream is opened.

Deployer performance

The performance of deployers varies with the type of action:

When object creation requests succeed, the object is created in the database and a deployer deploys the change to the network. If deployment fails, you must clear the deployers to remove the previously created object from the database.

When object modification requests succeed, the changed object is saved to the database before deployment. If deployment fails, clearing a deployment does not reset the modified object in the database. To reset to the previous object setting, resynchronize the object by using the generic.GenericObject.triggerResync method.

When object deletion requests succeed, the objects are marked for deletion in the database but not immediately removed. If deployment fails, clearing a deployment completes the object deletion.

See "Workflow to handle deployer failures" in this chapter for more information about handling a deployment error that has caused the 5620 SAM database to be out of synchronization with the managed network.

Deployer alarms

OSS applications can use the requestId and the requestUser to match failures with specific requests. Once the deployerId is extracted from the alarm, you can use the deployerId to query a specific deployer. See Appendix C for information on how to obtain a sample request to retrieve deployer information.

Code 16-8 shows a sample deployment alarm. The following information is contained in the deployer alarm:

- the alarmClass attribute identifies the alarm class as a deployment failure
- the deployer ID is last component of the objectName attribute, in this case deployerId=2
- the additionalText attribute also contains the deployerId, the requestId, and the client user (requestUser) that made the request in the format:

  deployerId=2;requestId=client1:0;requestUser=*user name*;deploymentType=1, where *user name* is the account name of the operator account in the SOAP request

Code 16-8: Sample deployment alarm

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP:Header>
      <header xmlns="xmlapi_1.0">
         <requestID>clientName@requestId</requestID>
      </header>
   </SOAP:Header>
   <SOAP:Body>
      <fm.FaultManager.findFaultsResponse xmlns="xmlapi_1.0">
         <result>
            <fm.AlarmInfo>
               <severity>minor</severity>
               <previousSeverity>indeterminate</previousSeverity>
               <originalSeverity>minor</originalSeverity>
               <highestSeverity>minor</highestSeverity>
               <probableCause>11</probableCause>
               <alarmName>13</alarmName>
               <type>5</type>
               <affectedObjectFullName>network:10.1.202.93:shelf-1:cardSlot-
1:card:daughterCardSlot-1:daughterCard:port-5</affectedObjectFullName>
               <affectedObjectClassName>equipment.PhysicalPort</affectedObje
ctClassName>
               <isAcknowledged>false</isAcknowledged>
               <wasAcknowledged>false</wasAcknowledged>
               <acknowldegedBy>N/A</acknowldegedBy>
               <firstTimeDetected>1128369209032</firstTimeDetected>
               <lastTimeDetected>1128369209032</lastTimeDetected>
               <lastTimeSeverityChanged>0</lastTimeSeverityChanged>
               <lastTimeCleared>0</lastTimeCleared>
               <lastTimePromoted>0</lastTimePromoted>
               <lastTimeDemoted>0</lastTimeDemoted>
               <lastTimeEscalated>0</lastTimeEscalated>
               <lastTimeDeEscalated>0</lastTimeDeEscalated>
```

```
                <lastTimeAcknowledged>0</lastTimeAcknowledged>
                <frequency>0</frequency>
                <occurences>N/A</occurences>
                <numberOfOccurences>2</numberOfOccurences>
                <numberOfOccurencesSinceClear>2</numberOfOccurencesSinceClear
>
                <numberOfOccurencesSinceAck>0</numberOfOccurencesSinceAck>
                <isServiceAffecting>false</isServiceAffecting>
                <additionalText>deployerId=771;requestId=AreqGenericObjectCon
figureInstance-CLIENT-admin-5620SAM@13@138.120.135.164-15;requestUser=admin;
deploymentType=3;</additionalText>
                <operatorAssignedUrgency>indeterminate</operatorAssignedUrgen
cy>
                <urgencyAssignedBy>1</urgencyAssignedBy>
                <relatedObjects>
                   <null/>
                </relatedObjects>
                <affectingObjects>
                   <relationshipTreeList>
                      <relationshipTree>
                         <objectFullName>Network
Queue:default</objectFullName>
                         <subtree/>
                      </relationshipTree>
                   </relationshipTreeList>
                </affectingObjects>
                <nodeId>10.1.202.93</nodeId>
                <nodeName>sim202_93</nodeName>
                <affectedObjectDisplayedName>Port
1/1/5</affectedObjectDisplayedName>
                <applicationDomain>equipment</applicationDomain>
                <displayedClass>PhysicalPort</displayedClass>
                <alarmClassTag>generic.DeploymentFailure</alarmClassTag>
                <affectedObjectClassIndex>87</affectedObjectClassIndex>
                <affectedObjectInstanceIndex>5</affectedObjectInstanceIndex>
                <deployerName>N/A</deployerName>
                <deployerId>0</deployerId>
                <requestId>N/A</requestId>
                <requestUser>N/A</requestUser>
                <objectFullName>faultManager:network@10.1.202.93@shelf-1@card
Slot-1@card@daughterCardSlot-1@daughterCard@port-5|alarm-13-5-11-deployerId=
771</objectFullName>
                <objectClassName>fm.AlarmObject</objectClassName>
                <allomorphicClassName>fm.AlarmObject</allomorphicClassName>
                <objectId>-578593836</objectId>
                <displayedName>Port 1/1/5 -
network@10.1.202.93@shelf-1@cardSlot-1@card@daughterCardSlot-1@daughterCard@
port-5|alarm-13-5-11-deployerId=771</displayedName>
                <lifeCycleState>1</lifeCycleState>
                <deploymentState>0</deploymentState>
                <neId>10.1.202.93</neId>
             </fm.AlarmInfo>
          </result>
       </fm.FaultManager.findFaultsResponse>
    </SOAP:Body>
</SOAP:Envelope>
```


Deployer simulation

By default, the 5620 SAM server deploys configuration changes to network elements through SNMP. You can configure the 5620 SAM-O to save configuration changes in the 5620 SAM database without deploying the changes, which facilitates OSS development in the absence of real equipment or simulators.

You can use the 5620 SAM-O to configure a mediation.DeploymentPolicy with a mediation.DeploymentMode property set to 1, to disable deployment. The default setting is 2, SNMP.

See chapter 20 for more information about how to configure policies.

Caution 1 — Since configuration errors may not always be caught by the 5620 SAM server before configuration changes are committed to the database, you must disable deployer simulation and test with live network equipment to validate changes prior to OSS deployment in the production network.

Caution 2 — Deployment should never be disabled in a production network.

16.4 Workflow to handle deployer failures

The following workflow outlines the high-level steps necessary to handle a deployment error that may cause the 5620 SAM database to be out of synchronization.

1 Clear the deployer.

2 Create a SOAP request to resynchronize the object using the generic.GenericObject.triggerResync method.

Note 1 — When you use the generic.GenericObject.triggerResync method for a specific instanceName, you can specify whether a resynchronization on the object is performed by setting the resyncSelf flag to true, or whether the resync is performed on its children by setting the resyncChildren flag to true. For example:

```
<generic.GenericObject.triggerResync xmlns="xmlapi_1.0">
<instanceName>network:10.1.186.88:shelf-1:cardSlot-1:car
d:daughterCardSlot-1:daughterCard:port-1</instanceName>
<resyncSelf>true</resyncSelf>
<resyncChildren>false</resyncChildren>
</generic.GenericObject.triggerResync>
```

The 5620 SAM does not notify 5620 SAM-O clients when objects are resynchronized.

Note 2 — Contact your Alcatel-Lucent account or OIPS technical support representative for more information about how to manage deployer failures.

3 Perform one of the following steps, depending on your 5620 SAM-O client implementation.

a Set the object back to the previous setting. To reset the previous object setting, resynchronize the object by using the generic.GenericObject.triggerResync method.

See section "Deployer performance" for more information about the different request types that you must consider when you set an object back to the previous setting.

For example, if a request to create an object with children objects successfully creates the parent object, but creates only some of the children objects, you should create a request to delete the partially created objects.

For partial failures on a request to modify a port mode, the OSS client should create a request to change the port mode to return it to the original setting.

For failed deletion requests, performing step 2 resynchronizes the 5620 SAM with the network. Depending on the state the deleted object or objects in the 5620 SAM, the 5620 SAM-O client should create a request to re-create the deleted objects to restore the 5620 SAM and the network back to their original state before the failed deployment.

Warning — Subsequent requests to set objects back to their previous settings may fail. After an unsuccessful attempt to set objects back to their previous setting, Alcatel-Lucent recommends that the OSS client not retry.

Note — The 5620 SAM continues to try and deploy the change, even after the failure, according to the deployment policies configured for the router. If an automatic recovery mechanism is implemented by the OSS application, the OSS application should use a recovery mechanism that is consistent with the configured deployer policies. For example, if the policy defines that deployers should redeploy after a failure, the OSS application should use a timer mechanism to wait until the deployment is retried before attempting another recovery method.

b Raise an alarm for the administrator to investigate the deployment failure.

In many cases, such as a loss of connectivity, deployments that fail will continue to fail if retried. If an OSS client continues to retry deployments after a failure, 5620 SAM resources will be consumed that will be ineffective, ultimately until all deployer resources are consumed. Alcatel-Lucent recommends that you do not continue to retry failed deployment requests. Instead, the OSS client can raise an alarm to alert the administrator that there is a deployment problem.

Note — Contact your Alcatel-Lucent OIPS technical support representative for information about how to manage deployer failures.

17 — Equipment configuration management

17.1 Equipment configuration overview *17-2*

17.2 Workflow to configure equipment *17-2*

17.3 Workflow to configure a router *17-3*

17.4 Generic NE profiles *17-3*

17.5 Workflow to create a generic NE profile *17-3*

17.6 Device software upgrades *17-4*

17.1 Equipment configuration overview

You can use the 5620 SAM-O to configure equipment based on the XML schema. The user privileges associated with the OSS user define the ability to configure 5620 SAM objects.

The 5620 SAM-O XML interface uses the following packages to manage equipment:

- equipment for physical equipment
- fr for frame relay equipment
- ethernetequipment for Ethernet equipment
- sonetequipment for SONET and SDH equipment
- sonettiming for SONET and SDH timing
- tdmequipment for TDM equipment

Note — Alcatel-Lucent recommends using the generic package for configuring all objects in the above mentioned packages. See section 16.2 for more information about generic methods. See section 14.2 for more information about the 5620 SAM equipment object hierarchies.

17.2 Workflow to configure equipment

Use the following steps to configure equipment.

1. Configure each card slot. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure the card type in a router.
2. Configure each daughter card slot. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure a daughter card slot.
3. Configure all ports. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure an Ethernet port.
4. Configure the channels, if necessary. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to create a channel on a SONET port.

17.3 Workflow to configure a router

Use the following steps to configure a router.

1. Configure the router name and system ID.
2. Configure the network interfaces, which are associated with network ports. The configuration of the interfaces requires the following substeps:
   - **i** Assign a name to the interface.
   - **ii** Associate IP addresses, a network port, and QoS policy settings to the interface.
   - **iii** Enable IPv6 for the interface if required.

   Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain the following sample requests to create a network interface:

   - sample request to create a default network ingress policy
   - sample request to create a network interface
3. Cable the network ports on the router to the network ports on other routers.
4. Create static routes, if required. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure a static route.
5. Configure a routing policy. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure a routing policy.
6. Configure and enable the associated routing protocol. See chapter 18 for more information.

17.4 Generic NE profiles

The 5620 SAM uses profiles to discover generic network elements. After the generic network element profiles are created, the network elements can be discovered.

17.5 Workflow to create a generic NE profile

1. Configure the product name and description.

   Note — You cannot modify the generic NE type after you create the generic NE profile. You must delete the generic NE profile and create a new profile with the generic NE type. You cannot delete the generic NE profile if any network elements have been discovered using the profile.

2. Enter a valid system object ID.

The system object ID can be one of the following types:

- system ID of the network element
- partial system ID to facilitate the discovery of different network element types

Go to step 3 if the devices discovered by the profile are using the script management features of 5620 SAM.

3 Enter the common CLI-related parameters used for read and write CLI access, such as:

```
commandPrompt

disablePagingCommand

errorIndicator

resetCommand

twoStepsLogin
```

Note — Network element that use the script management features of 5620 SAM should have the ability to disable paging. The disablePagingCommand attribute controls network element paging.

4 Enter the valid CLI commands for read and write access privileges, such as:

```
readLoginPrompt

readPasswordPrompt

writeAccessLoginCommand

enablingWriteAccessLoginPrompt

writeLoginPrompt

writePasswordPrompt
```

Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to create a generic network element profile.

17.6 Device software upgrades

The 5620 SAM supports software upgrades of managed devices via the OSS interface. Software images must be extracted relative to the *installation_directory*/server/nms/nodeSoftware directory on the 5620 SAM server. Files in the nodeSoftware directory are automatically deleted by the 5620 SAM after a configurable period of time expires. The nodeSoftware timeToKeepFile property in the nms-server.xml file specifies the period of time, in minutes, that the 5620 SAM will hold files in the nodeSoftware directory. The default value of the nodeSoftware timeToKeepFile property is 1440.

Procedure 17-1 To configure the nodeSoftware timeToKeepFile property

Note — In a redundant 5620 SAM deployment, you must perform this procedure on each main server in the 5620 SAM system.

1. Log on to a 5620 SAM main server station as the samadmin user.
2. Open a console window.
3. Navigate to the server configuration directory, typically /opt/5620sam/server/nms/config.
4. Open the nms-server.xml file using a plain-text editor.
5. Locate the following line:

   ```
   <nodeSoftware timeToKeepFile="time" />
   ```

   where

   The value of *time* specifies the period of time, in mins, that the 5620 SAM waits before deleting files in the nodeSoftware directory. The default value is 1440.
6. Modify the value of the timeToKeepFile property, as required.
7. Save and close the nms-server.xml file.
8. Navigate to the server binary directory, typically /opt/5620sam/server/nms/bin.
9. Enter the following:

   ```
   bash$ ./nmsserver.bash read_config ↵
   ```

   The main server reads the nms-server.xml file and puts the configuration changes into effect.

18 — Routing protocol configuration management

18.1 Routing protocol configuration management overview *18-2*

18.2 Workflow to configure a routing protocol *18-4*

18.3 Workflow to configure a PIM or IGMP multicast protocol *18-7*

18.4 VRRP virtual router configuration *18-7*

18.5 Workflow to configure a virtual router *18-8*

18.1 Routing protocol configuration management overview

You can use the 5620 SAM-O to configure network protocol communications and connections, such as MPLS LSPs, to allow customer traffic to flow from the access network through the core network.

Supported routing protocols include:

- BGP
- RIP
- OSPF
- LDP
- IS-IS
- L2TP

The 5620 SAM supports the following multicast protocols:

- PIM
- IGMP
- MSDP
- MLD

Note 1 — Alcatel-Lucent recommends using the generic package for configuring all objects. See section 16.2 for more information about generic methods. See section 14.2 for more information about the logical object hierarchy for protocols.

Note 2 — See the *5620 SAM User Guide* for more information on routing protocols and MPLS.

BGP

BGP is an inter-AS routing protocol. An AS is a network or a group of devices logically organized and controlled by a common network administration. BGP enables devices to exchange network reachability information. AS paths are the routes to each destination. There are two types of BGP: IBGP and EBGP.

- IBGP is used to communicate with peer devices in the same AS. Routes received from a device in the same AS are not advertised to other devices in the same AS but can be advertised to an EBGP peer.
- EBGP is used to communicate with peers in different ASs. Routes received from a device in a different AS can be advertised to both EBGP and IBGP peers.

MP-BGP

As BGP was designed to distribute IPv4 routing information, the addition of VPN IPv4 addresses required an extension to BGP. The extension is MP-BGP. The new addressing now includes a 12-byte address, consisting of a eight-byte RD and the four-byte IPv4 address. To use MP-BGP:

- MP-BGP must be enabled on all PE devices from the VPN tab of the BGP configuration form
- all PE devices must be connected as peers

RIP

RIP is an IGP that uses a distance-vector algorithm to determine the best route to a destination, using hop count as the deciding factor. In order for the protocol to provide complete information about routing, every device in the domain must participate in the protocol. RIP, a UDP-based protocol, updates its neighbors, and the neighbors update their neighbors.

Unlike OSPF and other link-state protocols, RIP directly advertising reachability information to its neighbors. RIP advertises reachability information by sending prefix, mask, and either hop count or cost metric data. Each device running the RIP protocol advertises all RIP devices periodically by sending RIP update PDUs. The route with the lowest metric is advertised as the best route.

LDP

LDP is used to distribute labels in non-traffic-engineered MPLS applications. Routers can establish LSPs across a network by mapping network-layer routing information directly to the data link layer switched paths. After LDP distributes the labels to the LSR, the LSR assigns the label to a FEC, and then informs all other LSRs in the path about the label and how the label will switch data accordingly.

A FEC is a collection of common actions associated with a class of packets. LDP helps establish the LSP by providing a set of procedures that LSRs can use to distribute labels.

When a service tunnel is configured on managed routers using LDP signaling in an MPLS environment, LDP sessions are set up based on the configured hello and other PDU values. If another service tunnel is created to the same destination, the LDP session is reused.

IS-IS

IS-IS is a link-state interior gateway protocol that uses the shortest path first algorithm to determine a route. Routing decisions are made using the link-state information. IS-IS entities include:

- networks, which are autonomous system routing domains
- intermediate systems, which are routers, such as the 7750 SR and 7710 SR
- end systems, which are network devices that send and receive PDUs

End systems and intermediate system protocols allow devices and nodes to identify each other. The IS-IS protocol sends link state updates periodically through the network, so each device can maintain current network topology information.

PIM

PIM is a component of multicast routing that defines the one-to-many or many-to-many transmission of information. You can use the following variations for PIM configurations:

- sparse mode
- dense mode
- source-specific multicast
- bidirectional

Sparse mode is the most common PIM configuration. Sparse mode is used for data transmission to nodes in multiple Internet domains that contain a small ratio of nodes that subscribe to the multicast traffic. Dense mode is used when a large ratio of the potential nodes subscribe to the multicast traffic. In source-specific multicast, paths originate at a single, defined source. Bidirectional PIM is not source-specific.

MSDP

MSDP is a protocol that enables multiple PIM-SM domains to communicate with each other using their own RPs. MSDP also enables multiple RPs in a single PIM-SM domain to establish MSDP mesh-groups and to synchronize information between anycast RPs about the active sources being served by each anycast RP peer. The 7750 SR and 7710 SR support MSDP.

IGMP

The 5620 SAM supports IGMP configuration for core router functionality and IP services.

18.2 Workflow to configure a routing protocol

Use the following steps to configure a BGP, RIP, OSPF, LDP, ISIS, or MPLS routing instance.

1. For a BGP routing instance:

   i. Configure an AS number. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure an autonomous system number on the routing instance.

   ii. Configure a number for the confederation-autonomous system if your configuration requires confederations. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure a confederation-autonomous system number on the routing instance.

iii Enable BGP on a router. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to enable a BGP confederation.

iv Configure global-level BGP elements. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure global-level BGP parameters.

v Configure a BGP confederation. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure a BGP/RIP/OSPF/LDP/ISIS/MPLS routing instance.

vi Configure a peer group-level BGP. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure a BGP peer group.

vii Configure a peer-level BGP. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure a BGP peer.

2 For a RIP routing instance:

i Enable RIP on a router. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure a BGP/RIP/OSPF/LDP/ISIS/MPLS routing instance.

ii Configure a global-level RIP. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure a global-level RIP.

iii Configure a group-level RIP. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure a group-level RIP.

3 For an OSPF routing instance:

i Enable OSPF on a router. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure a BGP/RIP/OSPF/LDP/ISIS/MPLS routing instance.

ii Configure OSPF elements.

iii Configure an OSPF area. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to create an OSPF area.

iv Add an L3 interface to the OSPF area. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to assign an L3 interface to an OSPF site.

4 For an LDP routing instance:

   i Enable LDP on a router. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure a BGP/RIP/OSPF/LDP/ISIS/MPLS routing instance.

   ii Configure global-level LDP elements. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure global-level LDP parameters.

   iii Configure LDP interfaces. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure LDP interfaces.

   iv Configure LDP targeted peers. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure LDP targeted peers.

5 For an ISIS routing instance:

   i Enable ISIS on a router. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure a BGP/RIP/OSPF/LDP/ISIS/MPLS routing instance.

   ii Configure router-wide ISIS elements. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure router-wide ISIS parameters.

   iii Configure ISIS NET addresses.

   iv Configure ISIS interfaces.

6 For an MPLS routing instance:

   i Enable MPLS routing on all applicable routers and the L3 interfaces. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure a BGP/RIP/OSPF/LDP/ISIS/MPLS routing instance.

   ii Assign a network interface to an MPLS instance. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to assign a network interface to an MPLS instance.

   iii Create a full mesh of MPLS paths, which is the route to be used by an LSP, between the routers using the Layer 3 interfaces. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to create an MPLS path.

   iv Create a full mesh of LSPs that use the MPLS paths. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to create an LSP.

18.3 Workflow to configure a PIM or IGMP multicast protocol

Use the following steps to configure a PIM or IGMP multicast instance.

1 For PIM:

   i Enable PIM. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to enable PIM.

   ii Configure global-level PIM. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure global PIM.

   iii Configure a PIM source-specific multicast group. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure a PIM source-specific multicast group.

   iv Configure a PIM static rendezvous point. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure a PIM static rendezvous point.

   v Configure a PIM interface. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure a PIM interface.

   vi Configure the PIM candidate rendezvous point. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure the PIM candidate rendezvous point.

   vii Configure the PIM anycast rendezvous point. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure the PIM anycast rendezvous point.

2 For IGMP:

   i Enable IGMP. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to enable IGMP.

   ii Configure IGMP. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure IGMP.

   iii Configure an IGMP interface. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure an IGMP interface.

18.4 VRRP virtual router configuration

VRRP allows the creation of a redundant routing system that takes over packet transmissions on a common LAN segment when a router fails. VRRP designates alternative routing paths in the form of a virtual router, without changing the IP address or MAC address of a protected router. The 5620 SAM-O supports the creation of virtual routers on IES services and core network LAN services.

With VRRP, a protected router owns the IP address of the virtual router. In the role of master, the owner router normally forwards packets to hosts on the default gateway. If the owner router fails, the virtual router, which has the same IP address as the owner, shifts packet-forwarding responsibilities to a designated backup router. The backup router then becomes the master router and forwards packets on the virtual router IP address.

18.5 Workflow to configure a virtual router

1. Ensure that an L3 interface primary IP address is available for the network or IES service. Each IP interface must have a primary IP address.
2. Configure a virtual router on a core network LAN or an IES service.
   - **a** Use the following steps to configure a Virtual Router on a core network LAN.
     - **i** Create a global VRRP priority-control policy for non-owner VRRP instances. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to create a global VRRP priority-control policy for non-owner VRRP instances.
     - **ii** Distribute the global VRRP priority-control policy to a non-owner VRRP router. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to distribute a global VRRP priority-control policy to a non-owner VRRP router.
     - **iii** Create a virtual router for a core network LAN. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to create a virtual router for a core network LAN.

iv Create an owner VRRP instance in the virtual router as required. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to create an owner VRRP in a network virtual router.

v Create a non-owner VRRP instance in the virtual router as required. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to create a non-owner VRRP instance in a network virtual router.

b Use the following steps to configure a Virtual Router on an IES service.

i Create a global VRRP priority-control policy for non-owner VRRP instances. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to create a global VRRP priority-control policy for non-owner VRRP instances.

ii Distribute the global VRRP priority-control policy to a non-owner VRRP router. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to distribute a global VRRP priority-control policy to a non-owner VRRP router.

iii Create a virtual router for an IES service. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to create a virtual router for an IES.

iv Create an owner VRRP instance in the IES virtual router as required. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to create an owner VRRP instance in an IES virtual router.

v Create a non-owner VRRP instance in the IES virtual router as required. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to create a non-owner VRRP instance in an IES virtual router.

19 — Customer and residential subscriber configuration management

19.1 Customer and residential subscriber configuration overview *19-2*

19.2 Workflow to configure and manage customers *19-3*

19.3 Workflow to configure and manage residential subscribers *19-3*

19.1 Customer and residential subscriber configuration overview

The emergence of residential broadband networks and the availability of multiple service offerings, such as triple play applications, have created a requirement for greater service differentiation and control at the level of the individual service recipient. The service recipient is called a residential subscriber in the 5620 SAM.

Note — Customers are called subscribers in the 5620 SAM releases earlier than Release 4.0. The differentiation of service recipient types has not changed in the 5620 SAM-O object model. The 5620 SAM-O continues to use the subscr package to manage customers. The 5620 SAM-O uses the ressubscr package to create and manage residential subscribers.

Customer

The 5620 SAM-O allows you to manage customers. An end user is the recipient of application content that is delivered through a service. The service is a means of transport for the application content and is owned by a service subscriber. For example, an end user subscribes to an high-speed Internet access service, and the Internet access service is owned by a service provider who is a service customer of the network provider. The 5620 SAM-O provides only customer management, not end-user subscriber management.

Note — The 5620 SAM-O uses the subscr package to create and manage customers.

Residential subscriber

The 5620 SAM-O provides functionality for the efficient provisioning of access, QoS, and security features on IES and VPLS for residential subscribers on the 7450 ESS-7, 7450 ESS-12, 7750 SR-7, and 7750 SR-12. Service-level customization for end users is available through systems such as the 5750 SSC.

In the context of the 5620 SAM-O, a residential subscriber, sometimes called a subscriber, has a unique identifier that associates a group of end-user devices with policies. A subscriber can be associated with multiple SAPs on multiple NEs, and a customer can be associated with multiple subscribers.

A subscriber host, sometimes called a host, is an end-user device, such as a computer, VoIP telephone, or set-top box, that connects to the provider network and receives the service traffic. Hosts that use the same subscriber identifier share overall HQoS and accounting characteristics as defined in a customer SLA, but may use QoS policies and queues that differ by the type and class of service offering.

Note 1 — The 5620 SAM-O uses the ressubscr package to create and manage residential subscribers.

Note 2 — Alcatel-Lucent recommends using the generic package for configuring all objects. See section 16.2 for more information about generic methods. See section 14.2 for more information about the logical object hierarchy for subscriber and residential subscriber ojbjects.

19.2 Workflow to configure and manage customers

Use the following steps to configure and manage customers.

1. Create customers that purchase and use services. See Appendix C for information on how to obtain a sample request to create or modify a customer.
   - a. Configure or modify key customer contact information.
   - b. Associate equipment or resources to the customers, if required.
2. Create services for customers. See Appendix C for information on how to obtain a sample request to create a VPLS.
3. Monitor or troubleshoot subscribers based on the SLAs between the customer and the service provider.
   - a. Retrieve customer information and contact the subscriber if service problems occur or maintenance activities are scheduled.
   - b. Use 5620 SAM tools to monitor alarms that are raised against the customer-assigned equipment or services.
   - c. Perform the appropriate diagnostics to troubleshoot problems.

19.3 Workflow to configure and manage residential subscribers

1. Create a service that effectively delivers the proposed service offerings.
2. Create ingress and egress scheduler policies for the subscriber hosts in accordance with the customer SLA.
3. Create an accounting policy for the customer.
4. Create access ingress and access egress QoS policies for the different applications, for instance, HSI and VoIP, and levels of service, for example gold, silver, and bronze, that subscriber hosts are to receive. Ensure that the queues defined in the QoS policies reference the previously created scheduler policies.
5. Create a unique subscriber identification string for the subscriber.

6 Create SLA profiles that name the previously created QoS policies.

- Create a different SLA profile for each class of service offering.
- Use override values to customize the policy values, as required.

See Appendix C for information on how to obtain a sample request to create an SLA profile.

7 Create a subscriber profile.

- Choose the previously created ingress and egress schedulers.
- Choose the previously created accounting policy.
- Enable accounting.
- Associate one or more of the previously created SLA profiles with the subscriber profile.

See Appendix C for information on how to obtain a sample request to create a subscriber profile.

8 Develop scripts for extracting the DHCP Option 82 information.

9 Create a subscriber identification policy.

See Appendix C for information on how to obtain a sample request to create a subscriber identification policy.

10 Create a subscriber explicit map, if required.

11 Analyze the subscriber profile and SLA profile assignments with the Profile Assignment Analysis Tool.

12 Supply the customer with the subscriber identification string and any IP information that is required for provisioning the residential subscriber hosts.

13 Assign the SLA profile, subscriber profile, and subscriber identification profile to the service, if required.

See Appendix C for information on how to obtain a sample request to configure subscriber management on an access interface.

14 Turn up the service after the subscriber hosts are provisioned.

20 — Policy configuration management

20.1 Policy configuration overview *20-2*

20.2 Workflow to configure ACL IP and MAC filter policies *20-7*

20.3 Workflow to configure routing management policies *20-7*

20.1 Policy configuration overview

The 5620 SAM-O supports the template-based creation of policies. There are three types of policies:

- service management
- routing management
- network management

The 5620 SAM-O supports the creation and modification of policies using the following packages:

- accounting
- aclfilter
- acl
- aengr
- aingr
- atmpolicy
- file
- mediation
- multicast
- niegr
- nqueue
- pae802_1x
- policy
- portscheduler
- qos
- rp
- snmp
- slope
- squeue
- svq
- tod
- vs
- telcoaclfilter
- telcoqos

Note — Alcatel-Lucent recommends using the generic package for configuring all objects. See section 16.2 for more information about generic methods. See section 14.2 for more information about the logical object hierarchy for all policy objects.

The abstract policy class and the policy manager control all QoS policies. All other QoS policies inherit from classes in this package.

The qos policy class defines the basic types related to quality of service, such as enumerations, that are common to and used by other QoS-related packages, such as acl, aclfilter, file, niegr, and vs. There are no methods associated with this package.

Service management policies

Service management policies specify how service traffic is handled by network resources such as interfaces, ports, daughter cards, and circuits. Service management policies are globally and seamlessly distributed to routers when they are used by resources on the router. You can also manually distribute the policies to the routers. Service management policies include:

- access ingress
- access egress
- network policy
- slope
- network queue
- scheduler
- ACL IP filter
- MAC IP filter

See "Workflow to configure ACL IP and MAC filter policies" for more information.

aingr package

Access ingress policies are applied to access interfaces and specify QoS on ingress.

Access ingress policies define ingress service forwarding class queues and map flows to those queues. You cannot delete the following queues when you create an access ingress policy:

- default unicast traffic
- default multipoint traffic

The queues exist within the definition of the policy. The queues are only instantiated in hardware when the policy is applied to an access interface. In the case where the service does not have multipoint traffic, the multipoint queue is not instantiated. In the simplest access ingress policy, all traffic is treated as a single flow and mapped to a single queue, and all flooded traffic is treated with a single multipoint queue.

The required access ingress policy elements include:

- a unique access ingress policy ID
- at least one default unicast forwarding class queue
- at least one multipoint forwarding class queue

The optional access ingress policy elements include:

- additional unicast queues up to a total of eight for each of the eight forwarding classes
- additional multipoint queues up to three for each forwarding class for each type of multipoint traffic (broadcast, multicast, and destination unknown unicast)
- QoS policy match criteria to map packets to a forwarding class

Each queue can have unique queue parameters to allow individual policing and rate shaping of the flow mapped to the forwarding class. Mapping flows to forwarding classes is controlled by comparing each packet to the match criteria in the policy. There is one default access ingress policy. The default policy gives all traffic equal priority with the same chance of being sent or dropped during periods of congestion.

You can use forwarding subclasses for additional access ingress packet classification. You can associated one or more subclasses with each forwarding class. The designations for forwarding subclasses are the same as the designation used for the forwarding classes. Each subclass assumes the behavior of the parent forwarding class. The two-tier class designation provides enhanced classification for access ingress QoS policies.

aengr package

Access egress policies are applied to access egress interfaces and specify QoS on egress.

Access egress policies define egress service queues and map forwarding class flows to queues. In the simplest access egress policy, all forwarding classes are treated as a single flow and mapped to a single queue.

The required access egress policy elements include:

- a unique access egress policy ID
- at least one defined default queue

The optional egress policy elements include:

- additional queues up to eight separate queues for each of the eight supported forwarding classes
- IEEE 802.1p priority value remarking based on forwarding class

Each queue in a policy is associated with one or more of the supported forwarding classes. Each queue can have its own queue parameters that allow individual rate shaping of the forwarding classes mapped to the queue. More complex service queuing models are supported on the managed devices where each forwarding class is associated with a dedicated queue.

niegr package

Network policies are applied to network interfaces and specify QoS on egress and ingress.

On ingress, a network policy maps incoming DSCP and EXP values to forwarding class and profile state for traffic received from the core network. On egress, the policy maps forwarding class and profile state to DSCP and EXP values for traffic to be transmitted into the core network.

Routing management policies

Routing management policies control the size and content of the routing tables, advertised routes, and the recommended route to each destination. You can use routing management policies to configure:

- AS paths
- communities
- dampings
- policy statements
- prefix lists

See "Workflow to configure routing management policies" for more information.

Network management policies

Network management policies specify how the 5620 SAM communicates with network resources, handles alarms, manages billing statistics, and stores information. Network management policies include:

- alarm
- file
- accounting

- mediation
- poller

Policy distribution mode

Global policies are policies that you create using the 5620 SAM-O or the 5620 SAM GUI, or that are created through node discovery or CLI. Local policies are instances of global policies that are assigned to individual NEs. Depending on the distribution mode configuration of a local policy, when you modify a global policy, all local instances of the policy can be automatically updated. This ensures that all instances of the policy in the network are synchronized.

Table 20-1 describes the policy distribution modes.

Table 20-1 Distribution mode

Option	Option description
Sync With Global	The local policy is synchronized with the global policy at all times. The local instance cannot be modified.
Local Edit Only	You can modify the local instance only, which affects the associated network element. Changes to the global policy do not affect the local policy unless a synchronization operation is manually performed.

When you distribute a global policy, local policies using the Sync With Global distribution mode allow the NE to receive the policy.

Note — Local policies using the Local Edit Only distribution mode do not allow the NE to receive the distribution of a global policy. You must ensure that the policy distribution mode for the local policy is set to Sync With Global if you want the NE to receive the distribution of a global policy.

The isPolicyDiscoveredInLocalEditOnlyMode flag in the nms-server.xml file determines the distribution mode to which a policy is set when it is created through node discovery or CLI. By default, the flag is set to false and local policies that are created through NE discovery or CLI are set to Sync With Global distribution mode.

