NOKIA

5529 OSS Alarm Dispatcher, Release 9.6.07

Northbound Interface Guide

3JL-04001-BRAA-PWZZA

Edition 01

August 2018

Table of contents

Getting started

5529 OAD Northbound interface

List of figures

List of tables

List of procedures

1 Preface

The *5529 OAD Northbound Interface Guide* contains information about the following:

- alarm retrieval operations
- alarm and event notifications
- alarm acknowledgment and unacknowledgment
- SNMP trap forwarding
- OSS client applications
- MTOSI objects
- SOAP envelope components

1.1 Related documentation

Table 1 describes the related documentation sources that you may need to reference.

Table 1 **Related documentation**

Customer documentation	Description
5520 Access Management System	
5520 AMS Administrator Guide	Information about administrative functions, including management of client-server communication, users, NE communication, and schedules
Release Notice for the NE support plug-in	NE-specific information about the mapping of alarm IDs to the 5529 OAD specific problem, native probable cause, and MTOSI probable cause attributes
5520 AMS and 5529 Enhanced Applications	
5520 AMS and 5529 Enhanced Applications Alarm Search Tool	5520 AMS, 5529 Enhanced Applications, and NE alarm descriptions
5520 AMS and 5529 Enhanced Applications Privacy Considerations	Information about the product features that impact privacy and the measures taken to protect such data
5520 AMS Solution Glossary	Terms and acronyms related to the 5520 AMS and 5529 Enhanced Applications
5520 AMS Solution Planning Guide	Information about the system requirements for the installation of the 5520 AMS server and client
5529 Enhanced Applications	
5529 Enhanced Applications Release Notice	Information about updates to the product, software and documentation delivery, known restrictions, and fixed issues
5529 OSS Alarm Dispatcher	
5529 OAD Installation and Administration Guide	Information about how to install and administer the 5529 OAD

1.2 Conventions used in this guide

Table 2 describes the conventions that are used in this guide.

Table 2 **Documentation conventions**

Convention	Description	Example
Italics	Identify a variable	*hostname*
Key+Key	Type the appropriate consecutive keystroke sequence.	CTRL+G
Key–Key	Type the appropriate simultaneous keystroke sequence.	CTRL–G
↵	Press the Return key.	↵
—	An em dash in a table cell indicates that there is no information.	—
→	A right arrow graphic following the menu label indicates that a cascading submenu results from selecting a menu item.	File→Save

1.2.1 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 — A note provides information that is, or may be, of special interest.

1.2.2 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.

Procedure 1 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.

Procedure 2 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.

1.3 Multiple PDF file search

You can use Adobe Reader, Release 6.0 or later, to search multiple PDF files for a term. Adobe Reader displays the results in a display panel. The results are grouped by PDF file. You can expand the entry for each file.

Note — The PDF files in which you search must be in the same folder.

Procedure 3 To search multiple PDF files for a term

1. Open the Adobe Reader.

2. Choose Edit→Advanced Search from the Adobe Reader main menu. The Search window opens.

3. Enter the term to search for.

4. Select the All PDF Documents in radio button.

5. Choose the folder in which to search using the drop-down menu.

6. Select the following search criteria, if required:

   - Whole words only
   - Case-Sensitive
   - Include Bookmarks
   - Include Comments

7. Click on the Search button.

   Adobe Reader displays the search results. You can expand the entries for each file by clicking on the + symbol.

Note — After you click on a hyperlink, you can right-click and choose Previous View from the contextual menu to return to the location of the hyperlink that you clicked on.

Getting started

2 What's new

2.1 What's new in Release 9.6.07

Table 3 describes the 5529 OAD features and enhancements added to the *5529 OAD Northbound Interface Guide* for Release 9.6.07.

Table 3 **What's new in Release 9.6.07**

Feature/enhancement	Description	Edition	See
New features and enhancements			
neIpAddress parameter	The 5529 OAD NBI is updated to support NEs that have IPv6 address management. The neIpAddress parameter in the 5529 OAD alarm response and notification messages is updated.	01	Tables 18 and 20
Documentation changes			
5520 AMS and 5529 Enhanced Applications Privacy Considerations	Added the *5520 AMS and 5529 Enhanced Applications Privacy Considerations* document to the list of related documentation sources	01	Table 1
Alarm event notifications	Added a note about how the 5529 OAD NBI user that is subscribed to JMS event notifications receives alarm notifications for all of the NEs that are managed by the 5520 AMS	01	Section 5.2
JMS application parameters	Added the following missing information: • parameters required to establish a connection over SSL between the JMS client application and JMS server • system java properties of the JMS client application that need to be configured correctly	01	Section 5.4.5
Libraries	Removed the following libraries: • jaxen-1.1.1.jar • jaxen-jdom-1.0.jar • jdom-1.1.jar • saxpath-1.0-FCS.jar The libraries are not required by the OSS client application to process the getSystemInfo operation, and they are not provided in the 5529 OAD OSS client .tar file.	01	Section 5.4.6
SOAP version 1.2	Added SOAP version 1.2 to the list of supported technologies and standards that allow the 5529 OAD to easily integrate with OSS client applications. SOAP 1.2 is supported on 5529 OAD R9.6 or later.	01	Section 3.5

2.2 What’s new in Release 9.6.05

Table 4 describes the 5529 OAD features and enhancements added to the *5529 OAD Northbound Interface Guide* for Release 9.6.05.

