Alcatel-Lucent 5620

SERVICE AWARE MANAGER | RELEASE 9.0 R1

3GPP OSS INTERFACE COMPLIANCE STATEMENTS

3HE 06510 AAAA 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

1.1Subject 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.

1.2Customer 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

2.1Customer 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.

2.2All 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

3.1This 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.

8.4The 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.

8.5Alcatel-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.

8.6All 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.

8.7If 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.

8.8Customer 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 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.
5620 SAM optical documentation	

(3 of 4)

Guide	Description
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

5620 SAM 3GPP OSS Interface Compliance Statements

1 — Introduction **1-1**

1.1 Overview 1-2

2 — Interface Behaviour **2-1**

2.1 Transfer Method 2-2

2.2 Available References 2-2

2.3 Notification Service 2-2

3 — IRP Support 3-1

3.1 IRP Overview ... 3-2
3.2 Entry Point IRP ... 3-2
References ... 3-2
General ... 3-2
Operations ... 3-3
getIRPOutline ... 3-3
getIRPReference ... 3-5
3.3 Notification IRP ... 3-7
References ... 3-7
General ... 3-8
Operations ... 3-8
subscribe ... 3-8
unsubscribe ... 3-10
getSubscriptionStatus ... 3-11
getSubscriptionIds ... 3-12
getNotificationCategories ... 3-13
changeSubscriptionFilter ... 3-14
3.4 Communication Surveillance IRP ... 3-15
References ... 3-15
Operations ... 3-16
GetHeartBeatPeriod ... 3-16
Notifications ... 3-16
notifyHeartbeat ... 3-16
3.5 Basic CM IRP ... 3-17
References ... 3-17
General ... 3-18
getMoAttributes and GetContainment ... 3-18
cancelOperation ... 3-20
3.6 Kernel IRP ... 3-21
References ... 3-21
Operations ... 3-21
getNRMIRPVersion ... 3-21
Notifications ... 3-22
notifyObjectCreation ... 3-22
notifyObjectDeletion ... 3-23
notifyAttributeValueChange ... 3-23
3.7 Fault Management IRP ... 3-24
References ... 3-25
Common Information ... 3-25
Handling of Alarm Id ... 3-25
Retrieved Alarms & Alarm Lifecycle ... 3-25
Alarm Comments ... 3-26
Interaction Between Notifications and Alarm Retrieval Requests ... 3-26
AlarmInformation ... 3-26
AlarmIRPOperations ... 3-28
Get Alarm List ... 3-28
AcknowledgeAlarms ... 3-31
getAlarmCount ... 3-32
UnacknowledgeAlarms ... 3-34
SetComment ... 3-34
ClearAlarms ... 3-36

notifyNewAlarm 3-37
notifyAckStateChanged 3-38
notifyClearedAlarm 3-39
notifyChangedAlarm 3-40
NotifyPotentialFaultyAlarmList 3-41
notifyAlarmListRebuilt 3-41
3.8 Generic IRP 3-41
References 3-42
Background 3-42
genericIRPVersionOperations 3-42
getIRPVersion 3-42
getOperationsProfile 3-43
getNotificationsProfile 3-45
3.9 Inventory Management IRP (R3.1 only) 3-47

Appendices

A. Definition of Filter Language A-1

A.1 Overview A-2

B. Use of Object name for Alarm filtering B-1

B.1 Use of Object name for Alarm Filtering B-2

5620 SAM 3GPP OSS Interface Compliance Statements

1 — Introduction

2 — Interface Behaviour

3 — IRP Support

1 — Introduction

1.1 Overview *1-2*

1.1 Overview

This document describes the CNBI R3.0 & R3.1 statements of compliance with 3GPP R7 and R8.

The following IRPs are supported:

- Entry Point IRP
- Notification IRP
- Communications Surveillance IRP
- Basic CM IRP
- Kernel IRP
- Fault Management IRP
- Inventory Management IRP (R3.1 only).

Additionally, the Generic IRP is inherited by the above IRPs. It will support operations to get versions, operations and notifications profile of each supported IRP.

Note — Source for IRP information is taken from the September 2008 3GPP R7 & December 2009 R8 32-series specifications, except for the security IRP which is a proprietary implementation.

2 — Interface Behaviour

2.1 Transfer Method *2-2*

2.2 Available References *2-2*

2.3 Notification Service *2-2*

2.1 Transfer Method

The CORBA transfer method is used for this interface. The application level protocol used over the interface between the agent and manager is CORBA-IIOP.

2.2 Available References

CNBI R3.0 will support 3GPP CORBA R7 or R8, but not both simultaneously. It will be configurable to work in either R7 or R8 mode. The system shall be re-started to switch from one mode to the other.

At start-up, CNBI R3.0 will generate one single EP IRP reference that will provide access to either R7 or R8 IRPs depending on the configured mode.

2.3 Notification Service

The system will provide its own notification service which will be started and stopped with CNBI.

The system will support notifications using the attach_push model.

3 — IRP Support

3.1 IRP Overview *3-2*

3.2 Entry Point IRP *3-2*

3.3 Notification IRP *3-7*

3.4 Communication Surveillance IRP *3-15*

3.5 Basic CM IRP *3-17*

3.6 Kernel IRP *3-21*

3.7 Fault Management IRP *3-24*

3.8 Generic IRP *3-41*

3.9 Inventory Management IRP (R3.1 only) *3-47*

3.1 IRP Overview

The system will support a single IRPAgent which supports the following IRPs – for full details of the methods supported and any limitations see this chapter for more details.

For the methods marked below as not supported, the system will raise an exception:

- OperationNotSupported if declared in the interface (for methods with O qualifier)
- <method_name>Exception in other cases.

Methods in not supported IRPs will raise a generic CORBA exception (as CNBI will not include the IDL for them).

3.2 Entry Point IRP

The following interfaces are defined:

Table 3-1 Entry Point Interfaces

Interface Name	Method name	Support Qualifier	3GPP R7	3GPP R8
EPIRPOperations	getIRPOutline	M	Y	Y
	getIRPReference	M	Y	Y
	releaseIRPReference	M	N	N
EPIRPNotifications	notifyIRPInfoChanges	M	N	N

References

Entry Point (EP) Integration Reference Point (IRP)

Table 3-2 Entry Point references

TS 32.361	Requirements
TS 32.362	Information Service (IS)
TS 32.363	Common Object Request Broker Architecture (CORBA) Solution Set (SS)

General

Supported IRP Version Information

For reference, the table below shows the IRPs and version information which can be returned or used by the notifications/operations in this section.

Table 3-3 Supported IRP versions

iRPId (R7)	rDN (R8)	iRPVersionSet_R7	iRPVersionSet_R8
EPIRP	EPIRP = 1	{ “32.363 V7.0” }	{ “32.363 V8.2” }
NotificationIRP	NotificationIRP = 1	{ “32.303 V7.0” }	{ “32.303 V8.0” }
CSIRP	CSIRP = 1	{ “32.353 V7.0” }	{ “32.353 V8.0” }
BasicCMIRP	BasicCMIRP = 1	{ “32.603 V7.0” }	{ “32.603 V8.1” }
KernelIRP	KernelIRP = 1	{ “32.663 V7.0” }	{ “32.663 V8.1” }
AlarmIRP	AlarmIRP = 1	{ “32.111-3 V7.1” }	{ “32.111-3 V8.0” }

Operations

getIRPOutline

Definition

This is used to access information on the supported IRPs.

Note — The system will only return IRP information for its own IRPAgent.

```
EPIRPConstDefs::Result get_irp_outline(
   in ManagedGenericIRPConstDefs::VersionNumber irp_version,
   out EPIRPConstDefs::SupportedIRPList supported_irp_list
)
raises (GetIRPOutline, InvalidIRPVersion);
```

R7 Parameters

Table 3-4 R7 parameters

IS Name	Qualifier	CORBA SS Details	Purpose
iRPVersion	Input	ManagedGenericIRPConstDefs::VersionNumber	An IRP version to use as a filter for the output - see Table 3-3 for a list of the supported IRP versions If this field is empty, no filtering will be performed and all data is returned.

(1 of 2)

IS Name	Qualifier	CORBA SS Details	Purpose

supportedIRPList	Output	A
		`DN system_dn;`
		and a sequence of
		`struct IRPElement`
		`{`
		`IRPId irp_id;`
		`ManagedGenericIRPConstDefs::`
		`VersionNumberSet irp_versions;`
		`IRPManagementScopeOpt irp_management_scope;`
		`};`
			Returns:
			• the systemDN of the local agent
			• the list of iRPId and iRPVersionSet as shown in Table 3-3
			The iRPManagementScope will always be empty.