```
<policyConfig  isPolicyDiscoveredInLocalEditOnlyMode="false"/>
```

If you set the isPolicyDiscoveredInLocalEditOnlyMode flag to true in the nms-server.xml file, local policies that are discovered from the network or created through CLI are set to Local Edit Only mode.

Use the setDistributionModeToLocalEditOnly method to change a specific policy to Local Edit Only distribution mode. Use the setDistributionModeToSyncWithGlobal method to change a specific policy to Sync With Global distribution mode and automatically synchronize with the previous released global policy. See the *5620 SAM-O XML Reference* for more information.

Note — You must have the appropriate permissions to use these methods. See the *5620 SAM User Guide* for more information about 5620 SAM user groups and the associated scope of command roles and permissions.

Policy configuration mode

The isPolicyConfigurationModeUsed flag in the nms-server.xml file determines the configuration mode in which a policy is created through the 5620 SAM-O. By default, the flag is set to false, or released mode.

```
<policyOssiConfig isPolicyConfigurationModeUsed="false" />
```

Global policies that are created by local policy discovery are in draft mode.

When the isPolicyConfigurationModeUsed flag is set to false, global policies that are configured through the 5620 SAM-O are automatically released and distributed to local policies. If you update a policy, regardless of the configuration mode, the 5620 SAM-O accepts the changes and redistributes the policy.

When you distribute a global policy, local policies using the Sync With Global distribution mode allow the NE to receive the policy.

Note — Local policies using the Local Edit Only distribution mode do not allow the NE to receive the distribution of a global policy. You must ensure that the policy distribution mode for the NE is set to Sync With Global if you want the NE to receive the distribution of a global policy.

When the isPolicyConfigurationModeUsed flag is set to false and you attempt to distribute a policy that is in draft mode—created or modified from the 5620 SAM GUI and not yet released—the policy is distributed to the network and the local instances are updated.

If you set the isPolicyConfigurationModeUsed flag to true in the nms-server.xml file, all global policies are created in draft mode. If the policy is later released, the 5620 SAM-O returns the policy to draft mode when any subsequent modifications to the policy are made. When a draft policy is bound to a service and a local policy does not exist, the 5620 SAM deploys the backup policy, or previously released policy, and distributes the policy to the associated site. If a backup policy does not exist, an exception is raised.

Use the setConfigurationModeToReleased method to release and distribute a draft policy. See the *5620 SAM-O XML Reference* for more information.

Note — You must have the appropriate permissions to use this method. See the *5620 SAM User Guide* for more information about 5620 SAM user groups and the associated scope of command roles and permissions.

20.2 Workflow to configure ACL IP and MAC filter policies

Use the following steps to configure ACL IP and MAC filter policies.

1. Create the filter. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to find ACL IP filters.
2. Distribute the filter. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to distribute an ACL IP filter.
3. Synchronize the filter. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to synchronize an ACL IP filter.

20.3 Workflow to configure routing management policies

Use the policy.Manager.configure method, as defined in policyMethods.xsd and policyTypes.xsd files, and the following steps to configure routing management policies.

1. Configure the following parameters for AS paths:
   - **i** Specify a path name. The range is 1 to 32 characters.
   - **ii** Specify a description, if required. The range is 0 to 80 characters.
   - **iii** Specify a regular expression. The range is 1 to 80 characters.
2. Configure the following parameters for communities:
   - **i** Specify a community name.
   - **ii** Configure the community members for the community defined in step 2 i. You can configure multiple community members.
3. Configure the following parameters for Dampings:
   - **i** Specify a damping name. The range is 1 to 32 characters.
   - **ii** Specify a half-life value. The range is 0 to 45.
   - **iii** Specify a reuse value. The range is 0 to 20 000.
   - **iv** Specify a suppress value. The range is 0 to 20 000.
   - **v** Specify a maximum suppression value. The range is 0 to 720.

4 Configure the following parameters for prefix lists:

i Specify a prefix list name.

ii Specify the prefix list members to be added to the prefix list defined in step 4 i. You can configure multiple prefix list members. Configure the following parameters for each member:

- Specify a prefix. Enter an IP address.
- Specify a mask. Enter a network mask in bitmask format. The range is 0 to 32.
- Specify a type. The options are: Exact, Longer, Through, or Range.
- Configure the Begin Length and Through Length parameters. The availability of these parameters depends on the option selected for the Type parameter. The range for both parameters is 24 to 32. The value is in bitmask format.

5 Configure the following parameters for policy statements:

i Specify a policy statement name. The range is 1 to 32 characters.

ii Specify a description. The range is 0 to 80 characters.

iii Specify a default action. The options are: Reject or Accept.

iv Specify policy entries to add to the policy statement. You can configure multiple policy entries. Configure the following parameters for each policy entry:

- Specify an entry ID. The range is 131 072.
- Specify a description. The range is 0 to 80 characters.
- Specify an action. The options are: Reject or Accept.
- Specify the following parameters for the From Criteria:
  - AS Path Name. The range is 0 to 32 characters.
  - Protocol. The options are: None, Direct, Static, BGP, ISIS, OSPF, RIP, Aggregate, BGP-VPN, IGMP, or PIM.
  - Community List Name. The range is 0 to 32 characters.
  - Interface Name. The range is 0 to 32 characters.
  - ISIS Route Level. The options are: 0, 1, or 2.
  - ISIS External Route. The options are: True or False.
  - OSPF Route Type. The options are: 0, 1, or 2.
  - OSPF Area. Enter an IP address.
  - OSPF Origin. The options are: None, IGP, EGP, or Incomplete.
  - OSPF Area Set. The options are: True or False.
  - Neighbor IP Address. Enter an IP address.
  - Neighbor Prefix List Name. The range is 0 to 32 characters.
  - Prefix Lists. You can enter up to five prefix lists. The range is 0 to 32 characters.

v Specify the following parameters for the To Criteria:

- Protocol. The options are: None, BGP, ISIS, OSPF, RIP, or BGP-VPN.
- ISIS Route Level. The options are: 0, 1, or 2.
- Neighbor IP Address. Enter an IP address.
- Neighbor Prefix List Name. The range is 0 to 32 characters.

6 Construct a valid SOAP request.

7 Send the request.

Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure a policy.

21 — Service configuration management

21.1 Service configuration overview *21-2*

21.2 Workflow to configure a VLL, VPLS, IES, VPRN, or VLAN service *21-10*

21.3 Workflow to configure a mirror service *21-12*

21.4 Workflow to configure an optical transport service *21-12*

21.5 Composite services *21-13*

21.6 Workflow to configure a composite service *21-17*

21.7 IGMP snooping *21-18*

21.8 Workflow to configure IGMP snooping *21-18*

21.9 DHCP relay configuration *21-18*

21.10 Workflow to configure DHCP relay *21-19*

21.11 IPsec management *21-19*

21.12 Workflow to configure IPsec — option A *21-20*

21.13 Workflow to configure IPsec — option B *21-21*

21.1 Service configuration overview

You can use the 5620 SAM-O to create, configure, and manage services based on the XML schema. The user privileges associated with the OSS user define the ability to configure 5620 SAM objects.

The 5620 SAM-O XML interface uses the service.serviceManager (svc-mgr) package to manage services.

The 5620 SAM supports the following service types:

- VLL
- VPLS
- IES
- VPRN
- VLAN
- Mirror
- Optical transport service

Note — Alcatel-Lucent recommends using the generic package for configuring all objects in the service.serviceManager (svc-mgr) package. See section 16.2 for more information about generic methods. See section 14.2 for more information about the logical object hierarchy for the service packages listed above.

VLL

The 5620 SAM supports the provisioning of L2 VLL services on edge devices. A VLL is a pipe that connects access interfaces. A VLL that connects access interfaces on one router (or site) is called a local VLL service. A VLL that connects access interfaces on two routers (or sites) is called a distributed VLL service.

For the distributed VLL service, subscriber data enters the service through two access interfaces on different edge devices. The VLL is transported across an IP and/or IP/MPLS provider core network in circuits that are carried by service tunnels. Service tunnels are created using GRE or MPLS LSPs.

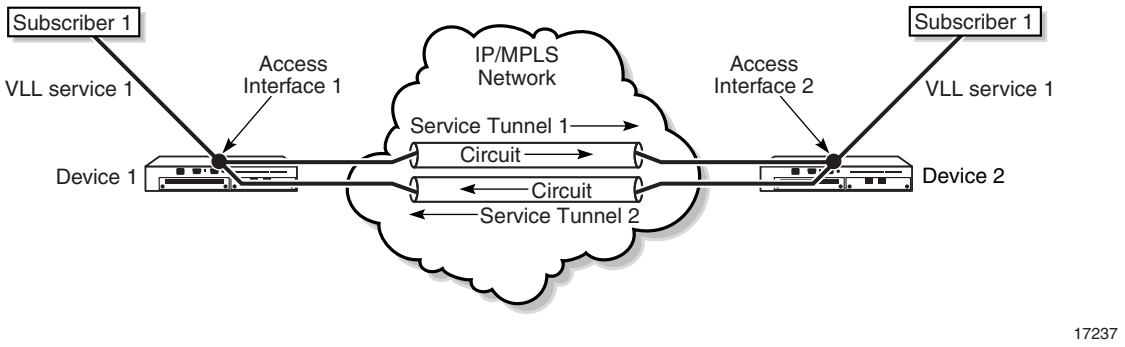
You can configure and specify circuits for the VLL service during service creation. You must create service tunnels before you create the VLL service.

Packets that arrive at an edge are associated with a VLL service based on the access interface on which they arrive. An access interface is uniquely identified by the:

- physical Ethernet port or POS port and channel
- encapsulation type (none or dot1q)
- encapsulation identifier (if encapsulation type is dot1q)

Figure 21-1 shows a sample VLL service.

Figure 21-1 Sample VLL service



VPLS

The 5620 SAM supports VPLS multipoint switched service on edge devices. VPLS is a class of VPN multipoint L2 service that connects multiple customer sites in a single, bridged domain. The service provider managed IP/MPLS network contains the domain. Customers sites in the VPLS appear to be on the same LAN, even when the sites are geographically dispersed. The advantages of VPLS include:

- uses an Ethernet interface on the customer access side to simplify provisioning
- enables customers to control and simplify routing strategies, as all routers in the VPLS are part of the same LAN, which simplifies IP addressing
- Independent of protocol. There is no Layer 2 protocol conversion between LAN and WAN technologies.

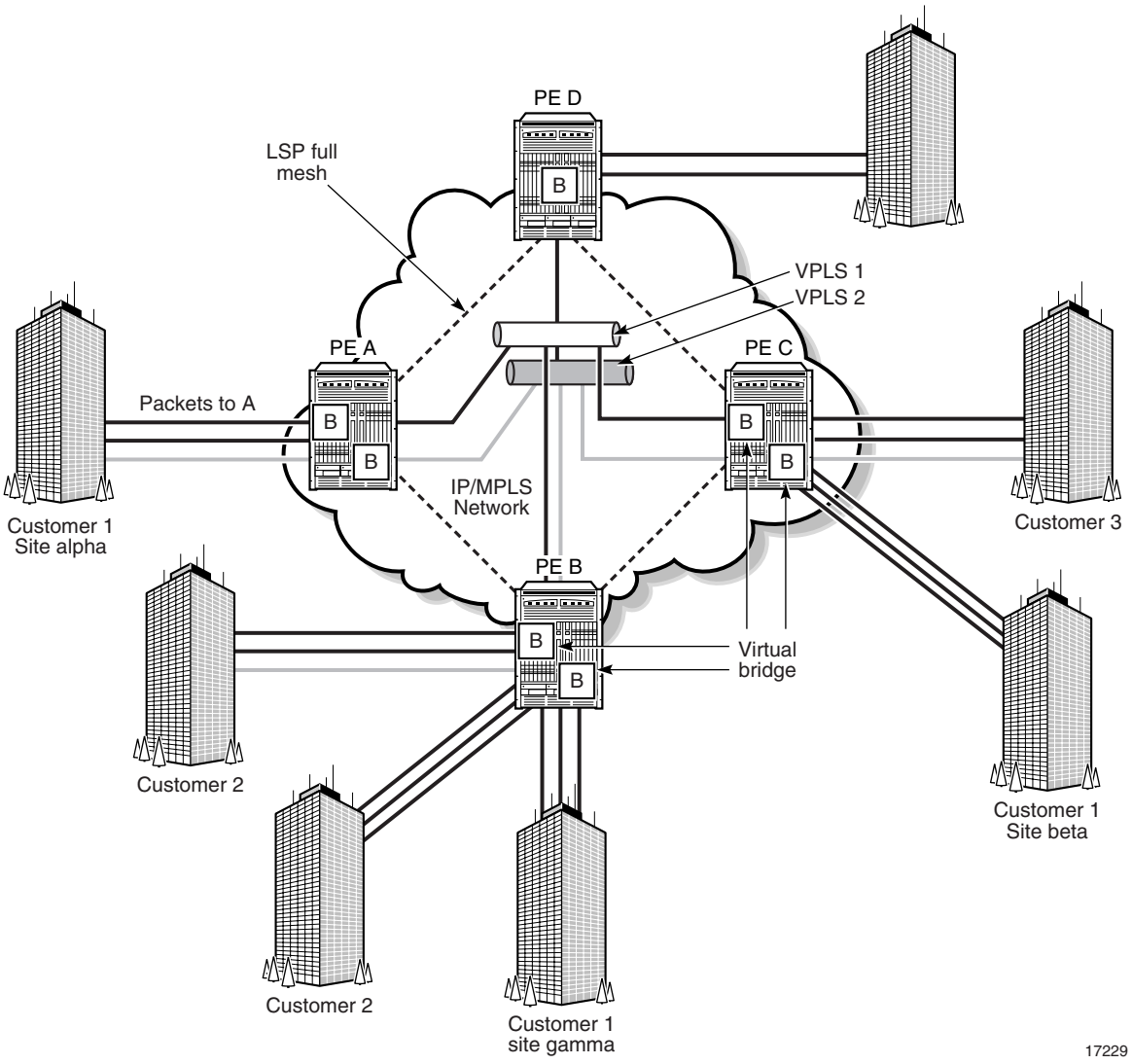
A VPLS can span single or multiple sites.

A VPLS that spans a single site is called a local VPLS. In a local VPLS, subscriber data enters the service through multiple access interfaces on a single PE device. Circuit provisioning is not required for the local VPLS.

A VPLS that spans multiple sites is called a distributed VPLS. In a distributed VPLS, customer data enters the service using two or more interfaces on different PE devices. The VPLS is transported by service circuits over an IP/MPLS provider core network that is carried by service tunnels. Service tunnels are an abstraction of the transport implementation. Service tunnels are created using GRE or MPLS LSPs.

Figure 21-2 shows a sample VPLS configuration.

Figure 21-2 Sample VPLS



IES

An IES is a routed connectivity service where the subscriber communicates with an IP router interface to send and receive Internet traffic. The IP router interface is an L3 interface.

IES allows customer-facing IP interfaces in the same routing instance to be used for connectivity in service network core routing. IES requires that the IP address scheme that is used by the subscriber is unique among other provider address schemes and potentially the entire Internet.

When you configure or modify a service, you can:

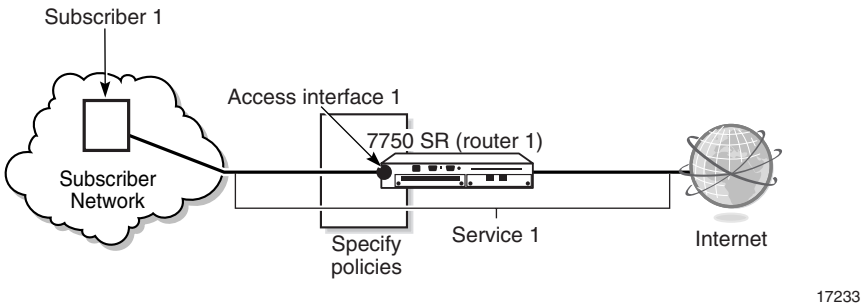
- add access interfaces
- configure or modify existing access interfaces

Packets that arrive at the edge 7750 SR are associated with an IES based on the access interface on which they arrived. An access interface is uniquely identified by the:

- port
- service ID
- IP address

Figure 21-3 shows a sample IES.

Figure 21-3 Sample IES



VPRN

The 5620 SAM supports the creation of VPRN services using managed devices as a PE and P router. VPRNs are also called IP VPNs or BGP/MPLS VPNs. RFC 2547bis defines VPRNs and describes a method for forwarding data and distributing routing information across an IP/MPLS provider core network.

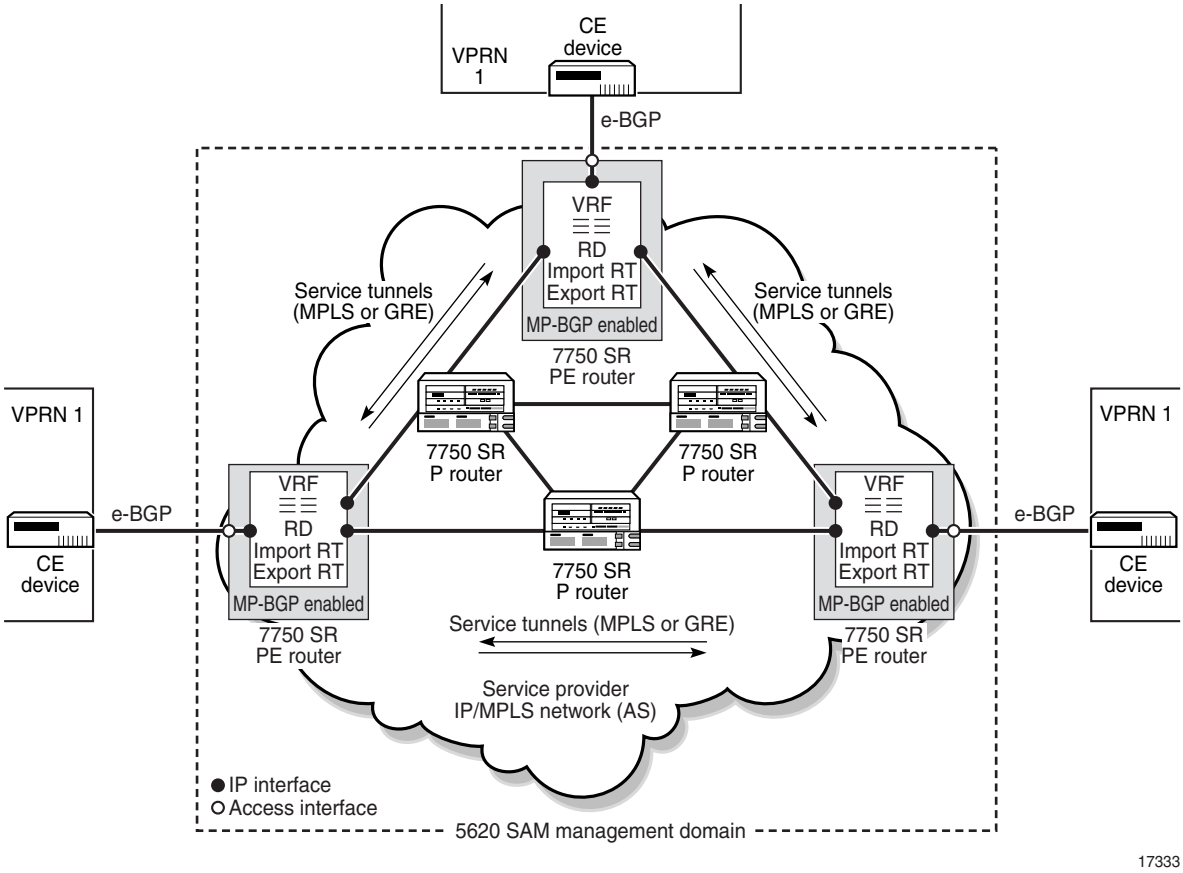
A VPRN service consists of CE routers or devices that are connected to PE routers. PE routers connected to P routers transport data across the IP/MPLS provider core network in service tunnels. The 5620 SAM supports the creation of GRE or MPLS LSP service tunnels.

Packets that arrive at an edge 7750 SR are associated with a VPRN service based on the access interface on which they arrive. An access interface is uniquely identified by the:

- physical Ethernet port or POS port and channel
- encapsulation type
- encapsulation identifier, if required

Figure 21-4 shows a sample VPRN service configuration.

Figure 21-4 Sample VPRN



VLAN

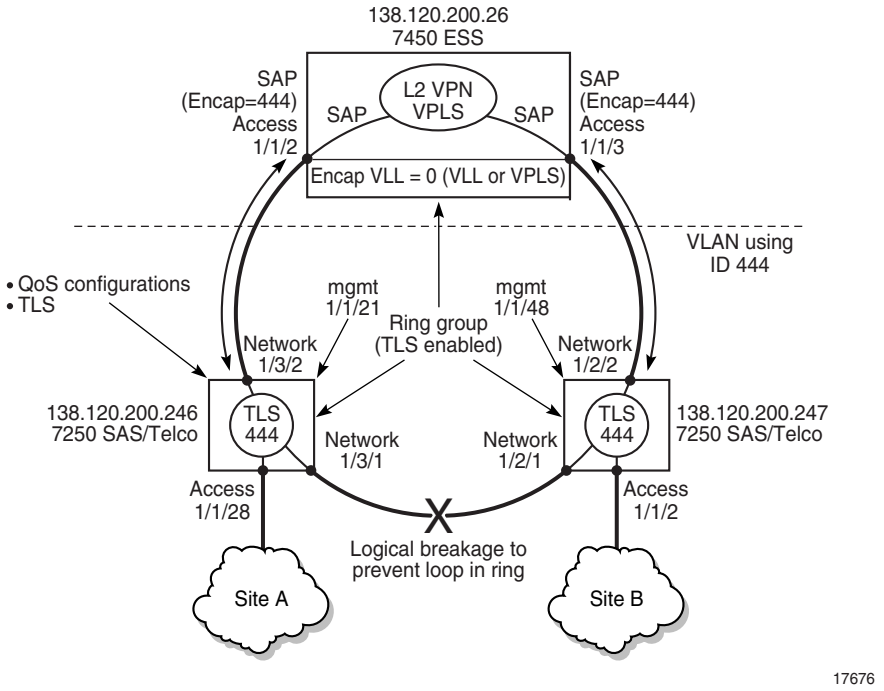
The 5620 SAM supports the creation of VLAN services using Telco devices configured as PE devices. VLANs can be configured as:

- L2 VPN VLAN services
- BTV VLAN services
- Internet access (super-VLAN) services

VLAN ring groups are used to send traffic across an Ethernet ring using copper or fiber optic connections from the source traffic device, for example, from a 7450 ESS to all devices in the ring. Automatic STP configuration on the Telco devices ensures that there is a constant stream of traffic in either direction. Any breaks in the physical links between Telco devices are rerouted.

Figure 21-5 shows a sample VLAN for L2 VPNs.

Figure 21-5 Sample VLAN configuration for L2 VPNs



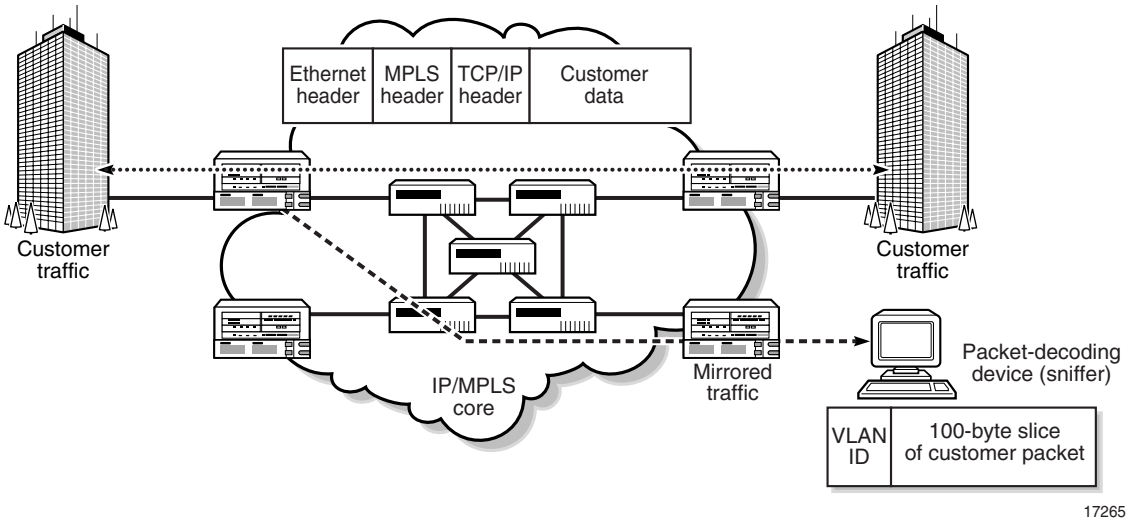
Mirror

The 5620 SAM-O implementation of service mirroring provides mirroring of service traffic packets from any service type.

In a mirror service, originating packets are forwarded to their service destination, and a copy of the entire packet, or a portion of the packet, is sent to the mirror destination. The mirrored packet can be viewed by an operator using a packet sniffer device that is attached to the destination mirror port. A service mirror can have many mirror sources and one destination. Mirroring is performed on a unidirectional basis. The mirrored packets are transported across the core network using IP or MPLS tunneling.

Figure 21-6 shows how service mirroring is performed.

Figure 21-6 Service mirroring