Table 4 **What’s new in Release 9.6.05**

Feature/enhancement	Description	Edition	See
New features and enhancements			
getSystemHealthInfo operation	Added the getSystemHealthInfo operation, which allows the client application (and load balancers in a cluster deployment) to determine the availability of the 5529 OAD NBI server	01	Tables 6 and 9 Sections 4.8 and 7.5.5
Documentation changes			
Sample NBI operations	Added information about downloading a .zip file that contains samples of all of the 5529 OAD NBI request and response operations from the services page	01	Section 3.4

2.3 What’s new in Release 9.6.03

Table 5 describes the 5529 OAD features and enhancements added to the *5529 OAD Northbound Interface Guide* for Release 9.6.03.

Table 5 **What’s new in Release 9.6.03**

Feature/enhancement	Description	Edition	See
New features and enhancements			
HTTPS interface	The HTTPS interface is the default interface. The HTTP interface is disabled by default.	01	Sections 3.4, 4.1.2, and 4.3.1
JBoss libraries for JMS client application	The name of the oad-oss-client-1_*release-version*.jar library is changed to oad-oss-client-*release-version*.jar. The jboss-client.jar library is removed. The following libraries are new: • axs-encription-app-*release-version*.jar • commons-io-2.4.jar • jboss-logging-3.3.0.Final.jar • picketbox-4.9.6.Final.jar • picketbox-infinispan-4.9.6.Final.jar • slf4j-simple-1.7.21.jar • wildfly-client-all.jar • xbean-2.6.0.jar	01	Section 5.4.6
Login mechanism	Added a note stating that the 5520 AMS implements a rate-limiting login mechanism	01	Section 4.3

(1 of 2)

Feature/enhancement	Description	Edition	See
Documentation changes			
Alarm source names	Alarm source names are described in the *5520 AMS and 5529 Enhanced Applications Alarm Search Tool* (previously described in the Operations and Maintenance guide for the NE).	01	Tables 1 and 20
getActiveAlarms operation	Updated the description of the getActiveAlarms operation	01	Section 4.4.1
getActiveAlarmsCount operation	Updated the description of the getActiveAlarmsCount operation	01	Section 4.4.3
getActiveAlarmsIterator operation	Added the minimum and maximum number of alarms that can make up a batch for the getActiveAlarmsResponse message	01	Section 4.4.2
getSystemInfo operation	Updated the description and value for the timestamp element	01	Table 22
JBoss libraries for JMS client application	The oad-oss-client-1_*release-version*.jar file is removed from the list of JBoss libraries, as it is not a library. The file and the JBoss libraries are provided in the 5529 OAD OSS client tar file.	01	Section 5.4.6
MTOSI UTC format	Added information about the MTOSI UTC format used to display a date and time in an NBI operation	01	Section 3.5.1 Tables 19, 20, 21, and 23
Service endpoints	Combined the service endpoint information in a new section	01	Section 4.3
Supported version of Java	The supported version of Java is 1.8 as of 5529 OAD R9.6.	01	Section 3.5

(2 of 2)

3 Getting started

3.1 General

The 5529 OSS Alarm Dispatcher is an application that integrates with the 5520 AMS, and facilitates the incorporation of alarm flows into OSS client applications.

The 5529 OAD can also collect alarm information from one 5523 AWS R7.x and one 5520 AMS, and forward alarms simultaneously from both EMSs running at burst rates. This feature can prove very useful when preparing for the migration from the 5523 AWS to the 5520 AMS.

3.2 Functionality

The 5529 OAD simplifies the alarm collection, aggregation, and dispatching, and allows the OSS client applications to perform the following tasks:

- request and receive active alarms and alarm counts from the 5520 AMS
- subscribe to JMS fault topics, and receive notifications of alarms from 5520 AMS systems, as well as alarms and events from the 5529 OAD
- receive alarm information as SNMP traps

The 5529 OAD architecture is based on two northbound interfaces that support data exchange with OSS client applications:

- a web service interface over HTTP/S, which allows OSS client applications to send requests and receive responses about current active alarms
- a JMS interface, which allows OSS client applications to subscribe to and receive alarm and event notifications by way of a JMS provider

The 5529 OAD also provides the option to forward alarms as SNMP traps. The SNMP alarm traps are forwarded as UDP packets over the LAN.

3.3 Deployment models

For information about 5529 OAD standalone and cluster deployments, see the *5529 OAD Installation and Administration Guide*.

3.4 Support files

The 5529 OAD is delivered with a set of support files designed to help you develop client applications. The support files contain the following:

- sample OSS client code for both HTTP/S and JMS client applications
- libraries required to compile JMS client applications created in Java

For information about downloading and installing the support files, see the *5529 Enhanced Applications Release Notice*.

You need the 5529 OAD WSDL and XSD schema source files to design the OSS client applications. The schema documentation for the activated 5529 OAD is available at the following URLs:

- https://*host*:8443/oad/services
- http://*host*:8080/oad/services

where *host* is the IP address or host name of the application server.

Note — The HTTPS interface is the default interface. The HTTP interface is disabled by default. See section 4.1.2 for more information about the HTTPS interface. See the *5520 AMS Administrator Guide* for information about how to enable HTTP in the 5520 AMS.

Log in to the services page with the user name and password of a 5520 AMS user with the OAD NBI role. You can download the schema files individually or as a .zip file. You can also download a .zip file that contains XML samples of all of the 5529 OAD NBI request and response operations.

Note — You cannot open multiple NBI schema documentation pages that belong to different 5529 Enhanced Applications in the same web browser. If you attempt to do so, you receive a “File not found” error message.

If you want to open two different NBI schema documentation pages at the same time, use one of the following options:

- Use different web browsers for different schema documentation.
- Log in with a user ID that has the AMS NBI role and all of the 5529 Enhanced Applications NBI roles.
- Each time you refresh the web page or when you navigate to the WSDL link of a different schema that was opened before the current schema documentation in the same browser, you need to clear the browser private data, specifically website login information. Different browsers perform this function differently, and use different terminology to refer to private data, such as cache, cookies, and active logins. Consult information about the browser to determine how to perform this function.