Status	Output	`EPIRPConstDefs::Result`	

(2 of 2)

R8 Parameters

Table 3-5 R8 parameters

IS Name	Qualifier	CORBA SS Details	Purpose
iRPVersion	Input	ManagedGenericIRPConstDefs::VersionNumber	An IRP version to use as a filter for the output. See Table 3-3 for a list of the supported IRP versions. If this field is empty, no filtering will be performed and all data is returned.

(1 of 2)

IS Name	Qualifier	CORBA SS Details	Purpose

supportedIRPList	Output	A
		`DN system_dn;`
		And a sequence of
		`struct IRPElement`
		`{`
		`RDN r_DN;`
		`ManagedGenericIRPConstDefs::`
		`VersionNumberSet irp_versions;`
		`IRPManagementScopeOpt irp_management_scope;`
		`};`
			Returns
			• the systemDN of the local agent
			• the list of r_DN and iRPVersionSet as shown in Table 3-3
			The iRPManagementScope will always be empty.

Status	Output	`EPIRPConstDefs::Result`	

(2 of 2)

Exceptions

Table 3-6 getIRPOutline exceptions

Exception Name	Purpose
InvalidIRPVersion	The supplied IRP version is not supported by this agent (that is, is not in the list in Table 3-3).
GetIRPOutline	Generic software error

getIRPReference

R7 Definition

This operation is used to request an IRP reference.

```
EPIRPConstDefs::Result get_irp_reference(
   in EPIRPConstDefs::ManagerIdentifier manager_identifier,
   in EPIRPConstDefs::DN system_dn,
   in EPIRPConstDefs::IRPId irp_id,
   out string irp_reference
)
```

```
raises (GetIRPReference,
        ManagedGenericIRPSystem::InvalidParameter);
```

R8 Definition

This operation is used to request an IRP reference.

```
EPIRPConstDefs::Result get_irp_reference(
   in EPIRPConstDefs::ManagerIdentifier manager_identifier,
   in EPIRPConstDefs::DN system_dn,
   in EPIRPConstDefs:: RDN r_DN,
   out string irp_reference
)
raises (GetIRPReference,
        ManagedGenericIRPSystem::InvalidParameter);
```

R7 Parameters

Table 3-7 R7 parameters

IS Name	Qualifier	CORBA SS Details	Purpose
managerIdentifier	Input	ManagerIdentifier manager_identifier	An identifier for the manager
systemDN	Input	DN system_dn	The value of the systenDN system parameter
iRPId	Input	IRPId irp_id	The IRP Id – from the set shown in Table 3-3
iRPReference	Output	String irp_reference	A reference to the requested IRP.
Status	Output	EPIRPConstDefs::Result	

R8 Parameters

Table 3-8 R8 parameters

IS Name	Qualifier	CORBA SS Details	Purpose
managerIdentifier	Input	ManagerIdentifier manager_identifier	An identifier for the manager
systemDN	Input	DN system_dn	The value of the systenDN system parameter

(1 of 2)

IS Name	Qualifier	CORBA SS Details	Purpose
R_DN	Input	RDN r_DN	The IRP RDN from the set shown in Table 3-3
iRPReference	Output	String irp_reference	A reference to the requested IRP.
Status	Output	EPIRPConstDefs::Result	

(2 of 2)

Exceptions

Table 3-9 getIRPReference exceptions

Exception Name	Purpose
InvalidParameter	The supplied IRP version is not supported by this agent (from the set shown in Table 3-3)
GetIRPReference	Generic software error

3.3 Notification IRP

The following interfaces are defined:

Table 3-10 Notification Interface

Interface Name	Method name	Support Qualifier	3GPP R7	3GPP R8
notificationIRPManagement	Subscribe	M	Y	Y
	Unsubscribe	M	Y	Y
subscriberManagement	getSubscriptionIds	O	Y	Y
subscriptionStatusOperations	getSubscriptionStatus	O	Y	Y
subscriptionFilterOperations	changeSubscriptionFilter	O	Y	Y
subscriptionSuspendOperations	suspendSubscription	O	N	N
	resumeSubscription	O	N	N
IRPManagementOperations	getNotificationCategories	O	Y	Y
NotificationIRPNotification	Push_structured_events	M	Y	Y

References

Notification Integration Reference Point (IRP)

Table 3-11 NIRP References

NIRP Reference	Description
TS 32.301	Requirements
TS 32.302	Information Service (IS)
TS 32.303	Common Object Request Broker Architecture (CORBA) Solution Set (SS)

General

Notification Id

The system will use the following rules for the generation of the notificationId field:

- For alarm notifications and alarm data retrievals: the field is not supported.
- For other notifications: the value is not supported and fixed at 0.

Subscription Removal and Time Tick Support

The system will support the following range of values for the time tick attribute:

- 0: infinite (does not time out)
- 15-1440: time in minutes between which the manager must perform a GetSubscriptionStatus operation.

If the manager does not perform a GetSubscriptionStatus operation in the time internal the subscription will be automatically removed.

Suspend and Resume

The operations to suspend and resume a subscription are not supported.

Operations

subscribe

Definition

This operation is used to create a subscription to receive notifications. If additional requests are made when a subscription is already present, the notification categories will be added to those already supported for this subscription.

```
NotificationIRPConstDefs::SubscriptionId attach_push (

   in string manager_reference,

   in ManagedGenericIRPConstDefs::UnsignedLongOpt time_tick,
```

```
            in NotificationIRPConstDefs::NotificationCategorySetOpt

               notification_categories,

            in ManagedGenericIRPConstDefs::StringOpt filter

        )

        raises (Attach,
ManagedGenericIRPSystem::ParameterNotSupported,

                ManagedGenericIRPSystem::InvalidParameter,
AlreadySubscribed,

                AtLeastOneNotificationCategoryNotSupported);
```

Parameters

Table 3-12 subscribe parameters

IS Name	Qualifier	CORBA SS Details	Purpose
managerReference	Input	string manager_reference	Identifier for the manager
timeTick	Input	ManagedGenericIRPConstDefs::UnsignedLongOpt time_tick	See Subscription Removal and Time Tick Support for details of the supported values.
notification Categories	Input	A flag to indicate whether the category list is provided and a set of strings specifying the categories	See Table 3-13 for a list of supported notification categories An empty value means all notification categories
Filter	Input	ManagedGenericIRPConstDefs::StringOpt filter	See Filteringfor more details.
subscriptionId	Output	NotificationIRPConstDefs::SubscriptionId	The subscription id for this subscription

The following notification categories are supported:

Table 3-13 supported categories

Category name	R7 value	R8 value
Alarms	32.111-3 V7.1	32.111-3 V8.0
Kernel	32.663 V7.0	32.663 V8.1
Communications Surveillance	32.353 V7.0	32.353 V8.0

Exceptions

Table 3-14 subscribe exceptions

Exception Name	Purpose
ParameterNotSupported	Not supported.
InvalidParameter	Empty or invalid manager Reference
AlreadySubscribed	The managerReference is already subscribed to the notification category.
AtLeastOneNotificationCategoryNotSupported	One or more of the specified notification categories is already supported
Attach	Generic software error

Filtering

The following attributes will be supported for use in filtering:

- ProbableCause
- PerceivedSeverity
- Alarm type
- Notification Type
- Managed Object Instance (NE level only)

These attributes may be combined by using “AND” and “OR” to make specific filters.

See Appendix A for a description of how these attributes are represented in the filter.

unsubscribe

Definition

The manager can use this operation to cancel a subscription

```
void detach (

   in string manager_reference,

   in NotificationIRPConstDefs::SubscriptionIdOpt
subscription_id

)

raises (DetachException,

        ManagedGenericIRPSystem::ParameterNotSupported,

        ManagedGenericIRPSystem::InvalidParameter);
```

Parameters

Table 3-15 unsubscribe parameters

IS Name	Qualifier	CORBA SS Details	Purpose
managerReference	Input	string manager_reference	Identifier for the manager
subscriptionId	Input	NotificationIRPConstDefs::SubscriptionIdOpt subscription_id	The subscription id. If this value is supplied, only the subscriptions with this id are cancelled.
Status	Output	EPIRPConstDefs::Result	

Exceptions

Table 3-16 unsubscribe exceptions

Exception Name	Purpose
ParameterNotSupported	Not supported
InvalidParameter	The specified managerReference or subscriptionId is invalid
DetachException	Generic software error

getSubscriptionStatus

Definition

The manager uses this operation to find the status of its subscriptions and to inform the agent that the subscription is still required.