Service mirroring can be used to:

- troubleshoot problems with customer packet delivery and content
- allow service providers to meet regulations by obtaining itemized call records and wiretaps, as authorized by investigative authorities
- simplify the complex analysis networks that are often implemented as overlays to the customer-facing network
- specify full-packet or sliced-packet mirroring, depending on the requirements

Consider the following information before you implement service mirrors.

- The default subscriber is the only subscriber to a service mirror. The default subscriber ID is 1.
- The destination of a service mirror must be a Layer 2 SAP.
- Ingress or egress SAPs, or ingress or egress network ports, can be mirrored. The types of packets that are mirrored differ based on the type of service or port used.
- Service mirror IDs are obtained from the same pool of IDs used by other services. If you manually assign ID values, ensure that you do not assign an ID that belongs to another service.
- Up to 255 mirror destinations are supported per managed network.
- You can use the packet-slicing option to copy a specific packet size for each frame. This option is useful for monitoring network usage without copying the customer data. It also limits the amount of mirrored traffic that streams through the managed devices and the core network.
- If the mirror destination site is not on the same NE as the mirror source, there must be a service tunnel between the source and destination sites.
- The 5620 SAM can automatically create a service tunnel between the source and destination sites in the following case:
  - automatic SDP binding creation is enabled
  - GRE is the transport type
  - no other service tunnel is available between the source and destination sites

- The encapsulation type must be the same for all sites associated with a mirror.
- When the same packet is referenced by multiple mirrors (for example, from a SAP and a port), the packet can only be mirrored once in the following non-configurable order starting with MAC or IP filters:
  - MAC or IP filters
  - MPLS labels (ingress label)
  - SAP
  - port

Caution — Service mirroring can affect performance across the network and in the source and destination devices, and should be planned accordingly.

Optical transport service

An optical transport service is a wavelength that traverses the network between two endpoints, which can be tandem wavelengths in some cases. The path the service takes through the network is defined by the 5620 SAM using shortest path algorithm and the knowledge of the node adjacencies. The path the service takes through an NE is called a cross-connect, or XC. An XC is defined by the ingress and egress points for the service on the NE. The NE physical topology defines the internal path that the service takes through the NE.

The 5620 SAM recognizes the adjoining XCs and managed link XCs as an optical transport service. Each service is assigned a trail identifier and a pair of Wavelength Tracker wave keys. The trail identifier, the ITU channel number (wavelength) and wave key pair are unique in the network. Optical transport services create transport connectivity between router ports and hence they must be created before any IP services.

Services are created by selecting the endpoints, the 5620 SAM is then applied to complete the service creation.

The 5620 SAM supports optical transport services between:

- 1830 PSS
- SR to SR NEs
- alien wavelengths and

Protection schemes:

- diverse route no protection
- protection (Y-cable, OPS, ESNCP), discovery of services
- discovery of Regenerated service

Object life cycle

The object life cycle, or OLC, specifies whether or not excess alarms on a service are displayed in the alarm window. For example, when you configure a service, you can stop the flood of alarms by setting the OLC state of the service to *maintenance* until the service is fully configured. The OLC states are:

- maintenance
- in service

The OLC status can be configured by the user on the following service objects:

- service site
- service
- composite service

You cannot change the OLC state of the following service objects. The OLC state of these objects are affected by the OLC state of the parent objects.

- AccessInterface
  - ServiceAccessPoints
  - L2AccessInterface
  - L3AccessInterface
- SdpBindings
  - MeshSdpBinding
  - MirrorSdpBinding
  - SpokeSdpBinding

If you change the OLC state of an object, the OLC state changes for all objects below it in the containment hierarchy. If an object that is higher in the containment hierarchy is set to a *maintenance* OLC state, you cannot set child objects to an *in service* OLC state. When you change the OLC state of an object, the 5620 SAM-O sends an "AttributeValueChangeEvent" for the object changed. The 5620 SAM-O does not send this event message for the contained objects that are changed as a result of the parent object OLC state change.

21.2 Workflow to configure a VLL, VPLS, IES, VPRN, or VLAN service

Note — The 5620 SAM automatically selects unique values for <id> or <serviceId> elements when your create a service. Do not configure the these elements when you create a service. The request response uses the <objectFullName> element to identify the new service.

1. Configure routing between all the routers. The configuration allows the routers to build routing tables. See chapter 17.
2. Configure the required routing protocols. See chapter 18.
3. Create customers to use the service. See Appendix C for information on how to obtain a sample request to create or modify a customer.

4. Configure the access ports, which are the endpoints of the service. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure an Ethernet port.

5. Configure MPLS if the service is transported across MPLS. See chapter 18.

6. Create service tunnels between all routers that have access ports. The service tunnels direct traffic from one router to another. You must create service tunnels in both directions because service tunnels are unidirectional.

   a. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to create GRE service tunnels.

   b. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to create an MPLS LSP service tunnel.

7. Provision the service.

   i. Specify the subscriber for the service.

   ii. Specify the service type:

      - VLL
      - VPLS
      - IES
      - VPRN

   iii. Specify the transport type:

      - GRE
      - MPLS

   iv. Specify the sites with the configured access interfaces.

   v. Specify the access ports to be used for the service from each of the sites.

      - IES and VLL services require two access ports.
      - VPLS and VPRN services require multiple access ports.

   vi. For VLL and VPLS services, specify which service tunnels are used by configuring the circuits that associate a service and a service tunnel.

   vii. Specify the appropriate policies for the service.

See Appendix C for information on how to obtain the following sample requests:

- sample request to create an IES service
- VLL service:
  - sample request to configure an ATM VLL
  - sample request to configure an Epipe VLL
  - sample request to configure an Fpipe VLL
  - sample request to configure an Ipipe VLL
  - sample request to configure a Cpipe VLL
- sample request to configure a VPLS
- sample request to configure a VPRN
- sample request to configure an access ingress policy

21.3 Workflow to configure a mirror service

1. Specify the general mirror service information.
2. Specify the destination site.
3. Specify the destination Layer 2 access interface. This interface is the sniffer interface.
4. Specify the source site if the source site is different from the destination site.
5. Specify the source Layer 2 or Layer 3 interface. The source interface is the interface that you want to mirror.

   You can also specify whether you want to mirror the source egress stream, the ingress stream, or both.

   See Appendix C for information on how to obtain a sample request to provision a mirror service.

21.4 Workflow to configure an optical transport service

1. Specify the customer for the service.
2. Specify the A end optical service site.
3. Specify the termination point port.
4. Specify the Z end optical service site.
5. Specify the termination point port.

See Appendix C for information on how to obtain a sample request to provision an optical service.

Note 1 — Optical links have to be established between 1830 PSS/SR adjacencies and also the internal topological links with the 1830 PSS NEs in order to get the optical service up and running.

Note 2 — A maximum of two sites can be selected for the creation of an optical transport service. These are respectively named an A end and a Z end site by the 5620 SAM.

21.5 Composite services

The 5620 SAM-O XML interface uses the service.CompositeServiceManager (comp-svc-mgr) package to manage composite services.

Note — Alcatel-Lucent recommends using the generic package for configuring all objects in the service.CompositeServiceManager (comp-svc-mgr) package. See section 16.2 for more information about generic methods. See section 14.2 for more information about the logical object hierarchy.

A composite service is a set of linked services. Composite service functionality supports complex applications that require a combination of services, such as VLAN connections to an HVPLS, an IES spoke into a VPLS, or a VPRN-to-VPLS interconnection.

Services that are owned by different customers can be connected to form a composite service. An example is an HVPLS in which the core VPLS belongs to one customer and the satellite VPLS instances belong to other customers. An HVPLS is considered to be a composite service by the 5620 SAM-O.

Composite services consist of subscriber services, called SCs in the context of a composite service, and connectors. A connector is a bidirectional logical link between two SCs, such as a pair of PW spokes that carry traffic in opposite directions between VLL and VPLS instances, a dot1Q-encapsulated link between a VLAN and a VPLS, or an internal cross-connect.

The term SCP describes a type of connector endpoint. In the case of the services that are available on the 7450 ESS, 7710 SR, 7210 SAS-M, or the 7750 SR, an SCP is a service interface or SAP. For L2 switches, such as the Telco, an SCP may be a network interface, such as an uplink port.

Composite services exist only in the context of the 5620 SAM and are configured through the 5620 SAM GUI or an OSS application. They are unknown to individual network devices. To simplify composite service configuration and to ensure that non-5620 SAM device configuration does not disrupt the management of composite services, the following rules apply to the creation, deletion, modification, and presentation of composite services.

- A composite service can have no SCs.
- A composite service can have zero connectors.
- Two connected SCs can belong to only one composite service.
- A connector between two SCs belongs to only one composite service.

- An SC cannot be removed from a composite service until its connector to the composite service is removed.
- A group of connected services can be moved from one composite service to another.

If a service that is specified for inclusion in a composite service does not currently belong to a composite service, it is added to the composite service regardless of its administrative or operational state. If the specified service is part of an existing composite service, it can be moved to a different composite service. However, SCs that are connected to the specified service are also moved to the new composite service.

Network discovery of composite services

The 5620 SAM-O associates SCs and connectors with composite services during network discovery. The following rules apply to this process.

- When the 5620 SAM-O discovers a valid connector between two services that do not belong to a composite service, a composite service that contains the services and connector is automatically created.
- When the 5620 SAM-O discovers a valid connector between two services and one of the services belongs to a composite service, the other service is added as an SC of the composite service.
- When the 5620 SAM-O discovers a valid connector between two services and the two services are SCs of different composite services, an alarm is raised and the connector is excluded from the 5620 SAM database.

Connector types

There are three types of connectors that join SCs in a composite service:

- SCP-to-SCP
- internal cross-connect
- PW spoke

SCP-to-SCP connectors

SCP-to-SCP connectors can join any two SC types that have service interfaces on the same device or on different devices. A connector between VPLS and VPRN SAPs is an SCP-to-SCP connector, as is a connector between a dot1Q-encapsulated VPLS SAP and L2 switch uplink port of the same encapsulation in a VLAN ring group. The two interfaces must be of the same encapsulation type and encapsulation value. Table 21-1 describes the supported encapsulation types.

Table 21-1 Supported encapsulation types

SAP type	Encapsulation type
Ethernet	Dot1 Q
	Q in Q
	Null
ATM	VPI/VCI
	VPI
FR	DLCI
SONET/SDH	BCP Null
	BCP Dot1 Q
	IPCP
	PPP Auto
	cHDLC
	WAN Mirror
LAG	Null
	Dot1 Q

The operational status of an SCP-to-SCP connector depends on the operational status of its endpoints. An alarm raised against one of the endpoints causes an alarm to be raised against the connector. Such alarms are aggregated within the composite service.

Internal cross-connect connectors

An internal cross-connect connector can join any SC types. It uses a CCAG to join two SCs that have SAPs or network interfaces on the same device. This functionality is available in the 7450 ESS, 7710 SR, and 7750 SR. The following rules apply to internal cross-connect connectors.

- A SAP can be connected to another SAP or to a network interface using a CCAG.
- When a SAP or network interface is deleted, the connector associated with it is also deleted.
- The deletion of an internal cross-connect connector causes the associated interfaces and SAPs to be deleted.

The operational state of an internal cross-connect connector depends on the operational state of the CCAG. An alarm raised against the CCAG causes an alarm to be raised against the connector. Such alarms are aggregated within the composite service.

PW spoke connectors

A PW spoke connector generally joins VPLS instances to create an HVPLS service. A PW spoke can, for example, connect IES and VPLS instances to provide distributed Internet access service. The endpoints of a PW spoke connector must be on different devices. PW spoke connectors are subject to restrictions on the SC types that they can join. Table 21-2 shows the SC types that can be linked by PW spoke connectors.