Table 6 lists the locations of the WSDL source files.

Table 6 **WSDL locations within the schema files**

Web services operations	WSDL file	Location
getActiveAlarms getActiveAlarmsCount getActiveAlarmsIterator	FaultService.wsdl	\schema\tmf854\tmf854_R1.1\wsdl\services\FaultService.wsdl
acknowledgeAlarms unacknowledgeAlarms	AlarmRetrievalMgrExtnService.wsdl	\schema\alu\wsdl\services\AlarmRetrievalMgrExtnService.wsdl
getSystemHealthInfo getSystemInfo	InventoryRetrievalMgrExtnsSOAP.wsdl	\schema\alu\wsdl\binding\soap_http\InventoryRetrievalMgrExtnsSOAP.wsdl

The default version of NotificationService.wsdl is available at the following URLs:

- https://*host*:8443/oad/services
- http://*host*:8080/oad/services

where *host* is the IP address or host name of the application server

Note — The HTTPS interface is the default interface. The HTTP interface is disabled by default.

For JMS notifications with alarm grouping, you can use an alternate version of NotificationService.wsdl. The file, called oad_doc_with_alarm_grouping.zip, is available in $AMS_LOCAL_DATA_DIR/ams/local/repository/app-oad-*version_release-revision*/docs, where *version_release-revision* is the version, release, and revision of the 5529 OAD software. An example of *version_release-revision* is 1.0_9.6.07-999000. The file contains both WSDL and XSD files for grouped notifications and HTML schema documents. The schema files are in a schema sub-directory.

The default version is MTOSI-compliant; the alternate version is not. Both versions are SOAP-compliant.

For JMS notifications with encoding, a utility called StringCodec is available in the sample OSS client file. If the JMS message type is “Bytes”, you need to use the StringCodec utility to decode the JMS SOAP payload. StringCodec is found in the following file when the OSS client file is extracted:

```
lib/oss-client-utils-src-1.0.jar
```

JMS notifications with alarm grouping use encoding.

3.5 Technologies and standards

The 5529 OAD software architecture is built on open-source interoperable technologies such as MTOSI, SOAP and XML, Java and J2EE framework, multi-tier layering, and web services interfaces. The use of standard interfaces allows the 5529 OAD to easily integrate with OSS client applications.

The supported version of Java is 1.8.

To create OSS client applications that interface with the 5529 OAD, you need to be familiar with the following technologies and standards:

- XML 1.0
- XSD (XML Schema instance) 1.0
- SOAP 1.1/1.2
- WSDL 1.1
- JMS 1.1
- HTTP/S 1.1
- MTOSI 1.1
- TMF OSS interfaces (TMF 854)
- SNMPv2

See the following websites for additional information:

- http://www.w3.org/TR/2004/REC-xml-20040204 (W3C standards about XML 1.0, third edition)
- http://www.w3.org/TR/2001/REC-xmlschema-1-20010502 (XSD)
- http://www.w3.org/TR/2000/NOTE-SOAP-20000508 (SOAP 1.1)
- https://www.w3.org/TR/soap12/ (SOAP 1.2)
- http://www.w3.org/TR/2001/NOTE-wsdl-20010315 (WSDL 1.1)
- http://www.oracle.com/technetwork/java/jms/index.html (JMS 1.1)
- http://www.ietf.org/rfc/rfc2616.txt (HTTP/S 1.1)
- http://www.tmforum.org (TMF and MTOSI)

3.5.1 Date and time in MTOSI UTC format

This section describes the MTOSI UTC format that is used for a date and time that may be displayed in a 5529 OAD NBI operation; for example, a timestamp or the current date and time on the local host.

The date and time in MTOSI UTC format is expressed as *yyyyMMddHHmmss.S*[Z|{+|-}*HHMm*]

where

yyyy is the four-digit year

MM is the month (01-12)

dd is the day (01-31)

HH is the hour (00-23)

mm is the minute (00-59)

ss is the second (00-59)

S is one or more digits representing a decimal fraction of a second

Z indicates UTC (rather than local time)

+ is the positive offset from UTC

- is the negative offset from UTC

HHMm is the number of hours and minutes that the local time is offset from UTC

where

The time zone designator (Z or +*HHMm* or -*HHMm*) defines two ways of handling time zone offsets:

- Time is expressed in UTC, with a special UTC designator (Z).
- Time is expressed in local time, together with a time zone offset in hours (*HH*) and minutes (*Mm*). The +*HHMm* designator indicates the hours and minutes the local time is ahead of UTC. The -*HHMm* designator indicates the hours and minutes the local time is behind UTC.

Examples:

20170103180510.242-0500 corresponds to January 03, 2017, 18:05:10.242 (6:05:10.242 p.m.), Eastern Standard Time (-05:00 from UTC).

20170103180510.242Z corresponds to the same date and time as the previous example but is expressed in UTC.

5529 OAD Northbound interface

4 Operations

4.1 Alarm information retrieval

This section describes the retrieval of active alarm information in the 5529 OAD architecture.

4.1.1 Web services

The 5529 OAD architecture supports alarm retrieval operations by way of a web services server established between the OSS and the 5529 OAD. The alarm retrieval operations are synchronous RPC-type transactions, in which the data types and the request and response messages are strictly defined.

The OSS client applications and the 5529 OAD use SOAP as the web service messaging format for exchanging XML-based messages over an HTTP/S interface. The web services operate as defined in WSDL files, which are structured XML documents. Schema files describe the structure and elements of SOAP XML messages, which are formatted according to the MTOSI 1.1 standard requirements.

To send requests to the 5529 OAD using web services, an OSS client application must specify a user ID and password as part of the HTTP/S header.