Note — A manager must perform this operation within the specified timeTick period otherwise the Agent will remove the subscription.

```
      NotificationIRPConstDefs::NotificationCategorySet
get_subscription_status

      (

         in NotificationIRPConstDefs::SubscriptionId
subscription_id,

         out ManagedGenericIRPConstDefs::StringOpt filter_in_effect,

         out NotificationIRPConstDefs::SubscriptionStateOpt
subscription_state,

         out ManagedGenericIRPConstDefs::UnsignedLongOpt time_tick

      )
```

```
raises (GetSubscriptionStatus,
        ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);
```

Parameters

Table 3-17 getSubscriptionStatus parameters

IS Name	Qualifier	CORBA SS Details	Purpose
subscriptionId	Input	NotificationIRPConstDefs::SubscriptionId subscription_id,	Subscription id to check the status
filterInEffect	Output	ManagedGenericIRPConstDefs::StringOpt filter_in_effect	The filter configuration for the subscription
SubscriptionState	Output	NotificationIRPConstDefs::SubscriptionStateOpt subscription_state	ACTIVE, SUSPENDED and INVALID will be supported
timeTick	Output	ManagedGenericIRPConstDefs::UnsignedLongOpt time_tick	The timeTick value provided with the subscription
notification CategorySet	Output	NotificationIRPConstDefs::NotificationCategorySet	A list of the notification categories supported. See Section 3.3 Operations

Exceptions

Table 3-18 getSubscriptionStatus exceptions

Exception Name	Purpose
OperationNotSupported	Not supported
InvalidParameter	The specified subscriptionId is invalid.
GetSubscriptionStatus	Generic software error

getSubscriptionIds

Definition

The manager uses this operation to get the values of all still valid (not unsubscribed or removed by IRPAgent) subscriptionIds assigned by NotificationIRP as result of previously subscribe operations performed by this IRPManager.

```
NotificationIRPConstDefs::SubscriptionIdSet get_subscription_ids (
        in string manager_reference
    )
    raises (GetSubscriptionIds,
```

```
ManagedGenericIRPSystem::OperationNotSupported,

ManagedGenericIRPSystem::InvalidParameter);
```

Parameters

Table 3-19 getSubscriptionIds parameters

IS Name	Qualifier	CORBA SS Details	Purpose
managerReference	Input	string manager_reference	Reference of IRPManager that requests the set of active subscription identifiers related to it
subscriptionIdSet	Output	NotificationIRPConstDefs::SubscriptionIdSet	The list of valid subscription identifiers for that IRP Manager

Exceptions

Table 3-20 getSubscriptionIds exceptions

Exception Name	Purpose
GetSubscriptionIds	Supported
OperationNotSupported	Not supported
InvalidParameter	The specified managerReference is invalid.

getNotificationCategories

Definition

The IRPManager invokes this operation to query the categories of notification supported by IRPAgent.

```
NotificationIRPConstDefs::NotificationCategorySet

   get_notification_categories (

      out NotificationIRPConstDefs::NotificationTypesSetOpt

         notification_type_list

)

raises (GetNotificationCategories,

        ManagedGenericIRPSystem::OperationNotSupported);
```

Parameters

Table 3-21 getNotificationCategories parameters

IS Name	Qualifier	CORBA SS Details	Purpose
NotificationCategoryList	Output	NotificationIRPConstDefs::NotificationCategorySet	The list of supported notification categories
Not present in IS	Output	NotificationIRPConstDefs::NotificationTypesSetOpt	The list of notification types for each category

Exceptions

Table 3-22 getNotificationCategories exceptions

Exception Name	Purpose
GetNotificationCategories	Supported
OperationNotSupported	Not supported

changeSubscriptionFilter

Definition

It allows the IRPManager to replace the current filter with a new one.

```
void change_subscription_filter (

   in NotificationIRPConstDefs::SubscriptionId
subscription_id,

   in string filter

)

raises (ChangeSubscriptionFilter,

        ManagedGenericIRPSystem::OperationNotSupported,

        ManagedGenericIRPSystem::InvalidParameter);
```

Parameters

Table 3-23 changeSubscriptionFilter parameters

IS Name	Qualifier	CORBA SS Details	Purpose
subscriptionId	Input	NotificationIRPConstDefs::SubscriptionId	Subscription identifier returned by subscribe operation
filter	Input	filter	New filter constraint

Exceptions

Table 3-24 changeSubscriptionFilter exceptions

Exception Name	Purpose
ChangeSubscriptionFilter	Supported
OperationNotSupported	Not supported
InvalidParameter	Invalid subscription identifier or filter constraint

3.4 Communication Surveillance IRP

The following interfaces are defined:

Table 3-25 Communication surveillance IRP interfaces

Interface Name	Method name	Support Qualifier	3GPP R7	3GPP R8
CSIRPOperations_1	getHeartbeatPeriod	M	Y	Y
	triggerHeartbeat	M	N	N
CSIRPOperations_2	setHeartbeatPeriod	O	N	N
CSIRPNotifications	notifyHeartbeat	M	Y	Y

References

Communication Surveillance Integration Reference Point (IRP)

Table 3-26 CSIRP references

TS 32.351	Requirements

(1 of 2)

TS 32.352	Information Service (IS)
TS 32.353	Common Object Request Broker Architecture (CORBA) Solution Set (SS)

(2 of 2)

Operations

GetHeartBeatPeriod

Definition

The manager can use this operation to retrieve the heart beat period.

```
CSIRPConstDefs::Result get_heartbeat_period(
   out CSIRPConstDefs::HeartbeatPeriod heartbeat_period
)
raises (GetHeartbeatPeriod);
```

Parameters

Table 3-27 GetHeartBeatPeriod parameters

IS Name	Qualifier	CORBA SS Details	Purpose
heartbeat_period	Output	CSIRPConstDefs::HeartbeatPeriod	The heartbeat period in minutes.

Exceptions

Table 3-28 GetHeartBeatPeriod exceptions

Exception Name	Purpose
GetHeartbeatPeriod	Generic software error

Notifications

notifyHeartbeat

Definition

The agent supports a configurable heartbeat timer via a system parameter – a notifyHeartbeat notification is sent when the heartbeat count-down timer reaches zero.

Parameters

Table 3-29 notifyHeartbeat parameters

IS Name	Qualifier	CORBA SS Details	Purpose
objectClass	Output	One NV pair of filterable_body_fields	Details of the CS IRP instance
objectInstance	Output	One NV pair of filterable_body_fields	
notificationId	Output	One NV pair of remaining_body	
eventTime	Output	One NV pair of filterable_body_fields	Set to the current time
systemDN	Output	One NV pair of filterable_body_fields	The value of the systenDN system parameter
notificationType	Output	One NV pair of remaining_body	Set to "notifyHeartbeat"
heartbeatPeriod	Output	One NV pair of remaining_body	Set to the heartbeat period value
Locator	Output	One NV pair of remaining_body	Empty
triggerFlag	Output	One NV pair of remaining_body	Empty
managerIdentifier	Output	One NV pair of remaining_body	Empty

Triggering Event

This event is triggered when the count down timer equals 0.

3.5 Basic CM IRP

The following interfaces are defined:

Table 3-30 Basic CM IRP interfaces

Interface Name	Method name	Support Qualifier	3GPP R7	3GPP R8
PassiveCmIRPOperations	getMoAttributes	M	Y	Y
	getContainment	O	Y	Y
	cancelOperation	O	Y	Y
ActiveCmIRPOperations	createMo	O	N	N
	deleteMo	O	N	N
	setMOAttributes	O	N	N

References

Basic CM Reference Point (IRP)

Table 3-31 Basic CMRP references

TS 32.601	Requirements
TS 32.602	Information Service (IS)
TS 32.603	Common Object Request Broker Architecture (CORBA) Solution Set (SS)

General

Basic CM is supported at an NE level only, for example, only NEs and NE level attributes can be retrieved.