Table 21-2 Valid PW spoke interconnections

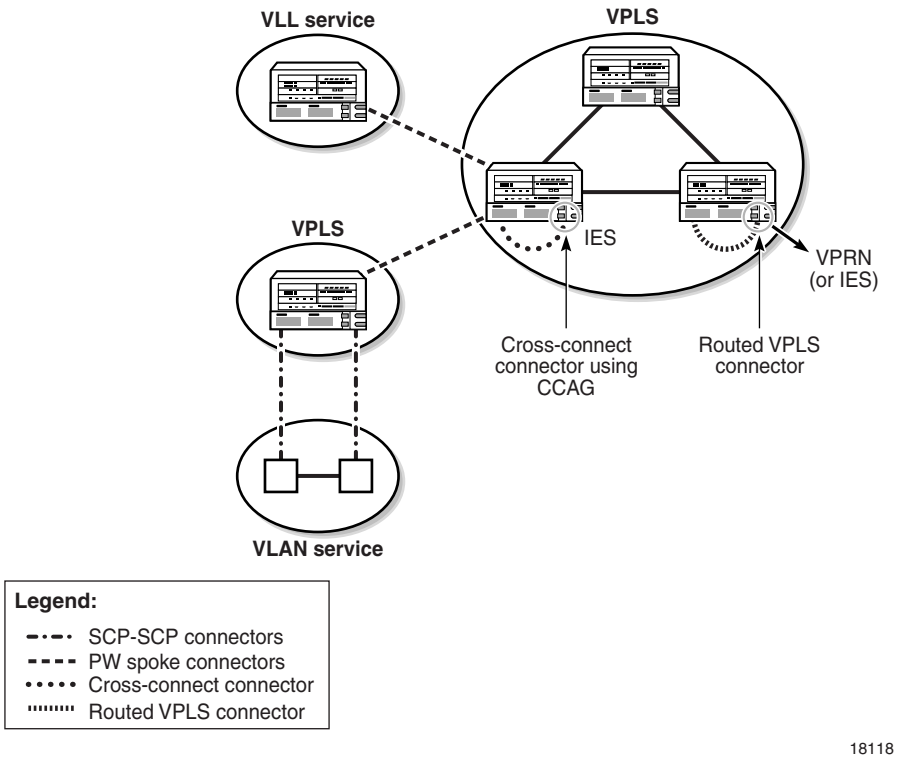
SC type	Valid PW spoke SC interconnections
VLL	IES, VPLS
VLAN	—
VPLS	IES, VLL, VPLS
MVPLS	MVPLS
IES	VLL, VPLS
VPRN	—

The operational state of a PW spoke connector depends on the operational state of the underlying SDP bindings. An alarm raised against one of the SDP bindings causes an alarm to be raised against the connector. Such alarms are aggregated within the composite service.

Sample composite service configuration

Figure 21-7 shows a sample composite service configuration that involves a variety of subscriber services and uses the three SC connector types.

Figure 21-7 Sample composite service configuration



21.6 Workflow to configure a composite service

1. Create the composite service. See Appendix C for information on how to obtain a sample request to create a composite service.
2. Add the service components to the composite service. See Appendix C for information on how to obtain a sample request to add the service components to the composite service.
3. Create a connector between the service components. The 5620 SAM-O supports the following types of connectors:
   - **a** SCP-to-SCP
   - **b** PW spoke
   - **c** Cross connect
     - **i** Create a CCAG on the router. See Appendix C for information on how to obtain a sample request to create a CCAG on the router.
     - **ii** Create the cross connect. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to create a cross connect.

21.7 IGMP snooping

IGMP snooping allows a device to snoop packets sent between IP multicast routers or switches and IP multicast hosts in order to learn the IP multicast group membership. The device checks the IGMP packets for the group registration information, and configures multicasting accordingly.

Without IGMP snooping, multicast traffic is forwarded to all ports, which is the same as broadcast traffic. IGMP snooping ensures that multicast traffic is only forwarded to ports that are members of the specific multicast group, which reduces the amount of multicast traffic passing through the device.

The 7450 ESS and 7750 SR support IGMP snooping for VPLS. A database of group members per VPLS instance is built by listening to IGMP queries and reports from each SAP and SDP of the instance. The reports are forwarded to the multicast routers.

21.8 Workflow to configure IGMP snooping

1. Create a VPLS service. See Appendix C for information on how to obtain a sample request to create a VPLS.
2. Enable IGMP snooping from the site of the VPLS. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to enable IGMP snooping from the VPLS site.
3. Configure IGMP snooping parameters from the site of the VPLS, if required. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure IGMP snooping parameters from the VPLS site.
4. Configure IGMP snooping parameters from the spoke SDP, if required. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure IGMP snooping parameters from the spoke SDP.
5. Configure MFib parameters from the site, if required. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure MFib parameters from the site.

21.9 DHCP relay configuration

DHCP relay interconnects DHCP clients with a DHCP server connected to another LAN segment or network. A DHCP relay agent listens for transmissions from DHCP clients and relays them to the DHCP server. The server responds back to the agent, which then sends the server response back to the client.

5620 SAM supports DHCP relay on VPLS SAPs, VPRN/IES SAPs, spoke/mesh SDP circuits.

21.10 Workflow to configure DHCP relay

1. Configure DHCP Relay on a VPLS SAP.
   - i. Create a VPLS service. See Appendix C for information on how to obtain a sample request to create a VPLS.
   - ii. On the VPLS L2 interface, configure DHCP relay. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure DHCP relay on a VPLS SAP.
2. Configure DHCP Relay on a VPRN SAP.
   - i. Create a VPRN service. See Appendix C for information on how to obtain a sample request to create a VPRN.
   - ii. On the VPRN L3 interface, configure DHCP relay. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure DHCP relay on a VPRN SAP.
3. Configure DHCP Relay on a SDP circuit.
   - i. Create a VPLS service. See Appendix C for information on how to obtain a sample request to create a VPLS.
   - ii. On the VPLS L2 interface SDP circuit, configure DHCP relay. Contact your Alcatel-Lucent OIPS technical support representative for information on how to obtain a sample request to configure DHCP relay on SDP circuit.

21.11 IPsec management

The 5620 SAM IPsec VPRN enables the secure extension of a corporate VPN over uncontrolled or untrusted private and public networks. The ISA-IPSEC MDA on the 7750 SR, Release 6.1 or later, provides IPsec tunneling and encryption between sites.

You can use the 5620 SAM to configure IPsec sessions and the security associations that are required in a bidirectional IPsec tunnel. You can configure multiple IPsec tunnels for each VPRN.

IPsec VPRN services include IPsec tunnels that terminate on IES or VPRN IPsec gateways, and support Layer 3 forwarding through an interface connected to an IPsec tunnel.

Typical IPsec configuration example

The following is a typical IPsec configuration example.

- Individual hosts connect over the Internet into an IES or VPRN service IP interface. The public IP address is local-gateway-address.
- The IES or VPRN service interface defines the public subnet for the secure tunnel and is associated with the ISA by the IPsec group.
- An IPsec interface is configured on a VPRN.
- Tunnel security profiles can be configured per IPsec interface.

See the *5620 SAM User Guide* for more information about IPsec management.

You can use the 5620 SAM-O to configure VPRN services to which individual hosts connect over the Internet on to an IES or VPRN IPsec gateway. You can configure individual or multiple IPsec interfaces for each VPRN service. Multiple tunnel security profiles can be configured for each IPsec interface.

5620 SAM-O IPsec configuration

The 5620 SAM-O XML interface uses the IPsec package to enable secure exchange of packets at the IP layer. This package allows you to create:

- IPsec endpoints on L3 access interfaces on IES and VPRN services
- IPsec interfaces
- IPsec tunnels
- IPsec gateways

You can also use the 5620 SAM-O to create the association between IPsec components, public and private services, to for a secured VPN. When creating secure or delivery services for the IPsec VPN, you need to specify the access interface for the secure service, and an access interface for the delivery service.

OSSI configuration requirements for static tunnel type

For creating static tunnels, the xml script must include parameters of an IPsec tunnel under the IPsec interface.

OSSI requirements for dynamic tunnel type

For creating dynamic tunnels, the xml script must include parameters of an IPsec gateway under the L3 access interface.

The OSSI requires that you specify all of the necessary objects and child objects you need to set up the desired tunnel.

21.12 Workflow to configure IPsec — option A

Contact your Alcatel-Lucent OIPs technical support representative for information on how to obtain a sample for creating the IPsec components in the following workflow.

1. Provision the ISA-IPSEC MDA on the 7750 SR, Release 6.1 or later.
   - i. Create or configure ISA-IPsec groups.
   - ii. Assign the active and backup IPsec group members to the ISA-IPsec groups.
2. Configure security policies.
3. Configure IPsec tunnel templates.

4. Create a VPRN.
   - i. Configure the IPsec security policy.
   - ii. Configure the IPsec security policy entries.
5. Create the private-facing IPsec interface on the VPRN.
   - i. Create a private IPsec SAP.
   - ii. Configure ingress and egress policies.
6. Create the public side of the IPsec service.
   - i. Create an L3 access interface on an IES or VPRN.
   - ii. Define the IPsec public SAP for the L3 access interface.
   - iii. Specify the IPsec gateway, if required.
7. Create IPsec tunnels on the VPRN IPsec interface.
   - i. Create one or more IPsec tunnels.
   - ii. Configure the manual or dynamic keying.
8. Configure the static route.

21.13 Workflow to configure IPsec — option B

Contact your Alcatel-Lucent OIPs technical support representative for information on how to obtain a sample for creating the IPsec components in the following workflow.

1. Create an IPsec VPN.
2. Select the service sites for an IPsec VPN.
3. Create or select the secure service for an IPsec VPN.
4. Create or select the delivery service for an IPsec VPN.
5. Select the IPsec group for an IPsec VPN and fill out the required IP addresses for the tunnel type.
6. Select all required policies and fill out the required information for the tunnel type.

22 — XML API service template configuration

22.1 XML API service template overview *22-2*

22.2 Workflow to manage XML API service templates *22-4*

22.1 XML API service template overview

The 5620 SAM supports the configuration of VLL, VPLS, IES, VLAN, mirror, and VPRN services using XML API script-based templates. The templates allow users to define common characteristics for a service or templatable service object, and the parameter values that can be configured.

Note — Only users that are assigned the mirror service management scope of command role can create, modify, delete, or view mirror services and mirror service templates.

Only users that are assigned the template management scope of command role can create, modify, or delete XML API templates.

See the *5620 SAM User Guide* for more information about user security and scope of command roles

You can create XML API configuration templates for services and child objects of services, and include sites, access interfaces, SDP bindings, group interfaces, SAPs, and other service-related objects. The 5620 SAM generates an XML API script with Velocity properties which you must modify to generate a UI for the configuration form that is tailored to your requirements and to NE-specific properties and attributes.

Note — See chapter 12 for more information about XML API scripts in the 5620 SAM-O. See the *5620 SAM Scripts and Templates Developer Guide* for more information about the script manager and Velocity.

Service templates in the 5620 SAM are based on XML API configuration scripts. XML API service configuration templates simplify service object creation by reducing the number of steps required to create a service or service component. You can use an existing valid service as a starting point for the creation of an XML API service configuration template or create a template without using a managed service object as a basis.

You can create, modify, or delete XML API service configuration templates using the Manage Service Templates form. The form allows you view all templates for services, sites, and templatable children network objects. You can also use the form to convert standard service templates that were created in earlier releases to XML API service templates.

A service template can use secondary, or child, templates to further define the configuration details for the created service. Templates for sites can be bound to the service template. Network objects, such as SDP bindings and interfaces, are considered child templates which can be bound to the site template. Unlike in previous releases, only site templates can be bound to service templates. When you create an XML API configuration template for a service, you can choose to create templates for child objects of the service and to bind the child templates to the service template. You can bind a child template to multiple parent templates. For example, a site template can be bound to several service templates.

Warning — Alcatel-Lucent recommends that you convert old service templates to XML API configuration template only one time. If you convert a template that has already been converted, a new template is generated. However, the association of templated objects — such as services—with the original template is removed and associated with the newly converted template.

Note — You must convert standard service templates that you created using Release 5.0 or earlier of the 5620 SAM to XML API service templates. All the child templates that are bound to the service templates are automatically converted when you convert the service template to an XML API configuration template.

The 5620 SAM user documentation includes a list of templatable objects. To display the list, choose Help→5620 SAM User Documentation in the 5620 SAM main menu, click on the Template Development Information link in the index page that opens, then click on the Classes that can be created from template link.

The association of a service template with service objects created using a 5620 SAM, Release 5.0 or earlier template is maintained after you convert the service template. Because Releases 5.0 and earlier of the 5620 SAM do not support an association between site templates, SDP binding templates, or access interface templates and network objects created from templates, no association is maintained after the conversion.

See the *5620 SAM Scripts and Templates Developer Guide* for more information about how to convert standard service templates to XML API configuration templates.

XML API template script format

Consider the following when you create an XML API configuration template script:

- the generic.GenericObject.configureChildInstance method is the only method that you can use in XML requests
- you must configure only the deployer, distinguishedName, and childConfigInfo parameters of the generic.GenericObject.configureChildInstance method. All other parameters of the are ignored, such as synchronousDeploy, deployRetries and clearOnDeployFailure, and must not be specified.
- you must set the deployer parameter to immediate
- you must set the value of the distinguishedName parameter to the fully distinguished name of the parent object

- use the getObjectFullName() method with the $parent Velocity variable to retrieve the fully distinguished name of the parent object. The Velocity variable $parent is set by the system and can be used in the script whenever the parent context is required
- you must add the value of the childConfigInfo parameter for templatable classes only, such as services, sites, or access interfaces

Code 22-1 shows a sample format of the XML API configuration template script.

Code 22-1: Sample format of the XML API configuration template