4.1.2 HTTPS interface

The HTTPS interface is the default interface and requires a valid security certificate on the 5520 AMS server. The HTTPS security certificate installation is part of the core 5520 AMS software installation. The 5520 AMS administrator is responsible for ensuring that a valid HTTPS security certificate is always present on the 5520 AMS server. See the *5520 AMS Administrator Guide* for information about HTTP/S and how to use a security certificate in the 5520 AMS.

4.1.3 Information model

To retrieve active alarm information, you need to create an OSS client application that incorporates the alarm retrieval request wrapped in a SOAP envelope. To do so, you need the WDSL files; see section 3.4.

The 5529 OAD receives the request, collects the requested alarm information from the 5520 AMS, and dispatches a SOAP envelope containing a response with alarm information to the OSS.

The SOAP envelope for the XML request and response contains information about the schema that is used to describe the information in the message, as well as the parent elements described in Table 7.

Table 7 **SOAP envelope parent elements**

Element	Description
Header	The SOAP envelope header defines the XML Solution Set (TMF 854) supported by MTOSI, Release 1.1, as well as the version of the supported interface.
Body	The SOAP envelope body carries the operation and filtering information.

See chapter 7 for more information about the types of SOAP envelopes and XML message elements (request and response).

4.2 MTOSI object names

The 5520 AMS raises alarms against NEs, and NE equipment objects. The 5529 OAD retrieves the 5520 AMS alarms, maps them to the appropriate MTOSI-compliant object names, and dispatches the alarm information to the OSS client applications.

The MTOSI object names are defined according to the network object containment hierarchy. Table 8 lists the supported physical objects, shows the alarm mapping to MTOSI objects, and describes the MTOSI object naming convention.

Table 8 Alarm mapping to MTOSI objects

5520 AMS or physical object	MTOSI object	MTOSI object type	MTOSI naming convention
Element management system (5520 AMS, 5523 AWS, or 5529 OAD)	OperationsSystem	OT_OS	<osNm>*EMS name*</osNm>
Management Domain [1]	ManagementDomain	OT_MANAGEMENT_DOMAIN	<mdNm>*MD name*</mdNm>
Node	ManagedElement	OT_MANAGED_ELEMENT	<mdNm>*MD name*</mdNm> <meNm>*NodeName*</meNm>
Rack	EquipmentHolder	OT_EQUIPMENT_HOLDER	<mdNm>*MD name*</mdNm> <meNm>*NodeName*</meNm> <ehNm>/rack=*rackNr*</ehNm> [2]
Shelf or subrack	EquipmentHolder	OT_EQUIPMENT_HOLDER	<mdNm>*MD name*</mdNm> <meNm>*NodeName*</meNm> <ehNm>/rack=*rackNr*/shelf=*subrackNr*</ehNm> [2]
Slot	EquipmentHolder	OT_EQUIPMENT_HOLDER	<mdNm>*MD name*</mdNm> <meNm>*NodeName*</meNm> <ehNm>/rack=*rackNr*/shelf=*subrackNr*/slot=*slotNr*</ehNm> [2] [3]
Card	Equipment	OT_EQUIPMENT	<mdNm>*MD name*</mdNm> <meNm>*NodeName*</meNm> <ehNm>/rack=*rackNr*/shelf=*subrackNr*/slot=*slotNr*</ehNm> [2] [3] <eqNm>1</eqNm>
Port	PhysicalTermination Point	OT_FLOATING_TERMINATION_ POINT	<mdNm>*MD name*</mdNm> <meNm>*NodeName*</meNm> <ptpNm>/rack=*rackNr*/shelf=*subrackNr*/slot=*slotNr*/ port=*portNr*</ptpNm> [2] [3]
Voice User Port	FloatingTermination Point	OT_FLOATING_TERMINATION_ POINT	<mdNm>*MD name*</mdNm> <meNm>*NodeName*</meNm> <ftpNm>/rack=*rackNr*/shelf=*subrackNr*/slot=*slotNr*/ voiceUserPort=*portNr*</ftpNm> [2] [3]
H248 CPE	FloatingTermination Point	OT_FLOATING_TERMINATION_ POINT	<mdNm>*MD name*</mdNm> <meNm>*NodeName*</meNm> <ftpNm>/cpe=*cpeNr*</ftpNm> [2]
Interface Group	FloatingTermination Point	OT_FLOATING_TERMINATION_ POINT	<mdNm>*MD name*</mdNm> <meNm>*NodeName*</meNm> <ftpNm>/igType=*igType*/ig=*igNr*</ftpNm> [2]

Notes

[1] The Management Domain is an object introduced for naming purposes only, as the root of all managed objects names. You can define it as an EMS.

[2] In these MTOSI object names, *rackNr*, *subrackNr*, *cpeNr*, and *igNr* represent a numeric value and *portNr* and *igType* represent an alphanumeric string.

[3] In these MTOSI object names, *slotNr* is NTA, NTB, ACU_NTIO, or LT*n*, where *n* is the numeric value for the LT slot; for example, LT1. In the case of a GENBAND G6 NE, slot=*n*. *n* is the numeric value for the NE slot.

See chapter 7 for more information about the MTOSI objects, and the naming conventions used to specify objects and object types in the SOAP body.

4.3 Service endpoints

The OSS client applications send requests to a service endpoint. Service endpoints may depend on the 5529 OAD deployment.

Table 9 lists the service endpoint information for 5529 OAD NBI operations.

Note 1 — The HTTPS interface is the default interface. The HTTP interface is disabled by default.

Note 2 — The 5520 AMS implements a rate-limiting login mechanism based on each incoming IP address. See the *5520 AMS User Guide* for more information.