The following set of attributes are supported:

- managedElementId
- dnPrefix - settable through a system parameter.
- userLabel
- locationName
- swVersion
- vendorName

Note — Fixed to the string “Alcatel-Lucent”

- userDefinedState

Note — Fixed to an empty string

- managedElementType
- managedBy - set through a system parameter

Note — The behaviour of the operations described here are solution set dependent – so their behaviour will be described in terms of the solution set.

getMoAttributes and GetContainment

Definition

These operations share the same definition in the CORBA solution set and so are described together. They are used to access a list of NE names and attribute values.

```
BasicCmInformationIterator find_managed_objects(

         in BasicCMIRPConstDefs::DN base_object,

         in BasicCMIRPConstDefs::SearchControl search_control,
```

```
        in BasicCMIRPConstDefs::AttributeNameSet
requested_attributes)

      raises (

        FindManagedObjects,

        ManagedGenericIRPSystem::ParameterNotSupported,

        ManagedGenericIRPSystem::InvalidParameter,

        ManagedGenericIRPSystem::ValueNotSupported,

        ManagedGenericIRPSystem::OperationNotSupported,

        UndefinedMOException,

        IllegalDNFormatException,

        UndefinedScopeException,

        IllegalScopeTypeException,

        IllegalScopeLevelException,

        IllegalFilterFormatException,

        FilterComplexityLimit);
```

Parameters

Table 3-32 getMoAttributes and GetContainment parameters

IS Name	Qualifier	CORBA SS Details	Details for getMOAttributes	Details for getContainment
invokeIdentifierIn	Input	N/A	-	
baseObjectInstance	Input	BasicCMIRPConstDefs ::DN base_object	If the value is set to a NE, information for that NE only is returned. If the value is set to the base object value information for the list of NEs will be returned.	If the value is set to the base object value information for the list of NEs will be returned.
scope	Input	Search_control - subfield are:		
Filter	Input	Type	Will be ignored	
		Level	Will be ignored	
		Filter	Will be ignored	
		Contents	NAMES_AND_ATTRIBUTES	NAMES
attributeListIn	Input	requested_attributes	Will be ignored	

(1 of 2)

IS Name	Qualifier	CORBA SS Details	Details for getMOAttributes	Details for getContainment
invokeIdentifierOut	Input	Supported via the returned iterator (BasicCmInformationIterator)	Handled by returning an iterator	
managedObjectClass	Output		The NE Name	
managedObjectInstance	Output			
attributeListOut	Output		The NE's attribute value; see getMoAttributes and GetContainment for a list of the supported attributes.	Always empty
status	Output	N/A	—	

(2 of 2)

Exceptions

Table 3-33 getMoAttributes and GetContainment exceptions

Exception Name	Purpose
ParameterNotSupported	Not supported
InvalidParameter	Invalid baseObjectInstance
ValueNotSupported	Not supported
IllegalDNFormatException	Not supported
UndefinedMOException,	Base object instance does not exist
IllegalDNFormatException	Not supported
UndefinedScopeException	Not supported
IllegalScopeTypeException	Not supported
IllegalScopeLevelException	Not supported
IllegalFilterFormatException	Not supported
FilterComplexityLimit	Not supported
FindManagedObjects	Generic software error

cancelOperation

Definition

This operation is supported at the solution set level by allowing the destroy operation to be performed on the iterator returned from getContainment or getMoAttributes.

Parameters

Not applicable – performed directly on the iterator returned by the previous operation.

Exceptions

Not applicable – the object must exist in order for the destroy operation to be performed

3.6 Kernel IRP

The following interfaces are defined:

Table 3-34 Kernel IRP interfaces

Interface Name	Method name	Support Qualifier	3GPP R7	3GPP R8
KernelCmIRPOperations	getNRMIRPVersion	M	Y	Y
KernelCmIRPNotifications	objectCreation	O	Y	Y
	objectDeletion	O	Y	Y
	objectAttributeValueChange	O	Y	Y
	notifyCMreSyncRequired	O	N	N
	notifyStateChange	O	N	N

References

Kernel Integration Reference Point (IRP)

Table 3-35 KIRP references

TS 32.661	Requirements
TS 32.662	Information Service (IS)
TS 32.663	Common Object Request Broker Architecture (CORBA) Solution Set (SS)

Operations

getNRMIRPVersion

Definition

It returns the list of supported Network Resource IRP versions:

```
void get_nrm_irp_version

      (

         out ManagedGenericIRPConstDefs::VersionNumberSet
version_number_list,
```

```
        out ManagedGenericIRPConstDefs::VersionNumberSet
vse_version_number_list

      )

      raises (GetNRMIRPVersion);
```

Parameters

Table 3-36 getNRMIRPVersion parameters

IS Name	Qualifier	CORBA SS Details	Purpose
versionNumberList	Output	ManagedGenericIRPConstDefs::VersionNumberSet version_number_list	NRM version number supported (3GPP Release dependant).
vSEVersionNumberList	Output	ManagedGenericIRPConstDefs::VersionNumberSet vse_version_number_list	Not supported.
status	Output	N/A	—

Notifications

notifyObjectCreation

Definition

This notification will be used to notify the creation of a managed object. For this release it will only be supported for object creation for NE objects.

Parameters

Table 3-37 notifyObjectCreation parameters

IS Name	Qualifier	Comments
objectClass	Output	Name of the instance which has been created.
objectInstance	Output	Name of the instance which has been created.
notificationId	Output	Fixed to 0
eventTime	Output	Set to current time
systemDN	Output	The value of the systenDN system parameter
notificationType	Output	Set to “x6”
correlatednotifications	Output	Empty
additionalText	Output	Empty
sourceIndicator	Output	Fixed to “MANAGEMENT OPERATION”

(1 of 2)

IS Name	Qualifier	Comments
attributeList	Output	Set to the attribute list for the NE as described in Section 3.5.

(2 of 2)

Triggering Event

Creation of a network element.

notifyObjectDeletion

Definition

This notification will be used to notify the deletion of a managed object. For this release it will only be supported for object deletion for NE objects.

Parameters

Table 3-38 notifyObjectDeletion parameters

IS Name	Qualifier	Comments
objectClass	Output	Name of the instance which has been deleted.
objectInstance	Output	
notificationId	Output	Fixed to 0
eventTime	Output	Set to current time
systemDN	Output	The value of the systenDN system parameter
notificationType	Output	Set to “x7”
correlatednotifications	Output	Empty
additionalText	Output	
sourceIndicator	Output	Fixed to “MANAGEMENT OPERATION”
attributeList	Output	Empty

Triggering Event

Deletion of the network element.

notifyAttributeValueChange

Definition

This notification will be used to notify the change in attribute values of a managed object. For this release it will only be supported for attributes on NE objects.

Input Parameters

Table 3-39 notifyAttributeValueChange parameters

IS Name	Qualifier	Comments
objectClass	Output	Name of the instance which has been deleted.
objectInstance	Output	
notificationId	Output	Fixed to 0
eventTime	Output	Set to current time
systemDN	Output	The value of the systenDN system parameter
notificationType	Output	Set to “x8”
correlatednotifications	Output	Empty
additionalText	Output	
sourceIndicator	Output	Fixed to “MANAGEMENT OPERATION”
attributeList	Output	The set of changed attribute values for the updated NE [(1)].

Note

(1) Only the attributes listed in Section 3.5 will be monitored for changes and the update provided will include only the new value and not the previous value.

Triggering Event

Change in attribute value.

3.7 Fault Management IRP

The following interfaces are defined:

Table 3-40 Fault management interfaces

Interface Name	Method name	Support Qualifier	3GPP R7	3GPP R8
AlarmIRPOperations	acknowledgeAlarms	M	Y	Y
	getAlarmList	M	Y	Y
	getAlarmCount	O	Y	Y
	unacknowledgeAlarms	O	N	N
	setComment	O	Y	Y
	clearAlarms	O	Y	Y
AlarmIRPNotifications	notifyNewAlarm	M	Y	Y

(1 of 2)

Interface Name	Method name	Support Qualifier	3GPP R7	3GPP R8
	notifyAckStateChanged	M	Y	Y
	notifyClearedAlarm	M	Y	Y
	notifyAlarmListRebuilt	M	Y	Y
	notifyChangedAlarm	O	Y	Y
	notifyComments	O	N	N
	notifyPotentialFaultyAlarmList	O	N	N

(2 of 2)

References

Fault Management Reference Point (IRP)

Table 3-41 FMRP references

TS 32.111-1-700	Requirements
TS 32.111-2-710	Information Service (IS)
TS 32.111-3-701	Common Object Request Broker Architecture (CORBA) Solution Set (SS)

Common Information

Handling of Alarm Id

From TS 32.311-3 it is optional for the agent to provide the alarm id in some cases and the manager should generate an alarm id as a concatenation of managedObjectInstance, eventType, probableCause and specificProblem

However, for all alarmId values in this interface, a name/value pair containing the actual alarm id will be provided. The alarmId will be based on the fault management provider's internal alarm identifier for the alarm. For all operations performed, the manager should provide an alarmId previously returned by the agent from a get alarm list operation.