```
<xmlapiRequest xmlns="xmlapi_1.0">
<generic.GenericObject.configureChildInstance xmlns="xmlapi_1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <deployer>immediate</deployer>

 <distinguishedName>$parent.getObjectFullName()</distinguishedName>
    <childConfigInfo>
      ## insert the child configuration information here
    </childConfigInfo>
  </generic.GenericObject.configureChildInstance>
</xmlapiRequest>
```

See the 5620 SAM online documentation for sample templates. To display the sample templates, choose Help→5620 SAM User Documentation in the 5620 SAM main menu, click on the Examples of custom templates link in the index page that opens, and then click on the Classes that can be created from template link.

22.2 Workflow to manage XML API service templates

1. Convert templates created with Release 5.0 or earlier of the 5620 SAM to XML API service templates. You can delete old templates, if required. See the *5620 SAM Scripts and Templates Developer Guide* for more information.
2. Create an XML API service template.
3. Create an XML API site template.
4. Create XML API templates for children network objects.
5. Bind an XML API site template to an XML API service template, if required.
6. Bind children network object XML API templates to XML API site templates.

See the 5620 SAM online documentation for sample templates. To display the sample templates, choose Help→5620 SAM User Documentation in the 5620 SAM main menu, click on the Examples of custom templates link in the index page that opens, and then click on the Classes that can be created from template link.

Integrated products

23 — 5650 CPAM integration configuration

23 — 5650 CPAM integration configuration

23.1 5650 CPAM integration overview *23-2*

23.2 5650 CPAM OSS interface overview *23-4*

23.3 5650 CPAM OSS packages *23-11*

23.1 5650 CPAM integration overview

The 5620 SAM can be configured to interwork with the 5650 Control Plane Assurance Manager. The 5650 CPAM provides real-time control IGP topology capture, inspection, visualization, and troubleshooting. The integration with the 5620 SAM allows the 5650 CPAM to cohesively tie routing layer information to infrastructure, such as network routes, service tunnels, LSPs, edge-to-edge service layer connectivity, and rich OAM test management.

Note — You require a 5650 CPAM license key to install the 5650 CPAM.

See the *5650 CPAM User Guide* for more information about the 5650 CPAM and licensing information.

The 5650 CPAM and 5620 SAM integration supports the following operational activities:

- **network planning**
  Planning activities are optimized with real-time topology and strong linkages between services and infrastructure layers in the 5620 SAM GUI and 5620 SAM-O OSS interfaces.
- **network operations**
  Real-time topology and multi-layer highlighting allows you to rapidly assess the state of services, tunnels, and routing on the IGP and IP/MPLS maps.
- **network troubleshooting**
  Historical OAM trace, SPF and RSVP path, and checkpoints allow you to rapidly detect and resolve service level issues whose root cause is in the IP or MPLS layers.
- **network restoration**
  Checkpoints and real-time views of IP/MPLS and service and tunnel infrastructure allow you to restore and plan networks.
- **proactive assurance**
  5650 CPAM alarms, network route and tunnel inspection lists, validation functions, checkpoints, and multi-layer views allow you to detect routing faults.

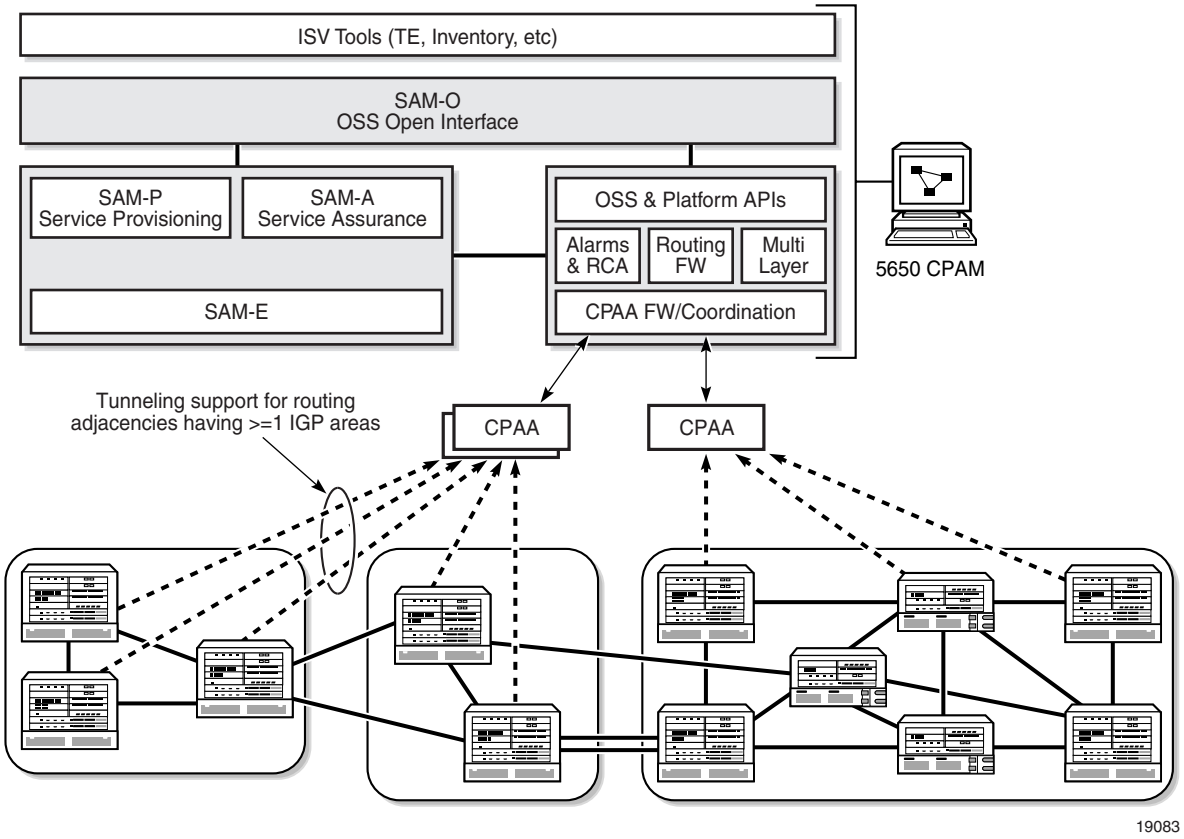
The 5650 CPAM provides a tight eastbound and westbound integration with the 5620 SAM. This integration allows for a real-time view of the network including routing topology and associated configurations whether performed on the GUI, OSS interface, or by CLI. The 5650 CPAM can leverage 5620 SAM redundancy and offer tight navigation between protocol maps and 5620 SAM-managed objects, such as protocol links. In addition, the 5620 SAM GUI supports the management of the 7701 CPAA platform.

When the 5650 CPAM and the 5620 SAM share the same database, the 5650 CPAM can access objects managed by 5620 SAM and display them on the 5650 CPAM topology views. Without integration with the 5620 SAM, the following functions are not available:

- service, LSP, multicast, and OAM highlights
- historical LSP active paths
- LDP and RSVP interface display on the MPLS view

Figure 23-1 shows the 5650 CPAM architecture. See the 5650 CPAM documentation suite for information on how to configure application interoperation.

Figure 23-1 5650 CPAM and 5620 SAM interworking



There are two main components to the control plane assurance management solution:

- a server component, the 5650 CPAM route controller
- a route analyzer component, the 7701 CPAA

The 5650 CPAM server component contains 7701 CPAA control frameworks, applications, and coordination functions for the distributed 7701 CPAAs and provides the applications that are required to leverage the data provided by the 7701 CPAA platform.

The 5650 CPAM route controller, or server, communicates with several GUI and OSSI clients using the same API as the 5620 SAM. When the 5650 CPAM route controller is integrated with a 5620 SAM, the 5620 SAM and 5650 CPAM versions must be compatible. In addition, the 5650 CPAM route controller can run independently with one or multiple 5620 SAM servers.

The 7701 CPAA is a mountable rack that provides an analysis and distributed computing platform. This platform uses a passive version of the industry-tested and -proven 7750 SR routing code base to guarantee a high degree of interoperability with customer networks. The 7701 CPAA acts as a special-purpose routing element whose role is to passively peer with the network.

The 7701 CPAA is supported by the 5620 SAM as a special-purpose NE.

23.2 5650 CPAM OSS interface overview

The 5620 SAM-O OSS interface module includes 5650 CPAM packages and methods that provide a smooth integration path for the export of real-time protocol or IP-level topology to advanced OSS applications, such as TE tools.

The following 5650 CPAM features are of interest to OSS developers:

- multiple topologies
- administrative domains
- checkpoints and real-time topology changes
- route management
- path monitoring
- fault management

Multiple topologies

The 5650 CPAM allows OSS applications to retrieve information on the following network protocol topologies:

- OSPF
- ISIS
- IGP
- BGP
- MPLS — MPLS and LDP information is collected from the 5620 SAM

The 5650 CPAM topology package is used to manage routing topology objects and routing alarms. The 5650 CPAM supports ISIS, OSPF, and BGP.

Both the OSPF and ISIS topologies include information on routers, subnet objects, links, OSPF areas, ISIS levels, and LSDB updates.

The IGP topology includes information on routers, protocols for both OSPF and ISIS, and information on IGP Links.

The MPLS topology includes information on IGP links, MPLS, RSVP, and LDP interfaces for routers.

The BGP topology includes BGP route information originating from a 7701 CPAA that monitors a BGP AS and maintains a BGP RIB. The 7701 CPAA, Release 2.0 and later, supports the following BGP configurations:

- BGP AS
- BGP confederation AS
- BGP sub-AS

5650 CPAM administrative domains

An administrative domain represents a logical routed network. Administrative domains are configured by the operator to reflect the logical structure of the network.

The following types of administrative domains are supported:

- **IGP Administrative Domain**
  Represents an IGP routed network. In an IGP administrative domain, there can be a single OSPF route network and a single ISIS routed network. An IGP Administrative Domain may have both OSPF and ISIS at the same time. For historical reasons, this is modeled as the topology.AutonomousSystem.
- **BGP Autonomous System**
  Represents either a standard BGP AS, a BGP confederation AS, or a BGP sub-AS (confederation member). BGP sub-ASs are configured as children of a confederation BGP AS. Each standard BGP AS (or sub-AS) may be associated with a single IGP administrative domain. The same IGP administrative domain may be assigned to multiple BGP AS (or sub-AS). Note that a confederation BGP AS cannot be assigned to an IGP administration domain. The class is topology.BgpAutonomousSystem.

You can specify the role of a 7701 CPAA, Release 2.0 and later: IGP, BGP, or both. The role of the 7701 CPAA indicates to the 5650 CPAM whether information from a protocol should be interpreted. A 7701 CPAA with an IGP role must be assigned to an IGP administrative domain. A 7701 CPAA with a BGP role must be assigned to a BGP AS.

Note — See the *5650 CPAM User Guide* for more information about the supported topologies.

Topology checkpoints

A topology checkpoint is a snapshot of some part of the network (specified by rules in the subclass) at some specific time. When you create a checkpointed object using a real network object, all of the properties of the real object-for example, metric and bandwidth on IGP links-are copied to the checkpointed object.

After you have set up your network and the network is operational, you can checkpoint the network to create a snapshot of the current state-which can include routers, links, metric configuration, or bandwidth usage-and compare it with checkpoints collected at different times.

All checkpoints exist within an IGP admin domain, this means that you can only create checkpoints on OSPF and ISIS routing protocols.

OSPF checkpoints

You can create OSPF checkpoints for an OSPF area within an IGP administrative domain. Each OSPF area can have multiple checkpoints separated by time.

The following OSPF objects are included in an OSPF checkpoint:

- OSPF area
  - backbone
  - standard
  - stub
  - total stub
  - NSSA no type 5
  - NSSA no summaries
- all OSPF area links (point to point and broadcast)
- all OSPF routers
  - ABR
  - ASBR
  - 5620 SAM-managed
  - third-party managed
- OSPF area subnets
- 7701 CPAA

ISIS checkpoints

You can create ISIS checkpoints in each ISIS routing domain within an IGP administrative domain. Each ISIS routing domain can have multiple checkpoints, separated by time.

The following ISIS objects are included in an ISIS checkpoint:

- ISIS routing domain
  - Level 1—all Level 1 routers that are connected in a single domain within an IGP administrative domain. All Level 1 routers are discovered by one 7701 CPAA.
  - Level 2—all Level 2 routers within an IGP administrative domain
- all ISIS routers
  - L1
  - L2
  - L1/L2
  - ASBR
  - 5620 SAM-managed
  - third-party managed
- ISIS routing domain subnets
- 7701 CPAA

Checkpoints using OSS

When you gather topology information via an OSS, you must use a filter to separate checkpoint data from real-time data.

For real-time:

```
<equal name="isCheckpointObject" value="false"/>
```

For checkpoint:

```
<equal name="isCheckpointObject" value="true"/>
```

You can use the 5650 CPAM to diagnose problems in your network by comparing two checkpoints of the network and viewing the details about the differences in configuration and topology changes.

When you compare two checkpoints, you can specify whether the comparison is of checkpointed areas, routers, links, or subnets, or a combination of these objects.

The first checkpoint that you select is the base for the comparison. You can swap the order of the checkpoints so that the second checkpoint that you select becomes the base for the comparison. You can specify a filter to limit what comparison results are visible.

The objects involved in an OSPF checkpoint comparison include the following:

- OSPF area
- all OSPF area links
- all OSPF routers
- OSPF area subnets

The objects involved in an ISIS checkpoint comparison include the following:

- ISIS routing domain
- all ISIS routing domain links
- all ISIS routers
- ISIS routing domain subnets

Code 23-1 provides an example on how to do checkpoint comparison:

Code 23-1: Checkpoint comparison sample

```
<SOAP:Body>
    <generic.GenericObject.compareObjects xmlns="xmlapi_1.0">
      <comparisonFilter>
        <generic.ComparisonFilter>
          <classesToProcess>
            <!-- Insert classes to compare. Such as: -->
            <!-- <string>topology.OspfLink</string> -->
          </classesToProcess>
          <includeOnlyDiffs>true</includeOnlyDiffs>
        </generic.ComparisonFilter>
      </comparisonFilter>
      <!-- FDN of first checkpoint -->
      <base>tpgy-mgr:name-MyNetwork-AS-0:checkpoint-8</base>
      <!-- FDN of second checkpoint -->
<compareTo>tpgy-mgr:name-MyNetwork-AS-0:checkpoint-2</compareTo>
    </generic.GenericObject.compareObjects>
  </SOAP:Body>
```

Checkpoint schedule policies

You can use the 5650 CPAM to schedule checkpoints by creating checkpoint schedule policies. The 5650 CPAM scheduling functionality allows the creation of 5620 SAM-based schedules for the automatic execution of checkpoint tasks at designated times.

You can associate a schedule that you create using the 5650 CPAM with a checkpoint scheduled task that can be immediately processed, scheduled for later execution, or retained for future use. A scheduled task must be created in the checkpoint scheduler policy configuration form for the scheduled task. Once scheduled tasks are created they are associated with a schedule. A schedule is configurable for one-time or ongoing task execution. You can optionally specify the time at which an ongoing schedule is to stop functioning.

The following are the topology checkpoint classes for scheduling checkpoints and defining schedule policies:

- **CheckpointScheduledTask**
  Represents a specialized Checkpoint Scheduled Task used for performing scheduled checkpoints.
- **CpSchedulePolicy**
  Defines the policy for creating scheduled checkpoints, sub-classes include topology.IsisCpSchedulePolicy and topology.OspfCpSchedulePolicy.

Route management

A managed route is an IP path from a given source router to a given destination router. An IP path represents one or more GRE or LDP tunnels, and loose-path RSVP LSPs, of the same source and destination. Managed routes can span multiple areas and, therefore, multiple 7701 CPAAs.

The route manager object can be used by an OSS application to store multiple managed routes. The OSS is then informed, via the OSSI event channel, of the changes to the paths of the managed routes. A separate route manager must be created for each application. This allows for separate sets of managed routes for different applications. JMS events may be filtered by application.

The OSS application is responsible for creating the route manager using the regular generic.GenericObject.configureChildInstance method using the topology.TopologyManager singleton as the parent.

The route manager and its set of managed routes are persisted. The paths of the managed route are not persisted.

Route management sample workflow

The following workflow provides sample OSS XML requests for performing route management tasks.

1. Create a managed route client, as shown in Code 23-2.
2. Create a request to add or register managed routes, as shown in Code 23-3.
3. Monitor managed routes, as shown in Code 23-4.

Code 23-2: Create managed route client

```
<SOAP:Body>
    <generic.GenericObject.configureChildInstance
xmlns="xmlapi_1.0">
      <deployer>immediate</deployer>
      <distinguishedName>tpgy-mgr</distinguishedName>
      <childConfigInfo>
        <topology.RouteManager>
          <actionMask><bit>create</bit></actionMask>
          <applicationName>TestRoutes</applicationName>
        </topology.RouteManager>
      </childConfigInfo>
    </generic.GenericObject.configureChildInstance>
  </SOAP:Body>
```

Code 23-3: Request to add or register managed routes

```
<SOAP:Body>
    <topology.RouteManager.registerRoutes xmlns="xmlapi_1.0">
      <deployer>immediate</deployer>
      <!-- FDN of Managed Route Client -->
<instanceFullName>tpgy-mgr:application-TestRoutes</instanceFullName>
      <routeKeySet>
        <topology.RouteKey>
          <sourceType>ipv4</sourceType>
          <source>10.1.202.167</source>
          <sourceLen>32</sourceLen>
          <destType>ipv4</destType>
          <dest>10.1.202.10</dest>
          <destLen>32</destLen>
        </topology.RouteKey>
      </routeKeySet>
    </topology.RouteManager.registerRoutes>
  </SOAP:Body>
```

Code 23-4: Monitor managed routes

```
<SOAP:Body>
    <topology.RouteManager.retrieveRoutes xmlns="xmlapi_1.0">
      <!-- FDN of Managed Route Client -->
<instanceFullName>tpgy-mgr:application-TestRoutes</instanceFullName>
      <routeKeySet>
        <topology.RouteKey>
          <sourceType>ipv4</sourceType>
          <source>10.1.202.167</source>
          <sourceLen>32</sourceLen>
          <destType>ipv4</destType>
          <dest>10.1.202.10</dest>
          <destLen>32</destLen>
        </topology.RouteKey>
      </routeKeySet>
    </topology.RouteManager.retrieveRoutes>
  </SOAP:Body>
```

Path monitoring

OSS applications that are integrated with the 5650 CPAM can be used to monitor a limited set of network routes—the route between any two routers seen by the 5650 CPAM. When a network topology changes, such as a link metric or state change, the system evaluates whether the routes of any registered path are affected. If this is the case, new routes are recorded and the OSS clients are immediately informed. If there is no route for a monitored path as a result of a topology change, a record is logged.

The 5650 CPAM supports the following types of path monitoring:

- **IP network path monitoring**
  Monitors the IP forwarding path between two system IP addresses. Both unidirectional and bidirectional IP path monitoring is supported within a single IGP administrative domain. Bidirectional IP paths monitor path divergence and the 5650 CPAM raises a self-clearing alarm when the two paths do not share the same set of IP links.
- **LSP path monitoring**
  Monitors the path of a given LSP. RSVP LSP path monitoring is similar to network route monitoring.

The 5650 CPAM monpath package is used to monitor changes to the paths of IP routes and LSPs. The changes to the path are recorded as they occur by 5650 CPAM IP and LSP path monitors.

The createMonitoredPath method is used to create multiple monitored paths from the passed in LSPs or service tunnels. If a monitored path is not created based on the *fullname* passed into the createMontoredPath method, the LSP or service tunnel is ignored.

You can use the 5650 CPAM to determine the cause event of an IP or LSP path record.

Fault management

The fault management system that the 5650 CPAM-O provides include:

- RCA audit policy for identifying problems or errors in protocol-related and control plane configuration
- threshold reaching alarms generated by the 7701 CPAA

RCA audit policies

The 5650 CPAM allows you to perform on-demand verifications of the IP/MPLS configuration of IS-IS and OSPF routing protocols on 5620 SAM-managed NEs. RCA audit policies identify problems or errors in protocol-related and control plane configurations. These configuration problems, if not discovered in a timely manner, can have significant effects on higher layers, such as MPLS and VPN.

Note — The OSS user must be assigned the Administrator or RCA Verification scope of command role in order to create, modify, and execute all RCA audit policies. The results of an RCA audit are associated with the object, not the user.

You can create RCA audit policies for IGP administrative domains to verify the configuration of network objects on 5620 SAM-managed routers. For each audit policy that you create, you can specify one or more policy entries that define the scope of configuration that is verified by the audit and the severity of the related problem.

For example, an RCA audit policy on an OSPF interface verifies the configuration of several default attributes, such as the configured MTU and Hello interval. A problem with the specified severity is raised against the IGP administration domain for IS-IS misconfigurations, the OSPF area for OSPF misconfigurations, and the IS-IS or OSPF interface for interface misconfigurations, and so on, depending on the policy entry. The results of the verification are displayed in two components trees—the problem tree and the object tree. You can expand the problem tree to view the problem type and other objects affected by the same problem.

The 5650 CPAM rca package is used for policy-driven IP/MPLS configuration audits for OSPF and ISIS routing protocols.

The following workflow provides sample workflow for performing RCA audit policy tasks.

1. Create an RCA audit policy. Configure the attributes that are verified for each RCA audit policy entry.
2. Run the RCA audit policy on a specific object.
3. Identify the problems and correct the configuration.
4. Repeat steps 2 and 3, as required.

Threshold reaching alarms

In addition to routing alarms generated by routers, there are threshold reaching alarms generated by the 7701 CPAA. The alarm generation process uses the routing data collected by the 7701 CPAA. The generated alarms are sent to the 5650 CPAM route controller using a proprietary protocol over the TCP channel. The alarms are then available to 5650 CPAM OSSI clients via JMS.

All 5650 CPAM-related messages are part of the CPAM category, for example ALA_category = CPAM.

The following are not included in the CPAM category:

- Configuration changes
  - interfaces
  - protocol
  - hardware (cards, daughter cards, ports)
  - MPLS
  - LDP

See the *5650 CPAM User Guide* for more information about the 5650 CPAM threshold reaching alarms for BGP, ISIS and OSPF.

23.3 5650 CPAM OSS packages

Table 23-1 describes the packages that include the 5650 CPAM objects.

Table 23-1 5650 CPAM packages

Package	Description	Dependencies
monpath	For configuring monitored paths on the network	—
rca	For 5650 CPAM policy-driven IP/MPLS configuration audits for OSPF and ISIS routing protocols	—
topology	For routing topology object configuration and management, such as: • checkpoints • alarm thresholds • administrative domains • autonomous systems	—
topologysim	For simulated routing topology object configuration and management, such as: • monitored paths • routers	—

Appendices

A. Troubleshooting

B. Accounting and performance

C. 5620 SAM-O SDK library of samples

A. Troubleshooting

A.1 Troubleshooting client OSS application problems *A-2*

A.1 Troubleshooting client OSS application problems

The following procedures describe how to troubleshoot OSS application-specific problems.

Note — See the *5620 SAM Troubleshooting Guide* for more information about how to troubleshoot common GUI and OSS client application issues.

Table A-1 Troubleshooting client OSS application problems procedures

Procedure
The client OSS application cannot communicate with the server
An attempt to log in to the 5620 SAM server fails
Receive insufficient privileges to perform this operation on an object exception when performing an action on a 5620 SAM object
An XML request fails
Unable to perform an action using the 5620 SAM-O
Identifying XML messages from specific users
Receive a java.lang.UnsupportedClassVersionError when sending scripts using the PostXML tool
Receive a java.net.ConnectException when sending scripts using the PostXML tool
Receive a java.net.ConnectException when sending HTTP requests to the 5620 SAM-O server
The OSS client cannot connect with the HTTP or JMS server
The OSS client cannot perform find or findTofile requests
The accounting statistics file has been removed from its location on the 5620 SAM server
The JMS client that collects statistics automatically unsubscribes and deregisters from the 5620 SAM server

Procedure A-1 The client OSS application cannot communicate with the server

The following XML API ping sample can be used to test whether the OSS application can access the 5620 SAM server. Information in the SOAP/XML request includes:

- standard SOAP user and password encoding
- the ping command to look for the release of XML on the server, which in this case, is Release 1.0

An exception response to a failed ping may result in many types of return messages, for example:

- socket timeouts
- HTTP 404 errors
- connection exceptions

1 Run a ping command similar to the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP:Envelope>
    <SOAP:Header>
        <header xmlns="xmlapi_1.0">
            <security>
                <user>user name</user>
                <password>MD5-hashed user password</password>
            </security>
            <requestID>clientName@requestId</requestID>
        </header>
    </SOAP:Header>
    <SOAP:Body>
        <ping xmlns="xmlapi_1.0"/>
    </SOAP:Body>
</SOAP:Envelope>
```

2 Review the three-digit HTTP response code.

- 200 to 299—indicates success, and the problem is not with communication to the server
- 400 to 499—indicates that the error is on the client side, and the request contains bad XML syntax, or cannot be fulfilled
- 500 to 599—indicates that the error is on the server side, and the request is not the problem

3 Check the XML API response for any indication of other communication problems. The following is a successful response message to the ping. The response indicates the correct release 1.0 of XML on the server that responds to the ping.

```
<SOAP:Envelope>
   <SOAP:Header>
      <header xmlns="xmlapi_1.0">
         <requestID>clientName@requestId</requestID>
      </header>
   </SOAP:Header>
   <SOAP:Body>
      <pingResponse xmlns="xmlapi_1.0"/>
   </SOAP:Body>
</SOAP:Envelope>
```

Procedure A-2 An attempt to log in to the 5620 SAM server fails

You receive the following SOAP exception message if you enter an incorrect username or password, if the username or password is missing, or if the password is not hashed:

```
<SOAP:Fault>
<faultcode>SOAP:Client</faultcode>
<faultstring>[security] Login failure.</faultstring>
<faultactor>XmlApi</faultactor>
- <detail>
<requestID>XmlApiClient:0</requestID>
</detail>
```

```
</SOAP:Fault>
```

For more information about MD5-hashed passwords, see chapter 3.