Table 9 **Service endpoint information for 5529 OAD NBI operations**

Operation name	**Service endpoint links**		**Notes**
	Standalone deployment	**Cluster deployment**	
acknowledgeAlarms unacknowledgeAlarms	https://*5520_AMS_IP_address*:8443/oad/services/AlarmRetrievalMgrExt http://*5520_AMS_IP_address*:8080/oad/services/AlarmRetrievalMgrExt	https://*load_balancer_IP_address*:8443/oad/services/AlarmRetrievalMgrExt http://*load_balancer_IP_address*:8080/oad/services/AlarmRetrievalMgrExt	*5520_AMS_IP_address* is the IP address of a 5520 AMS server *load_balancer_IP_address* is the IP address of the load balancer
getActiveAlarms getActiveAlarmsIterator getActiveAlarmsCount	https://*5520_AMS_IP_address*:8443/oad/services/AlarmRetrievalMgr http://*5520_AMS_IP_address*:8080/oad/services/AlarmRetrievalMgr	https://*load_balancer_IP_address*:8443/oad/services/AlarmRetrievalMgr http://*load_balancer_IP_address*:8080/oad/services/AlarmRetrievalMgr	*5520_AMS_IP_address* is the IP address of a 5520 AMS server *load_balancer_IP_address* is the IP address of the load balancer
getSystemHealthInfo	https://*5520_AMS_IP_address*:8443/oad/services/InventoryRetrievalMgrExtns http://*5520_AMS_IP_address*:8080/oad/services/InventoryRetrievalMgrExtns	https://*load_balancer_IP_address*:8443/oad/services/InventoryRetrievalMgrExtns http://*load_balancer_IP_address*:8080/oad/services/InventoryRetrievalMgrExtns	*5520_AMS_IP_address* is the IP address of a 5520 AMS server *load_balancer_IP_address* is the IP address of the load balancer
getSystemInfo	https://*host*:8443/oad/services/InventoryRetrievalMgrExtns http://*host*:8080/oad/services/InventoryRetrievalMgrExtns		*host* is the IP address or host name of the application server

4.3.1 Verifying the 5529 OAD service endpoints

Before sending requests, it is good practice to verify the 5529 OAD service endpoints. Use a web browser to navigate to the service endpoints, and look for a message similar to the following:

```
Hi there, this is an AXIS service!
```

This message indicates that the service endpoint is functional.

To be able to access the services web page, you need to have a user ID with a role that contains the OAD NBI - View function, and a password for the user. See the *5529 OAD Installation and Administration Guide* for information about the 5529 OAD user functions and roles. See the *5520 AMS Administrator Guide* for information about managing user functions and roles.

4.4 Alarm retrieval operations

Table 10 describes the standard alarm-retrieval MTOSI-compliant operations that the 5529 OAD supports.

You can use alarm retrieval operations for one-time or occasional updates. For real-time updates to the OSS client, use JMS subscriptions; see chapter 5.

Table 10 **Alarm retrieval MTOSI-compliant operations**

Operation	Description	Response
getActiveAlarms	Requests all or a specified subset of the current active alarms from the 5520 AMS	A successful getActiveAlarms operation returns a message with information about the current active alarms from the 5520 AMS in an MTOSI -compliant getActiveAlarmsResponse structure.
getActiveAlarmsIterator	Requests current active alarms from the 5520 AMS in batches, according to a batch size specified in the header of a previous getActiveAlarms request	
getActiveAlarmsCount	Requests a count of active alarms from the 5520 AMS	A successful getActiveAlarmsCount operation returns the number of current active alarms in the 5520 AMS in an MTOSI-compliant getActiveAlarmsCountResponse structure.

Only one getActiveAlarms or getActiveAlarmsCount operation can run at a time. If a getActiveAlarms or getActiveAlarmsCount operation is already running, the 5529 OAD rejects a request for another getActiveAlarms or getActiveAlarmsCount operation with an exception of type EXCPT_CAPACITY_EXCEEDED.

See section 4.3 for information about alarm retrieval request endpoints.

4.4.1 getActiveAlarms

An OSS client sends a getActiveAlarms request to retrieve the current active alarms from a 5520 AMS. The 5529 OAD can retrieve all of the active alarms in the network if the OSS client request does not include any alarm filters. The 5529 OAD can retrieve a subset of the alarms depending in the filter constraints that the OSS client request specifies in the body of the SOAP envelope; for example, the scope filter attribute allows you to specify one or more managed elements. See section 4.5 for information about alarm filters.

The response includes an iteratorReference URI; for example, http://7612937130408079365.

Note — Only one getActiveAlarms or getActiveAlarmsCount operation can run at a time. If a getActiveAlarms or getActiveAlarmsCount operation is already running, the 5529 OAD rejects a request for another getActiveAlarms or getActiveAlarmsCount operation with an exception of type EXCPT_CAPACITY_EXCEEDED.

4.4.2 getActiveAlarmsIterator

The getActiveAlarmsIterator operation requests the next iteration of active alarms in a getActiveAlarms operation sequence. The getActiveAlarmsIterator request needs to include the iteratorReferenceURI that was returned by the previous request.

A successful operation returns the next batch of alarms from the 5520 AMS. Use this operation until the final batch of alarms is returned.

After receiving a getActiveAlarmsIterator request, the 5529 OAD checks whether it has enough alarms to create one batch (the default is 0 to 5000), and proceeds as follows:

- If there are enough alarms, the 5529 OAD prepares a getActiveAlarmsResponse SOAP envelope containing as many alarms as the OSS client application specified in the initial request, and sends the response to the OSS client.
- If the 5529 OAD determines that there are not enough alarms for one batch in the alarm cache, the 5529 OAD waits for more alarms (or for the end of alarms) before sending a new batch to the OSS client.