Retrieved Alarms & Alarm Lifecycle

The alarm list will contain all currently active alarms from the fault provider's current alarm list – depending on the fault provider configuration, this may contain:

- All currently active alarms (i.e. AlarmInformation whose perceivedSeverity is not Cleared) and
  all alarms that are Cleared but not yet acknowledged (or)
- All currently active alarms (i.e. AlarmInformation whose perceivedSeverity is not Cleared)

The alarm lifecycle reported by the system will match that provided by the fault provider.

Alarm Comments

The system will support the northbound system creating additional comments on alarms, but not reading the existing set of comments or receiving notifications for new comments.

Interaction Between Notifications and Alarm Retrieval Requests

The system will process notifications independently from any alarm retrieval requests - so if a 3GPP client is performing an alarm list retrieval, it may still receive alarm notifications at the same time.

The 3GPP client should queue any notifications received between the sending of the get all alarms request and the receipt of the final batch of data from the iterator. The 3GPP client should then replay the stored notifications on the data in its alarm database to pick up the missing updates.

AlarmInformation

The following attributes are supported from the alarm model assuming the hosting product 5620 SAM provides acceptable values for the alarm data for the attributes to be mapped. See notifyNewAlarm for more information.

Table 3-42 alarmInformation attributes

Attribute name	Comments
AlarmId	See Handling of Alarm Id for more information.
notificationId	N
alarmRaisedTime	Y
alarmClearedTime	Y
alarmChangedTime	Y
eventType	Y
probableCause	Y
perceivedSeverity	Y
specificProblem	Y
backedUpStatus	N
trendIndication	N
thresholdInfo	N
stateChangedDefinition	N
monitoredAttributes	N
proposedRepairActions	N
additionalText	Y
additionalInformation	N
ackTime	Y
ackUserId	Y
ackSystemId	N
ackState	Y
clearUserId	Y
clearSystemId	N
serviceUser	N
serviceProvider	N
securityAlarmDetector	N

Note — See notifyNewAlarm for a description of which attribute is applicable for the different types of alarms.

AlarmIRPOperations

Get Alarm List

Definition

GetAlarmList returns the set of active (see Retrieved Alarms & Alarm Lifecycle) alarms available in the system.

This operation is only support in synchronous mode – i.e. the data is returned via an iterator and not via notifications.

```
AlarmIRPConstDefs::AlarmInformationSeq get_alarm_list (
   in ManagedGenericIRPConstDefs::StringOpt filter,
   in AlarmIRPConstDefs::DNOpt base_object,
   out boolean flag,
   out AlarmInformationIterator iter
)
raises (GetAlarmList, FilterComplexityLimit,
ManagedGenericIRPSystem::ParameterNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);
```

Parameters

Table 3-43 Get Alarm List parameters

IS Name	Qualifier	CORBA SS Details	Purpose
alarmAckState	Input	ManagedGenericIRPConstDefs::StringOpt	The following fields will be supported for filtering: • probableCause • perceivedSeverity • alarm type • ack state See Appendix A for a description of the supported filter syntax.
Filter	Input		
baseObjectClass/ baseObjectInstance	Input	AlarmIRPConstDefs::DNOpt base_object	An object name as described in Appendix B
-	Output	boolean flag	Always "FALSE"
-	Output	AlarmInformationIterator iter	The iterator controlling the return of the alarms
-	Output	AlarmIRPConstDefs::AlarmInformationSeq	Not used

Output Data

The alarm data will be returned via an iterator. The tables below show the list of attributes returned and how they relate to the information in the original alarm data provided by the hosting product.

Note — Any restrictions in the hosting product for an attribute will limit the data provided across the 3GPP interface in an equivalent way.

Depending on the alarm type, different sets of attributes are reported:

- AlarmType = COMMUNICATIONS_ALARM, ENVIRONMENTAL_ALARM, EQUIPMENT_ALARM, QUALITY_OF_SERVICE_ALARM or PROCESSING_ERROR_ALARM – Non-security alarm
- AlarmType = SECURITY_SERVICE_OR_MECHANISM_VIOLATION, OPERATIONAL_VIOLATION, PHYSICAL_VIOLATION, TIMEDOMAIN_VIOLATION, INTEGRITY_VIOLATION – Security alarm.

Table 3-44 Get Alarm List output data

Name	Rules for Completion based on the received Alarm Data	Present in Alarm Type
notificationType	Data value based on the alarm state: • notifyNewAlarm – the alarm is in the raised state and alarmChangeTime has not been set • notifyChangedAlarm – the alarm is in the raised state but alarmChangeTime has been set • notifyClearedAlarm – the alarm is in the cleared state	Both
alarmType	The reported alarm type.	Both
objectClass, objectInstance	Generated from the reported object of the alarm.	Both
notificationId	not supported	
eventTime	The parameter carries the • alarmRaisedTime in case notificationType carries notifyNewAlarm • alarmClearedTime in case notificationType carries notifyClearedAlarm • alarmChangeTime in case the notificationType carries notifyChangedAlarm	Both
systemDN	The value of the systenDN system parameter	Both
alarmId	As described in	Both
alarmRaisedTime	The alarm raise time	Both
alarmClearedTime	For a cleared alarm, the reported clear or empty if the alarm has not been cleared.	Both
probableCause	The incoming probable cause will be mapped to the defined set provided by the hosting product If the probableCause is not known in the list, it will be reported as “INDETERMINATE”.	Both

(1 of 2)

Name	Rules for Completion based on the received Alarm Data	Present in Alarm Type
perceivedSeverity	The severity of the alarm as reported, modified by any rules in the fault management system.	Both
specificProblem	Copied from the reported alarm nam. .	Both
backedUpStatus	Not supported	Non-security
trendIndication	Not supported	Non-security
thresholdInfo	Not supported	Non-security
stateChangeDefinition	Not supported	Non-security
monitoredAttributes	Not supported	Non-security
proposedRepairActions	Not supported	Non-security
additionalText	The additionalText for the alarm as reported, modified by any rules in the fault management system.	Both
additionalInformation	Not supported	Both
ackTime	The current value of Ack time if the alarm is acknowledged	Both
ackUserId	The current value of Ack user if the alarm is acknowledged	Both
ackSystemId	Not supported	Both
ackState	The alarm acknowledgement state	Both
clearUserId	The clear User id if the alarm has been cleared	Both
clearSystemId	Not supported	Both
backUpObject	Not supported	Both
correlatedNotifications	Not supported	Both
comments	Not supported	-
serviceUser	Not supported	Security
serviceProvider	Not supported	Security
securityAlarmDetector	Not supported	Security

(2 of 2)

Filtering

The following attributes will be supported for use in filtering:

- ProbableCause

Note — Probable Cause value “INDETERMINATE” cannot be used for filtering.

- PerceivedSeverity
- Alarm type
- Ack state

These attributes may be combined by using “AND” and “OR” to make specific filters.

See Appendix A for a description of how these attributes are represented in the filter.

Exceptions

Table 3-45 Get Alarm List exceptions

Exception Name	Purpose
FilterComplexityLimit	Not supported
ParameterNotSupported	Not supported
InvalidParameter	Invalid filter or base object
GetAlarmList	Generic software error

AcknowledgeAlarms

Definition

The IRPManager invokes this operation to acknowledge one or more alarms by specific alarmId.

```
ManagedGenericIRPConstDefs::Signal acknowledge_alarms (

         in AlarmIRPConstDefs::AlarmInformationIdAndSevSeq

            alarm_information_id_and_sev_list,

         in string ack_user_id,

         in ManagedGenericIRPConstDefs::StringOpt ack_system_id,

         out AlarmIRPConstDefs::BadAcknowledgeAlarmInfoSeq

            bad_ack_alarm_info_list

      )

      raises (AcknowledgeAlarms,
ManagedGenericIRPSystem::ParameterNotSupported,

              ManagedGenericIRPSystem::InvalidParameter);
```

The specified alarm will be updated as follows:

- Acknowledgement State set to Acknowledged.
- Acknowledgement User set to a predefined value. The value of the ackUserId will be added as a comment to the alarm instance.
- Acknowledgement System id will be added as a comment to the alarm instance
- Acknowledgement Time of the alarm will be set to the current time.

The system will perform no verification of the provided values for the ackUserId and ackSystemId fields.

Note — Any perceivedSeverity in the request will be ignored and the alarms acknowledged on the basis of matching the alarmId only.