Procedure A-3 Receive insufficient privileges to perform this operation on an object exception when performing an action on a 5620 SAM object

You receive the following exception if you are not assigned the appropriate span of control for an object:

```
[ app: generic ] [ class: GenericObject ] [ instance: instance:object
] [ cause: Method Invocation Failed (13) ] [ descr: Insufficient
privileges to perform this operation. User does not have
create-update-delete access to class classname ][ nested exception:
null ]
```

Check with your administrator to determine whether you are assigned the appropriate span of control to allow you to access the 5620 SAM object.

Procedure A-4 An XML request fails

Check with your administrator to determine whether you have sufficient privileges to execute the request. If the request contains multiple requests, verify each request to isolate the problem.

Procedure A-5 Unable to perform an action using the 5620 SAM-O

Use the 5620 SAM GUI or the script manager to perform the action. See the *5620 SAM User Guide* and *5620 SAM Scripts and Templates Developer Guide* for more information.

Check with your administrator to determine whether you have sufficient privileges to perform the action. You receive the following exception if you are not assigned the appropriate scope of command for the 5620 SAM-O:

```
<?xml version="1.0" ?>
- <SOAP:Envelope
xmlns:SOAP="http://schemas.xmlsoap.org/soap/envelope/">
- <SOAP:Header>
- <header xmlns="xmlapi_1.0">
<requestID>XmlApiClient:2</requestID>
<requestTime>Apr 1, 2008 2:35:46 PM</requestTime>
<responseTime>Apr 1, 2008 2:35:46 PM</responseTime>
</header>
</SOAP:Header>
- <SOAP:Fault>
```

```
<faultcode>SOAP:Client</faultcode>
<faultstring>[security] Users require OSS Management privileges to
use SAM-O</faultstring>
<faultactor>XmlApi</faultactor>
- <detail>
<requestID>XmlApiClient:2</requestID>
</detail>
</SOAP:Fault>
</SOAP:Envelope>
```

Procedure A-6 Identifying XML messages from specific users

Use the following procedure to enable the logging of the 5620 SAM-O XML request, response, and exception messages on the active or standby 5620 SAM servers. The entries in the XML message log files can help you identify the XML requests of a user, multiple users, or of all of the users. See "Log file for XML requests, responses, and exceptions" in chapter 10 for more information about the logging of XML messages.

You must update the nms-sever.xml file on the 5620 SAM server with DEBUG logging to view OSS template information in 5620 SAM server logs.

Caution — The purpose of the XML message log is to debug activities in an OSS application development environment. The message log allows developers to evaluate the interaction of a third-party application with the 5620 SAM-O.

1. Open the nms-server.xml file in the 5620 SAM install directory.
2. Locate the <systemOssiLog> element.
3. Modify the attributes of the <systemOssiLog> element to enable the log option for an individual user or multiple users. In each case, the 5620 SAM creates a unique log file for each HTTP request and response associated with each 5620 SAM-O user. (An HTTP request and response can contain multiple XML requests.) The request log contains the body of the SOAP message. The response log contains the entire SOAP envelope of the response. You can also modify the path for the log file using the attributes associated with the <systemOssiLog> tag.

Caution — Do not modify other nms-server.xml parameters. A modification may seriously affect the network management and performance of the 5620 SAM.

   a. To log the XML messages for all 5620 SAM-O users, enter:

```
<systemOssiLog
        allUsers="yes"
        path="../log/all_users"/>
```

   The above example creates files named ossi<user>Request<n+>.log and ossi<user>Response<n+>.log in the <sam_install_dir>/nms/log/all_users/ directory, where <user> is the multiple 5620 SAM user names and <n+> is the incremental request number.

b To log the XML messages for a single 5620 SAM-O user, enter:

```
<systemOssiLog
        allUsers="no"
        ="yes"
        path="../log/individual"/>
```

The above example creates files named ossi<user>Request<n+>.log and ossi<user>Response<n+>.log in the <sam_install_dir>/nms/log/individual/ directory, where <user> is the 5620 SAM user name and <n+> is the incremental request number.

c To log the XML messages for multiple 5620 SAM-O users, enter:

```
<systemOssiLog
        allUsers="no"
        user_account_for_oss_app1="yes"
        user_account_for_oss_app2="yes"
        path="../log/multiple"/>
```

The above example creates the following files in the <sam_install_dir>/nms/log/multiple/ directory:

- ossiuser_account_for_oss_app1Request<n+>.log
- ossiuser_account_for_oss_app1Response<n+>.log
- ossiuser_account_for_oss_app2Request<n+>.log
- ossiuser_account_for_oss_app2Response<n+>.log

4 To change the log file retention period, add the "timeToKeepLogFile="*minutes*" attribute to the <systemOssiLog> element:

```
<systemOssiLog
      .
      .
      .
     "timeToKeepLogFile="minutes" />
```

where *minutes* is the number of minutes to retain log files

When the timeToKeepLogFile attribute is not entered in the <systemOssiLog> element, the default is 24 h.

5 Close the nms-server.xml file and save the changes.

6 Notify the 5620 SAM server of the changes made to the nms-server.xml file by entering:

```
nmsserver read_config
```

The changes made to the nms-server.xml file are read by the 5620 SAM server after issuing the read_config command or restarting the server.

Procedure A-7 Receive a java.lang.UnsupportedClassVersionError when sending scripts using the PostXML tool

The PostXML tool requires Java version 1.4.2_02 or later. This problem occurs when you install Oracle after installing Java. Verify that the right Java version is in the path. The following error message is an example of the notification that your receive when you run the PostXML batch file:

```
Exception in thread "main" java.lang.UnsupportedClassVersionError:
PostXML (Unsupported major.minor version 48.0)

at java.lang.ClassLoader.defineClass0(Native Method)

at java.lang.ClassLoader.defineClass(ClassLoader.java:495)

at
java.security.SecureClassLoader.defineClass(SecureClassLoader.java:1
10)

at java.net.URLClassLoader.defineClass(URLClassLoader.java:251)

at java.net.URLClassLoader.access$1(URLClassLoader.java:217)

at java.net.URLClassLoader$1.run(URLClassLoader.java:198)

at java.security.AccessController.doPrivileged(Native Method)

at java.net.URLClassLoader.findClass(URLClassLoader.java:192)

at java.lang.ClassLoader.loadClass(ClassLoader.java:300)

at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:290)

at java.lang.ClassLoader.loadClass(ClassLoader.java:256)

at java.lang.ClassLoader.loadClassInternal(ClassLoader.java:316)
```

Procedure A-8 Receive a java.net.ConnectException when sending scripts using the PostXML tool

You must verify that the correct 5620 SAM server is configured in the PostXML batch file. By default, the 5620 SAM server points to localhost. The following error message is an example of the notification that your receive when you run the PostXML batch file:

```
Exception in thread "main" java.net.ConnectException: Connection refused:
connect

at java.net.PlainSocketImpl.socketConnect(Native Method)

at java.net.PlainSocketImpl.doConnect(PlainSocketImpl.java:305)
```

```
at java.net.PlainSocketImpl.connectToAddress(PlainSocketImpl.java:171)

at java.net.PlainSocketImpl.connect(PlainSocketImpl.java:158)

at java.net.Socket.connect(Socket.java:452)

at java.net.Socket.connect(Socket.java:402)

at java.net.Socket.(Socket.java:309)

at java.net.Socket.(Socket.java:124)

at
org.apache.commons.httpclient.protocol.DefaultProtocolSocketFactory.createSo
cket(DefaultProtocolSocketFactory.java:118)

at
org.apache.commons.httpclient.HttpConnection.open(HttpConnection.java:683)

at
org.apache.commons.httpclient.HttpClient.executeMethod(HttpClient.java:662)

at
org.apache.commons.httpclient.HttpClient.executeMethod(HttpClient.java:529)

at PostXML.main(PostXML.java:136)
```

Procedure A-9 Receive a java.net.ConnectException when sending HTTP requests to the 5620 SAM-O server

You receive the following connection refused exception when the 5620 SAM-O server does not respond to HTTP requests:

```
java.net.ConnectException: Connection refused
```

The exception can be caused by the following server conditions:

- server is starting up and not ready to accept connections
- server session limit reached
- host name or IP address in the URL is incorrect
- maximum number of concurrent connections reached
- maximum number of active connections reached on user or user group

The maximum number of JMS connections is 10. The maximum number of concurrent HTTP OSS connections is 10 or more, depending on the server hardware. See the *5620 SAM Planning Guide* for information about platform sizing. If the number of HTTP OSS connections is exceeded, the client application receives an HTTP 503 message for maximum session reached, or an HTTP 903 message for maximum session per user.

Procedure A-10 The OSS client cannot connect with the HTTP or JMS server

Verify the following:

- The 5620 SAM-O client application points to the correct IP address and port of the servers.
- Firewalls between the 5620 SAM-O client and the servers are correctly configured.

1. Open the nms-server.xml file in the 5620 SAM install directory.
2. Verify that the IP addresses are correct.
3. Close the nms-server.xml file and save the changes.
4. Notify the 5620 SAM server of the changes made to the nms-server.xml file by typing:

   ```
   nmsserver read_config
   ```

   The changes to the nms-server.xml file are read by the 5620 SAM server after issuing the read_config command or restarting the server. See the *5620 SAM Planning Guide* for more information about firewall configuration.
5. If you still think that there are connectivity problems, perform the following:
   - Use a standard ICMP ping to verify that you can ping the server from the OSS client station.
   - For XML requests, perform an XML ping using the PostXML utility. See section 5.1 for more information.
   - For JMS problems, try to create the same connection using JMSTest. See Procedure 4-3 for more information.
   - Verify that the required ports are open between the OSS client station and the server. See the *5620 SAM Planning Guide* for firewall configuration information.

Procedure A-11 The OSS client cannot perform find or findTofile requests

If you receive an exception indicating that the number of inventory connections has reached the maximum, the request was rejected. You can retry again later when the number of requests are reduced. This occurs when there are five find or findToFile concurrent requests on the 5620 SAM server.

Procedure A-12 The accounting statistics file has been removed from its location on the 5620 SAM server

If you used the registerLogToFile method to collect accounting statistics, you may be able to remove the statistics file from the 5620 SAM server. This occurs when the timeToKeepFile value, which is configured in the nms-server.xml file, is exceeded. The default is to keep the files is 15 minutes.

Procedure A-13 The JMS client that collects statistics automatically unsubscribes and deregisters from the 5620 SAM server

Every five minutes, the 5620 SAM server verifies that a JMS subscription exists for a client that uses the registerLogToFile method. You can configure the number of times that the 5620 SAM server verifies using the numOfTriesForOfflineClient in the nms-server.xml file. If the value is exceeded, the client is automatically deregistered and unsubscribed from the 5620 SAM server.

B. Accounting and performance

B.1 Description of sample values for performance statistical counters *B-2*

B.1 Description of sample values for performance statistical counters

Refer to the *5620 SAM Statistics Management Guide* for a complete list and description of performance statistical counters. You can use the performance statistical counter information to:

- understand the elements (parameters) that make up each of the statistical counters for all supporting package types
- determine which MIB is the source of the statistical counter
- determine the properties of each statistical counter, for example the counter type and default value

C. 5620 SAM-O SDK library of samples

C.1 Description 5620 SAM-O SDK library of XML samples *C-2*

C.1 Description 5620 SAM-O SDK library of XML samples

The 5620 SAM-O SDK includes a library of XML scripts that provide configuration and network management samples to facilitate OSS integration with the 5620 SAM-O. The SDK consists of a Basic SDK and an Advanced SDK. These samples can be used as a base to build 5620 SAM-O requests.

Warning — Developers should thoroughly test their XML requests before network deployment.

The Basic SDK contains XML scripts that provide examples of how to perform general procedures, such as configure, create, delete, find and filter on 5620 SAM objects. The Advanced SDK provides more comprehensive examples for the entire network management domain that the 5620 SAM supports, such as configuration, fault, inventory, OAM, statistics, and proprietary 5620 SAM methods.

Note — Contact your Alcatel-Lucent OIPS representative for more information about how to obtain the 5620 SAM-O SDK library of sample XML scripts. If OSS developers need assistance developing XML scripts that are not included in the SDK, send a request should be sent to developer.support@alcatel-lucent.com.

Basic SDK

The Basic SDK contains the following XML scripts:

- ConfigureObject.xml
- CreateObject.xml
- DeleteObject.xml
- FilterExample.xml
- Find.xml
- FindToFile.xml
- Ping.xml
- CreateObjectWithResultFilter.xml
- FilterExamples-FilteringBasedOnChildren.xml
- ResultFilterExample-ExcludingChildren.xml
- ResultFilterExample-FilterChildren.xml
- ResultFilterExample-RecursiveResultFilter.xml

Advanced SDK

The Advanced SDK contains a comprehensive set of XML scripts that can be categorized into the following:

- Configuration
- Fault
- Inventory
- Miscellaneous
- OAM
- Server
- Statistics

Configuration

Samples that allow you to perform configuration on the following objects in the 5620 SAM:

- CPAM
- discovery
- equipment
- GNE
- interfaces
- policies
- protocols
- router
- script management
- services
- service tunnels
- shared queuing
- SRLG
- static routes
- subscribers

Fault

Samples that allow you to perform fault management operations, such as ackFaults, FindFaults, and clearFaults.

Inventory

Samples that allow you to perform find or findtoFile operations on objects in the following areas in the 5620 SAM:

- equipment
- policies
- services
- service tunnels
- static routes
- subscribers

Miscellaneous

Samples for specific methods in the following areas in the 5620 SAM:

- CLI
- Deployer

OAM

Samples that allow you to perform OAM tests, such as LspPing and MacPing.

Server

Samples that allow you to perform general server operations on the 5620 SAM server, such as accessing the 5620 SAM release and pinging the server.

Statistics

Samples that allow you to perform statistic operations to collect statistics; for example, registerLogToFile and FindToFile.

Customer documentation and product support

Customer documentation

http://www.alcatel-lucent.com/myaccess

Product manuals and documentation updates are available at alcatel-lucent.com. If you are a new user and require access to this service, please contact your Alcatel-Lucent sales representative.

Technical Support

http://support.alcatel-lucent.com

Documentation feedback

documentation.feedback@alcatel-lucent.com

© 2011 Alcatel-Lucent. All rights reserved.

3HE 06498 AAAB TQZZA Edition 01