The request and response sequence that begins with the getActiveAlarms operation and ends with the last getActiveAlarmsResponse is an iteration sequence. During the iteration sequence, the 5529 OAD must receive each request message within a preset time-out period. For information about how to change the time-out period, see the *5529 OAD Installation and Administration Guide*.

The behavior of the 5529 OAD time-out feature is as follows:

- If the 5529 OAD receives a getActiveAlarmsIterator request from the OSS client within the time-out period, the 5529 OAD resets the timer when the next response is sent to the client.
- If the 5529 OAD does not receive a getActiveAlarmsIterator request within the time-out period, the 5529 OAD terminates the iteration sequence, and discards the remaining alarm information.
- If the 5529 OAD receives a getActiveAlarmsIterator request after the iteration sequence has been terminated due to a time-out, the 5529 OAD raises an exception of type EXCPT_UNABLE_TO_COMPLY.

In cluster deployments, the load balancer is responsible for keeping data from multiple requests in the same user session. The load balancer selects a cluster node for the first request using a load balancing algorithm, and dispatches all the subsequent requests in the same session to the same cluster node. All iterations that are part of a getActiveAlarmsIterator operation are processed by the same 5529 OAD instance in the cluster.

The load balancer achieves this task by way of a layer 7 persistence method that uses a cookie to identify the cluster node that must process all the requests of an iterative operation.

The following is an outline of the way the cookie persistence method works in a cluster deployment.

- The load balancer receives a getActiveAlarms operation request and directs it to a cluster node.
- The cluster node receives the getActiveAlarmsIterator operation request, generates a cookie, and attaches the cookie to the response. The cookie has a name and a value:
  - name: AMS_WS_ITER (a constant value, the same for all operations)
  - value: a string of numeric characters unique for each operation (same as the value of the iteratorReferenceURI field in the message header)
- The load balancer receives the response message, associates the value of the cookie with the cluster node that sent the response, and dispatches the message to the OSS client.
- The next OSS client request that is part of the same getActiveAlarmsIterator operation must include the cookie, so that the load balancer will forward the request to the same cluster node.

The 5529 OAD, the load balancer, and the OSS client use the same cookie value for all the iterations in the same iterative operation.

4.4.3 getActiveAlarmsCount

An OSS client sends a getActiveAlarmsCount request to retrieve the count of current active alarms from a 5520 AMS. The 5529 OAD can retrieve the count of all alarms in the network if the OSS client request does not include any alarm filters. The 5529 OAD can retrieve a subset of the alarms depending on the filter constraints that the OSS client request specifies in the body of the SOAP envelope; for example, the scope filter attribute allows you to specify one or more managed elements. See section 4.5 for information about alarm filters.

Note — Only one getActiveAlarms or getActiveAlarmsCount operation can run at a time. If a getActiveAlarms or getActiveAlarmsCount operation is already running, the 5529 OAD rejects a request for another getActiveAlarms or getActiveAlarmsCount operation with an exception of type EXCPT_CAPACITY_EXCEEDED.

4.5 Filters for active alarms

Table 11 describes the supported getActiveAlarms and getActiveAlarmsCount filter attributes, as defined in the ActiveAlarmFilter_T XSD file. Table 11 lists only the attributes that are supported by the 5529 OAD for the operation. There may be additional standard attributes that are displayed in the WSDL file, but those attributes are not used nor supported by the 5529 OAD.

Note — If a request operation contains multiple attributes that are duplicates of one another, the NBI processes only the latest attribute in the list, which is the attribute listed last.

Table 11 **Filter attributes for getActiveAlarms and getActiveAlarmsCount**

Attribute	Description	Values
source	Filters the alarms based on the alarm source from the perspective of the application that receives the request	EXTERNAL NOT_APPLICABLE
scope	Filters the alarms based on a list of FDNs identifying managed elements in the network. • If the FDN of a managed element is specified, the 5529 OAD reports the alarms from that managed element object and its children. • If no managed element is provided, the 5529 OAD reports alarms from all the managed elements. The following exceptions apply: • If a managed element FDN in the list is incorrectly formed and therefore invalid, or the FDN identifies an object that is not a ManagedElement object, the 5529 OAD raises an exception of type EXCPT_UNABLE_TO_COMPLY. • If a managed object FDN in the list is correctly formed, but does not identify a ManagedElement object in the network, no alarms are returned.	The hierarchical names of the managed element objects

(1 of 2)

Attribute	Description	Values
perceivedSeverityList	Filters the alarms based on a list of severity values. The 5529 OAD returns only the alarms that conform to the listed severity information. If no severity value is provided, the 5529 OAD returns alarms of all severity types. The perceivedSeverityList attribute can contain one or more perceivedSeverity attributes. Each perceivedSeverity attribute contains a severity value.	PS_INDETERMINATE PS_CRITICAL PS_MAJOR PS_MINOR PS_WARNING PS_CLEARED
probableCauseList	Filters the alarms based on a list of probable causes. If no probable cause is provided, alarms with any probable cause are returned. This attribute is a container element for the probableCause attribute.	—
probableCause	This attribute is a container element for the following probable cause attributes: • ru • contra • type	—
ru	Probable cause attribute indicating the location of the network resource with which the alarm probable cause is associated. The attribute name stands for Remote Unit.	The options are true or false. The value is always false.
contra	Probable cause attribute indicating the direction of the signal-related alarm	The options are true or false. The value is always false.
type	Indicates the probable cause specific value. The list can contain one type value, or more values separated by commas.	Character string [1]
acknowledgeIndication	Filters the alarms based on their acknowledge indication	AI_EVENT_ACKNOWLEDGED AI_EVENT_UNACKNOWLEDGED AI_NA [2]

(2 of 2)

Notes

[1] The values of this parameter are specified in the MTOSI 1.1 standard, and are in the ProbableCause_T XSD file.