Parameters

Table 3-46 AcknowledgeAlarms parameters

IS Name	Qualifier	CORBA SS Details	Purpose
alarmInformationAndSeverityReferenceList	Input	List of AlarmInformation.alarmId and AlarmInformation.perceivedSeverity	Contains the alarm details to acknowledge. The information in the perceivedSeverity part will be ignored.
AckUserId	Input	AlarmInformation.ackUserId	Added as comment.
ackSystemId	Input	AlarmInformation.ackSystemId	Added as comment.
badAlarm Information ReferenceList	Output	AlarmIRPConstDefs:: BadAcknowledgeAlarmInfoSeq bad_ack_alarm_info_list	The alarms which were not successfully acknolwedged

Exceptions

Table 3-47 AcknowledgeAlarms exceptions

Exception Name	Purpose
ParameterNotSupported	Not supported
InvalidParameter	The alarm id information received is not valid
AcknowledgeAlarms	Generic software error

getAlarmCount

Definition

This operation allows the manager to retrieve a list of alarm counts by severity for alarms in various raise/clear and acknowledgement states.

```
void get_alarm_count (

   in ManagedGenericIRPConstDefs::StringOpt filter,

   out unsigned long critical_count,

   out unsigned long major_count,
```

```
            out unsigned long minor_count,

            out unsigned long warning_count,

            out unsigned long indeterminate_count,

            out unsigned long cleared_count

        )

    raises (GetAlarmCount, FilterComplexityLimit,
            ManagedGenericIRPSystem::OperationNotSupported,

                ManagedGenericIRPSystem::ParameterNotSupported,

                ManagedGenericIRPSystem::InvalidParameter);
```

The applicability of alarms in certain states depends on the fault provider's system configuration, see Retrieved Alarms & Alarm Lifecycle for more information. If the manager selects an acknowledgement state/alarm state combination which is not possible in the fault provider's configuration, the alarm counts will be returned as 0 for each severity and no exception or error will be returned.

Parameters

Table 3-48 getAlarmCount parameters

IS Name	Qualifier	CORBA SS Details	Purpose
Filter	Input	ManagedGenericIRPConstDefs::StringOpt filter	See Appendix A for a description of the supported filter syntax.
alarmAckState	Input		
criticalCount, majorCount, minorCount, warningCount, indeterminateCount, clearedCount	Output	out unsigned long critical_count out unsigned long major_count out unsigned long minor_count out unsigned long warning_count out unsigned long indeterminate_count out unsigned long cleared_count	The alarm count [(1)].
Status	Output	N/A	-

Note

(1) Any alarms which meet the filter criteria and were reported to the fault provider with a severity of INFO or NONE will be included in the "IndeterminateCount" counter.

Exceptions

Table 3-49 getAlarmCount exceptions

Exception Name	Purpose
FilterComplexityLimit	Not supported.

(1 of 2)

Exception Name	Purpose
OperationNotSupported	Not supported
ParameterNotSupported	Not supported
InvalidParameter	Invalid filter.
GetAlarmCount	Generic software error

(2 of 2)

Supported Filter Syntax

Table 3-50 getAlarmCount supported filter syntax

"IS" Values (i.e. client request)	Value of filter
all alarms	Empty string
all active alarms	Empty string
all active and acknowledged alarms	$.filterable_data(n) == 1
all active and unacknowledged	$.filterable_data(n) == 2
all cleared and unacknowledged alarms	
all unacknowledged	

UnacknowledgeAlarms

Not supported

SetComment

Definition

The IRPManager invokes this operation to record a comment in one or more AlarmInformation instances in AlarmList.

```
      ManagedGenericIRPConstDefs::Signal comment_alarms (

         in AlarmIRPConstDefs::AlarmInformationIdSeq
alarm_information_id_list,

         in string comment_user_id,

         in ManagedGenericIRPConstDefs::StringOpt comment_system_id,

         in string comment_text,

         out AlarmIRPConstDefs::BadAlarmInformationIdSeq

            bad_alarm_information_id_list

      )
```

```
      raises (CommentAlarms,
ManagedGenericIRPSystem::OperationNotSupported,

              ManagedGenericIRPSystem::ParameterNotSupported,

              ManagedGenericIRPSystem::InvalidParameter);
```

The specified commentText will be added to the existing list of notes associated with the specified alarms with the User Id for that note set to a predefined value. The input parameters commentUserId and commentSystemId will be appended to the note text. The notes time field will be set to the current time.

The system will perform no verification of the provided values for the commentUserId and commentSystemId fields.

Input Parameters

Table 3-51 SetComment input parameters

IS Name	Qualifier	CORBA SS Details	Purpose
AlarmInformation ReferenceList	Input	AlarmIRPConstDefs::AlarmInformationIdSeq alarm_information_id_list	List of alarm identifiers to which the comment will be applied
commentUserId	Input	string comment_user_id	Appendded to the commentText
commentSystemId	Input	ManagedGenericIRPConstDefs::StringOpt comment_system_id	Appendded to the commentText
commentText	Input	string comment_text	Comment text to apply
badAlarm Information ReferenceList	Output	AlarmIRPConstDefs::BadAlarmInformationIdSeq bad_alarm_information_id_list	
Status	Output	N/A	-

Exceptions

Table 3-52 SetComment exceptions

Exception Name	Purpose
OperationNotSupported	Not supported
ParameterNotSupported	Not supported
InvalidParameter	Used if an unknown alarm id appears in the alarm information parameter
CommentAlarms	Generic software error

ClearAlarms

Definition

The IRPManager invokes this operation to clear one or more AlarmInformation instances in AlarmList.

```
      ManagedGenericIRPConstDefs::Signal clear_alarms (

         in AlarmIRPConstDefs::AlarmInformationIdSeq
alarm_information_id_list,

         in string clear_user_id,

         in ManagedGenericIRPConstDefs::StringOpt clear_system_id,

         out AlarmIRPConstDefs::BadAlarmInformationIdSeq

            bad_alarm_information_id_list

      )

      raises (ClearAlarms,
ManagedGenericIRPSystem::OperationNotSupported,

              ManagedGenericIRPSystem::ParameterNotSupported,

              ManagedGenericIRPSystem::InvalidParameter);
```

This operation will only be available for ADMC alarms.

For alarms which are cleared by this operation, the alarm severity field will be updated to “Cleared”, the ClearTime field will be set to the current time, the ClearedBy field will be set to a predefined value. The clearUserId and clear clearSystemId input parameters will be added as a comment to the alarm instance. The alarm clear user id will be set to a preconfigured user.

The system will perform no verification of the provided values for the clearUserId or clearSystemId field.

Input Parameter

Table 3-53 ClearAlarms input parameters

IS Name	Qualifier	CORBA SS Details	Purpose
AlarmInformation ReferenceList	Input	AlarmIRPConstDefs::AlarmInformationIdSeq alarm_information_id_list	List of alarm identifiers to which the comment will be applied
clearUserId	Input	string clear_user_id	Added as comment.
clearSystemId	Input	ManagedGenericIRPConstDefs::StringOpt clear_system_id	Added as comment.

(1 of 2)

IS Name	Qualifier	CORBA SS Details	Purpose
badAlarmInformation ReferenceList	Output	AlarmIRPConstDefs::BadAlarmInformationId Seq bad_alarm_information_id_list	Alarms for which the clear couldn't be performed
Status	Output	N/A	-

(2 of 2)

Exceptions

Table 3-54 ClearAlarms exceptions

Exception Name	Purpose
OperationNotSupported	Not supported
ParameterNotSupported	Not supported
InvalidParameter	Used if an unknown alarm id appears in the alarm information parameter or if an alarm id refers to an ADAC alarm
ClearAlarms	Generic software error

notifyNewAlarm

Definition

This notification is reported when a new alarm has been added to the alarm list.

Parameters

The table below shows the attributes which will be reported with the alarm and any special behaviour which is different from that described in the attribute description of getAlarmList.

Table 3-55 notifyNewAlarm parameters

Parameter Name	Generation Rules
objectClass	As described in getAlarmList
objectInstance	As described in getAlarmList
notificationId	As described in getAlarmList
eventTime	Set to the deviceTime reported with the alarm.
systemDN	As described in getAlarmList
notificationType	Fixed to "notifyNewAlarm"
probableCause	As described in getAlarmList
perceivedSeverity	As described in getAlarmList
alarmType	As described in getAlarmList.

(1 of 2)

Parameter Name	Generation Rules
specificProblem	As described in getAlarmList.
correlatedNotifications	As described in getAlarmList.
backedUpStatus	As described in getAlarmList.
backUpObject	As described in getAlarmList.
trendIndication	As described in getAlarmList.
thresholdInfo	As described in getAlarmList.
stateChangeDefinition	As described in getAlarmList.
monitoredAttributes	As described in getAlarmList.
proposedRepairActions	As described in getAlarmList.
additionalText	As described in getAlarmList.
additionalInformation	As described in getAlarmList.
alarmId	As described in getAlarmList.
serviceUser	As described in getAlarmList.
serviceProvider	As described in getAlarmList.
SecurityAlarmDetector	As described in getAlarmList.

(2 of 2)

notifyAckStateChanged

Definition

The subscribed IRPManager instances are notified regarding changes in alarm Acknowledgement State between unacknowledged and acknowledged.

Parameters

The table below shows the attributes which will be reported with the alarm and any special behaviour which is different from that described in the attribute description of getAlarmList.

Table 3-56 notifyAckStateChanged parameters

Parameter Name	Comment to generate from faultEventNotification
objectClass	As described in getAlarmList
objectInstance	As described in getAlarmList
notificationId	As described in getAlarmList
eventTime	Set to the acknowledgeTime
systemDN	As described in getAlarmList
notificationType	Set to “notifyAckStateChanged”
probableCause	As described in getAlarmList

(1 of 2)

Parameter Name	Comment to generate from faultEventNotification
perceived Severity	As described in getAlarmList
alarmType	As described in getAlarmList
alarmId	As described in getAlarmList
ackState	The current acknowledgement state of the alarm.
ackUserId	The current value of the acknowledgedBy field
ackSystemId	The current value of the AckSystemId field.

(2 of 2)

Triggering Event

Acknowledgement of an alarm from:

- Hosting Product
- System GUI
- Northbound interface, including the connected northbound interface to which the notification is being sent.

notifyClearedAlarm

Definition

IRPAgent notifies the subscribed IRPManager of alarm clearing.

The set of correlatedNotification is never reported and the clearUserId and clearSystemId are completed whatever the source of the alarm clear event.

Parameters

The table below shows the attributes which will be reported with the alarm and any special behaviour which is different from that described in the attribute description of getAlarmList.

Table 3-57 notifyClearedAlarm parameters

Parameter Name	Comment
objectClass	As described in getAlarmList
objectInstance	As described in getAlarmList
notificationId	As described in getAlarmList
eventTime	Set to the deviceChangedTime
systemDN	As described in getAlarmList
notificationType	Set to “notifyClearedAlarm”
probableCause	As described in getAlarmList
perceivedSeverity	Fixed to Cleared

(1 of 2)

Parameter Name	Comment
alarmType	As described in getAlarmList
correlated Notifications	Not supported
clearUserId	Always forwarded, whatever the source of the clear
clearSystemId	Always forwarded, whatever the source of the clear
alarmId	See NotifyNewAlarm

(2 of 2)

Triggering Event

Clearing of an alarm from:

- Hosting Product
- System GUI
- Northbound interface, including the connected northbound interface to which the notification is being sent.

notifyChangedAlarm

Definition

The subscribed managers are notified of changes to alarm severity.

Parameters

The table below shows the attributes which will be reported with the alarm and any special behaviour which is different from that described in the attribute description of getAlarmList.

Table 3-58 notifyChangedAlarm parameters

Parameter Name	Comment
objectClass	As described in getAlarmList
objectInstance	As described in getAlarmList
notificationId	See NotifyNewAlarm.
eventTime	Set to the alarmChangeTime
systemDN	See NotifyNewAlarm
notificationType	Set to “notifyChangedAlarm”
probableCause	As described in getAlarmList
perceivedSeverity	See NotifyNewAlarm.
alarmType	As described in getAlarmList
alarmId	See NotifyNewAlarm.

Triggering Event

This notification is only triggered by a change in perceivedSeverity attribute value (except to the value "Cleared").

NotifyPotentialFaultyAlarmList

Not supported.

notifyAlarmListRebuilt

Definition

It is sent when the alarm list has been rebuilt or it is again available.

Parameters

The table below shows the notification parameters:

Table 3-59 notifyAlarmListRebuilt parameters

Parameter Name	Comment
objectClass	It carries the class of the instance identified in systemDN parameter or alternatively, the MonitoredEntity.objectClass
objectInstance	It carries the class of the instance identified in systemDN parameter or alternatively, the MonitoredEntity.objectInstance.
notificationId	See NotifyNewAlarm.
eventTime	Set to the time when the alarm list has been rebuilt or becomes available again
systemDN	See NotifyNewAlarm
notificationType	Set to “notifyAlarmListRebuilt”
reason	It carries the reason why the IRPAgent has rebuilt the AlarmList.
alarmListAlignmentRequirement	Not supported

Triggering Condition

CNBI has reestablished the subscription to the reception of alarm notifications from a remote alarm feed after the connection with that feed got broken and alarm notifications may have been lost during the connection unavailability time.

3.8 Generic IRP

Additionally the following operations are supported from the generic IRP and inherited into the other IRPs:

Table 3-60 Generic IRP interface

Interface Name	Method Name	Support Qualifier	3GPP R7	3GPP R8
genericIRPVersionOperations	Get IRP Versions	M	Y	Y
	Get Operations Profile	O	Y	Y
	Get Notification Profile	O	Y	Y

The behavior for these operations will be defined in a common place rather than in the inherited IRPs.

References

Generic Integration Reference Point (IRP)

Table 3-61 Generic IRP references

TS 32.311	Requirements
TS 32.312	Information Service (IS)
TS 32.313	Common Object Request Broker Architecture (CORBA) Solution Set (SS)

Background

The operations defined in this section are inherited into other IRPs (e.g. FM, Kernel).

genericIRPVersionOperations

getIRPVersion

Definition

This operation returns the supported IRP versions for an interface. The values returned will be dependent on the IRP – see below for details.

Parameters

Table 3-62 getIRPVersion parameters

IS Name	Qualifier	Used in…	CORBA SS Details	R7	R8
versionNumberSet	Output	FM IRP (get_alarm_irp_versions)	ManagedGenericIRPConstDefs::VersionNumberSet	32.111-3 V7.1	32.111-3 V8.0
		Kernel IRP (get_kernel_irp_version)	ManagedGenericIRPConstDefs::VersionNumberSet version_number_list	32.663 V7.0	32.663 V8.1
			ManagedGenericIRPConstDefs::VersionNumberSet vse_version_number_list	Empty set	Empty set
		Basic CM IRP (get_basic_cm_irp_version)	ManagedGenericIRPConstDefs::VersionNumberSet	32.603 V7.0	32.603 V8.0
		Entry Point IRP (get_ep_irp_versions)	ManagedGenericIRPConstDefs::VersionNumberSet	32.363 V7.0	32.363 V8.1
		Notifications IRP (get_notification_irp_versions)	ManagedGenericIRPConstDefs::VersionNumberSet	32.303 V7.0	32.303 V8.0
		Communication Surveillance (get_cs_irp_versions)	ManagedGenericIRPConstDefs::VersionNumberSet	32.353 V7.0	32.353 V8.0
Status	Output	-	-	-	-

Exceptions

Table 3-63 getIRPVersion exceptions

Exception Name	Used in...	Purpose
GetAlarmIRPVersions	FM IRP (get_alarm_irp_versions)	Generic software error
GetKernelCMIRPVersionsException	Kernel IRP (get_kernel_irp_version)	Generic software error
GetBasicCmIRPVersion	Basic CM IRP (get_basic_cm_irp_version)	Generic software error
GetEPIRPVersions	Entry Point IRP (get_ep_irp_versions)	Generic software error
GetNotificationIRPVersions	Notifications IRP (get_notification_irp_versions)	Generic software error
GetCSIRPVersions	Communication Surveillance (get_cs_irp_versions)	Generic software error

getOperationsProfile

Definition

This operation returns the supported operations in a specified IRP. The value returned will be dependent on the IRP; see Table 3-64 for details.

Parameters

The iRPVersion parameter is supported and specifies the IRP version number to look up. The version number is supported if it is listed in the description of get IRP version.