[2] The AI_NA value indicates that alarms with any acknowledge indication are returned. The value is provided in case the OS does not support acknowledgement for the event, or does not support acknowledgement at all.

4.6 Alarm acknowledgement and unacknowledgement operations

The 5529 OAD also supports the Nokia proprietary acknowledgeAlarms and unacknowledgeAlarms operations.

An OSS client application sends an acknowledgeAlarms or unacknowledgeAlarms request to acknowledge alarms received from the 5520 AMS, or mark previously acknowledged alarms as unacknowledged. The body of both requests contains an alarmList container element, which includes a notificationId attribute for each alarm to be acknowledged or unacknowledged.

The notificationId attribute contains a combination of the domain name (the name of the 5520 AMS, which is configured using the OAD NBI Settings page on the 5520 AMS GUI, as described in the *5529 OAD Installation and Administration Guide*) and the 5520 AMS alarm ID separated by a colon.

The body of an acknowledgeAlarms or unacknowledgeAlarms response contains an alarmAckStatusList container element, which includes an alarmAckStatus attribute for each alarm listed in the request. The alarmAckStatus attribute contains the domain name, the alarmId, and the operation status (success or failure), all separated by a colon.

When the unacknowledgeAlarms operation is performed, the values for the following 5529 OAD attributes are cleared:

- lastAcknowledgedUserId
- lastAcknowledgedInetAddress
- lastAcknowledgedTimeStamp

The OSS operations that modify the 5520 AMS and NEs are tracked in the user activity log. See the *5520 AMS Administrator Guide* for information about the user activity log.

See section 4.3 for information about alarm acknowledgement request endpoints.

4.7 getSystemInfo operation

The 5529 OAD supports the Nokia proprietary getSystemInfo operation.

An OSS client application uses this operation to request information about the 5529 OAD application and the IP address of the 5520 AMS master application server. The getSystemInfo operation request contains no input attribute. A successful getSystemInfo operation retrieves the applicable information, and returns it within a getSystemInfoResponse SOAP XML message.

See section 7.5.4 for information about the elements that are present in the header and body of the getSystemInfo request and response messages.

See section 4.3 for information about alarm getSystemInfo request endpoints.

4.8 getSystemHealthInfo operation

The getSystemHealthInfo operation allows the OSS client application (and load balancers in a cluster deployment) to determine the availability of the 5529 OAD NBI server. If the NBI server is available, it is able to process NBI operation requests and successfully provide operation responses.

See section 7.5.5 for information about the elements that are present in the header and body of the getSystemHealthInfo request and response messages.

4.9 Exceptions

Table 12 describes the exceptions that the 5529 OAD can raise in error conditions for the NBI operations.

Table 12 **Operation exceptions**

Exception	Applies to	Description
EXCPT_ACCESS_DENIED	acknowledgeAlarms getActiveAlarms getActiveAlarmsCount getActiveAlarmsIterator	Raised if there is an authentication error
EXCPT_CAPACITY_EXCEEDED		Raised if a getActiveAlarms or getActiveAlarmsCount operation is running, and the 5529 OAD receives a request for another getActiveAlarms or getActiveAlarmsCount operation
EXCPT_COMM_FAILURE	acknowledgeAlarms getActiveAlarms getActiveAlarmsCount getActiveAlarmsIterator getSystemInfo	Raised if the 5529 OAD detects a loss of communication with the 5520 AMS during an operation
EXCPT_INTERNAL_ERROR	acknowledgeAlarms getActiveAlarms getActiveAlarmsCount getActiveAlarmsIterator getSystemInfo	Raised if the 5529 OAD encounters an unexpected runtime error or a non-specific OS internal failure
EXCPT_INVALID_INPUT	acknowledgeAlarms getActiveAlarms getActiveAlarmsCount getActiveAlarmsIterator	Raised if an attribute value in the SOAP XML request is invalid (wrong format, incorrect value type, or out-of-range value). The reason provided in the exception body highlights the attribute that caused the error, and suggests valid values for that attribute.
EXCPT_NOT_IMPLEMENTED	getActiveAlarms getActiveAlarmsCount getActiveAlarmsIterator	Raised if the 5529 OAD does not support an operation used in a request, or the requested communication pattern. For example: • the only communication pattern supported for the getActiveAlarms and getActiveAlarmsIterator requests is MultipleBatchResponse. • the only communication pattern supported for the getActiveAlarmsCount request is SimpleResponse If any other communication pattern is used in the request, the 5529 OAD raises an EXCPT_NOT_IMPLEMENTED exception.
EXCPT_UNABLE_TO_COMPLY	getActiveAlarmsIterator	Raised if the 5529 OAD receives a getActiveAlarmsIterator request after terminating the iteration sequence due to a time-out

4.10 5523 AWS support considerations

The 5529 OAD supports the alarm collection from one 5523 AWS R7.x and one 5520 AMS, and forwards alarm information simultaneously from both EMSs. This feature is useful when you prepare the migration from the 5523 AWS to the 5520 AMS.

The following considerations apply to the alarm retrieval operations in configurations with one 5523 AWS R7.x and one 5520 AMS.

- The alarm retrieval operations that do not specify any attributes are directed to the 5520 AMS.
- The alarm retrieval operations that specify attributes, including a managed domain name, are directed to the EMS with the specified MD name.
- The alarm operations that specify attributes containing more than one distinct managed domain name are not supported.