Table 3-64 getOperationsProfile parameters

IS Name	Qualifier	Used in…	CORBA SS Details	Contents
iRPVersion	Input	FM IRP (get_alarm_irp_operations_profile)	ManagedGenericIRPConstDefs::VersionNumber alarm_irp_version	IRP Version number
		Kernel IRP (get_kernel_cm_irp operations_profile)	ManagedGenericIRPConstDefs::VersionNumber kernel_cm_irp_version	IRP Version number
		Basic CM IRP (get_basic_cm_operations_profile)	ManagedGenericIRPConstDefs::VersionNumber basic_cm_irp_version	IRP Version number
		Entry Point IRP (get_ep_irp_operations_profile)	ManagedGenericIRPConstDefs::VersionNumber irp_version	IRP Version number
		Notification IRP (get_notification_irp_operations_profile)	ManagedGenericIRPConstDefs::VersionNumber notification_irp_version	IRP Version number
		Communication Surveillance IRP (get_cs_irp_operations_profile)	ManagedGenericIRPConstDefs::VersionNumber irp_version	IRP Version number
operationNameProfile (and) operationParameterProfile	Output	FM IRP (get_alarm_irp_operations_profile)	ManagedGenericIRPConstDefs::MethodList	Supported set of operations and their parameters
		Kernel IRP (get_kernel_cm_irp operations_profile)		Empty list
		Basic CM IRP (get_basic_cm_irp_operation_profile)		Supported set of operations and their parameters
		Entry Point IRP (get_ep_irp_operations_profile)		Supported set of operations and their parameters
		Notification IRP (get_notification_irp_operations_profile)		Supported set of operations and their parameters
		Communication Surveillance IRP (get_cs_irp_operations_profile)		Supported set of operations and their parameters
status	Output	-	-	-

Exceptions

Table 3-65 getOperationsProfile exceptions

Exception Name	Used in...	Purpose
GetAlarmIRPVersions	FM IRP (get_alarm_irp_versions)	Generic software error
GetKernelCMIRPVersionsException	Kernel IRP (get_kernel_irp_version)	Generic software error
GetBasicCmIRPVersion	Basic CM IRP (get_basic_cm_irp_version)	Generic software error
GetEPIRPVersions	Entry Point IRP (get_ep_irp_versions)	Generic software error
GetNotificationIRPVersions	Notifications IRP (get_notification_irp_versions)	Generic software error
GetCSIRPVersions	Communication Surveillance (get_cs_irp_versions)	Generic software error

getNotificationsProfile

Definition

This operation returns the supported notifications in a specified IRP. The value returned will be dependent on the IRP; see Table 3-66 for details.

The value returned will be dependent on the IRP; see Table 3-66 for details for details.

Parameters

The iRPVersion parameter is supported and specifies the IRP version number to look up. The version number is supported if it is listed in the description of get IRP version.

Table 3-66 getNotificationsProfile parameters

IS Name	Qualifier	Used in…	CORBA SS Details	Contents
iRPVersion	Input	FM IRP (get_alarm_irp_notifications_profile)	ManagedGenericIRPConstDefs::VersionNumber alarm_irp_version	IRP Version number
		Kernel IRP (get_kernel_cm_irp notifications _profile)	ManagedGenericIRPConstDefs::VersionNumber kernel_cm_irp_version	IRP Version number
		Basic CM IRP (get_basic_notifications_profile)	ManagedGenericIRPConstDefs::VersionNumber basic_cm_irp_version	IRP Version number
		Entry Point IRP (get_ep_irp_ notifications _profile)	ManagedGenericIRPConstDefs::VersionNumber irp_version	IRP Version number
		Notification IRP (get_notification_irp_ notifications _profile)	ManagedGenericIRPConstDefs::VersionNumber notification_irp_version	IRP Version number
		Communication Surveillance IRP (get_cs_irp_ notifications _profile)	ManagedGenericIRPConstDefs::VersionNumber irp_version	IRP Version number
operationNameProfile (and) operationParameterProfile	Output	FM IRP (get_alarm_irp_ notifications _profile)	ManagedGenericIRPConstDefs::MethodList	Supported set of notifications and their parameters
		Kernel IRP (get_kernel_cm_irp notifications _profile)		Supported set of notifications and their parameters
		Basic CM IRP (get_basic_cm_irp_ notifications _profile)		Empty List
		Entry Point IRP (get_ep_irp_ notifications _profile)		Empty List
		Notification IRP (get_notification_irp_ notifications _profile)		Empty List
		Communication Surveillance IRP (get_cs_irp_ notifications _profile)		Supported set of notifications and their parameters
status	Output	-		-

Exceptions

Table 3-67 getNotificationsProfile exceptions

Exception Name	Used in...	Purpose
GetAlarmIRPNotificationProfile	FM IRP (get_alarm_irp_notification_profile)	Generic software error
InvalidParameter		Supported
OperationNotSupported		Not supported
GetKernelCMIRPNotificationProfileException	Kernel IRP (get_kernel_cm_irp_ notification _profile)	Generic software error
InvalidParameter		Supported
OperationNotSupported		Not supported
GetBasicCmIRPNotificationProfile	Basic CM IRP (get_basic_cm_irp_ notification _profile)	Generic software error
InvalidParameter		Supported
OperationNotSupported		Not supported
GetEPIRPNotificationProfile	Entry Point IRP (get_ep_irp_ notification _profile)	Generic software error
InvalidParameter		Supported
OperationNotSupported		Not supported
GetNotificationIRPNotificationProfile	Notifications IRP (get_notification_irp_ notification _profile)	Generic software error
InvalidParameter		Supported
OperationNotSupported		Not supported
GetCSIRPNotificationProfile	Communication Surveillance (get_cs_irp_ notification _profile)	Generic software error
InvalidParameter		Supported
OperationNotSupported		Not supported

3.9 Inventory Management IRP (R3.1 only)

3GPP does not define any interface for Inventory Management IRP. It does, however, define the InventoryUnit object class (TS 32.692) and the schema (TS 32.695, InventoryNrm.xsd) to which inventory files shall adhere.

CNBI R3.1 supports generation of 3GPP inventory reports compliant with the InventoryNrm.xsd schema for the following InventoryUnit attributes:

- inventoryUnitId
- inventoryUnitType
- vendorUnitFamilyType
- vendorUnitTypeNumber

- vendorName
- serialNumber

The inventoryUnitId will contain the DN of the InventoryUnit MO.

InventoryUnit instances will be grouped by ManagedElement instance.

In the following section it will be detailed each supported IRP interface and parameters. Unless stated otherwise, compliance applies to both R7 and R8.

Appendices

A. Definition of Filter Language ***A-1***

B. Use of Object name for Alarm filtering ***B-1***

A. Definition of Filter Language

A.1 Overview *A-2*

A.1 Overview

Table A-1 below describes how to represent a supported attribute as a filter criteria, the operations in which the filter criteria is supported, and the range of the valid values.

Table A-1 Filter language

Attribute Name	Reference String in Filter	Supported In Operations	Operands Supported	Supported Values
ackState	$.filterable_data(n)	get_alarm_list, get_alarm_count	==	Acknowledged -> 1 Unacknowledged -> 2
perceivedSeverity	$.filterable_data(h)	get_alarm_list, subscribe	==	INDETERMINATE -> 1 CRITICAL -> 2 MAJOR -> 3 MINOR -> 4 WARNING -> 5 CLEARED -> 6
probableCause	$.filterable_data(g)	get_alarm_list, subscribe	==	An integer from the available probable cause set
alarmType	$.header.fixed_header.event_name	get_alarm_list, subscribe	==	COMMUNICATIONS_ALARM -> x1 PROCESSING_ERROR_ALARM -> x2 ENVIRONMENTAL_ALARM -> x3 QUALITY_OF_SERVICE_ALARM -> x4 EQUIPMENT_ALARM -> x5 INTEGRITY_VIOLATION -> x6 OPERATIONAL_VIOLATION -> x7 PHYSICAL_VIOLATION -> x8 SECURITY_SERVICE_OR_MECHANISM_VIOLATION -> x9 TIME_DOMAIN_VIOLATION -> x10
Notification_type	$.header.fixed_header.type_name	subscribe	==	NOTIFY_FM_NEW_ALARM > x1 NOTIFY_FM_CHANGED_ALARM -> x2 NOTIFY_FM_ACK_STATE_CHANGED -> x3 NOTIFY_FM_CLEARED_ALARM -> x5
Object Instance	$filterable_data(e)	subscribe	==	String representing the object name

B. Use of Object name for Alarm filtering

B.1 Use of Object name for Alarm Filtering *B-2*

B.1 Use of Object name for Alarm Filtering

For some operations, the object name field can be used as a basis for alarm filtering. The support is described in Table B-1.

Table B-1 Object names for alarm filtering

Operation	Sample Object Name Format	Meaning
getAlarmList	<empty string>	All alarms
	SubNetwork=X, ManagedElement=Y	All alarms on a given Network Element
GetMoAttributes GetContainment	<empty string>	All NE data
	SubNetwork=X, ManagedElement=Y	Data for a specific NE

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 06510 AAAA TQZZA Edition 01