5 JMS notifications

5.1 Event mapping to MTOSI 1.1 objects

The 5529 OSS Alarm Dispatcher captures the following events in notification messages:

- 5520 AMS/5523 AWS and 5529 OAD alarms
- 5529 OAD events

The 5529 OAD translates event information into an MTOSI 1.1 alarm structure and sends the alarm structure to subscribed JMS clients by way of the JMS provider. Table 13 lists the mapping of events to MTOSI 1.1 objects.

Table 13 **Event mapping to MTOSI 1.1 objects**

Event	Type	Source	MTOSI 1.1 object
Alarm	5520 AMS/5523 AWS alarm	5520 AMS or 5523 AWS	Notify_T/Alarm_T
	License validation alarm	5520 AMS	Notify_T/Alarm_T
Heartbeat	Heartbeat	5529 OAD	Notify_T/Heartbeat_T
EventLossOccured	EventLossOccured	5529 OAD	Notify_T/EventLossOccured_T
EventLossCleared	EventLossCleared	5529 OAD	Notify_T/EventLossCleared_T

Regardless of the object type, the 5529 OAD notifies the JMS client of all alarm events generated by the 5520 AMS or 5523 AWS. When the object that is raising the alarm does not have a corresponding MTOSI object, the alarm from the 5520 AMS or 5523 AWS is mapped to its closest parent objects on the node. The supported parent objects are node, rack, shelf, slot, card, and port. These objects are mapped in MTOSI as indicated in chapter 7.

The license validation alarm, as well as the Heartbeat, EventLossOccured, and EventLossCleared events, are reported on the 5529 OAD.

5.2 Events

Table 14 describes the events for which the OSS clients can subscribe to get notifications.

Note — The 5529 OAD NBI user that is subscribed to JMS event notifications receives alarm notifications for all of the NEs that are managed by the 5520 AMS. The user does not receive alarm notifications only for NEs that have the same PAP group as the user.

If the alarms are encoded, the JMS msgType (message type) in the JMS header is “Bytes”. Otherwise, the JMS msgType (message type) in the message header is null.

Table 14 **Event description**

Event	Type	Description
Alarm	5520 AMS/5523 AWS alarm	Alarm that originates on the 5520 AMS or 5523 AWS
	License validation alarm	Alarm that the 5529 OAD generates when the license expires
Heartbeat	Heartbeat	Event that the 5529 OAD sends to an OSS client through JMS to indicate that the 5529 OAD is running and processing alarms
EventLossOccured	EventLossOccured	Event that the 5529 OAD sends to an OSS client through JMS to indicate that queue overflow condition has occurred in the 5529 OAD
EventLossCleared	EventLossCleared	Event that the 5529 OAD sends to an OSS through JMS to indicate that a queue overflow condition has been cleared in the 5529 OAD
Batch Notification	Batch Notification	Event that the 5529 OAD sends to an OSS through JMS that contains a group of alarms

5.2.1 Detailed information with cleared alarm events

When an alarm is cleared, by default the JMS provider sends an event that names the alarm and the time the alarm was cleared. You can configure the 5529 OAD to send detailed information with the event instead, including all of the attributes of the alarm. For more information about configuring the 5529 OAD to send detailed cleared alarm events, see the *5529 OAD Installation and Administration Guide*.

⚠ **Caution —** Enabling detailed cleared alarm events on a network with a high volume of alarm messages may impact performance, as it increases the size of the cleared alarm messages.

5.3 Information model

The 5529 OAD architecture supports event notifications by way of a JMS provider established between the OSS and the 5529 OAD. The subscription and notification process is an asynchronous MSG-type transaction, in which the data types and the notification messages are strictly defined.

5.3.1 SOAP XML messages

The JMS provider and the 5529 OAD use a SOAP envelope as the format for XML messages. The JMS notification is an XML-based SOAP envelope message formatted according to the MTOSI 1.1 standard requirements.

The SOAP envelope for the XML notification contains information about the schema that is used to describe the information in the message, as well as the parent elements described in Table 15.

Table 15 **SOAP envelope parent elements**

Element	Description
Header	The SOAP envelope header defines the XML Solution Set (TMF 854) supported by MTOSI, Release 1.1, as well as the version of the supported interface. The 5529 OAD supports MTOSI Version 1.1.
Body	The SOAP envelope body contains the alarm event notification.

See chapter 7 for more information about the types of SOAP envelopes and XML message elements. You can also find a detailed description of the attributes present in the SOAP header and body for each type of XML message.

The header attributes are the same in the alarm and event notifications, while the body attributes in the event notification are just a subset of those present in an alarm notification.

5.3.2 Notification identifier

Each event notification SOAP XML message includes a unique identifier (the notificationId attribute), which consists of a prefix followed by a colon and a number. Table 16 describes the format of the notification identifier for each supported event.

Table 16 **Event notification identifiers**

Event	Format	Description
5520 AMS or 5523 AWS alarm	*domain_name*:*number*	The *domain_name* variable is the 5520 AMS or 5523 AWS domain name that is configured in the OAD NBI settings in the 5520 AMS GUI; for example, AMS1, AMS2, AMS3, or AWS1. The default is AMS or AWS. The *number* variable is a unique integer assigned to the event notification.
Heartbeat	*domain_name*-OAD:*number*	
EventLossOccured	OAD:*number*	
EventLossCleared		

5.4 JMS client application guidelines

To receive event notifications from the 5529 OAD by way of the JMS provider, you need to create a JMS client application to subscribe to a JMS topic. Because the current 5529 OAD architecture does not support the MTOSI 1.1 subscribe and unsubscribe operations, the JMS client application must subscribe through the JMS subscription API.

5.4.1 System time

The system time on the OSS client workstation needs to be set correctly and synchronized with the 5520 AMS server on which the 5529 OAD is running. This ensures that the JMS client receives notifications properly.

5.4.2 Properties

A JMS client application needs to include the following information:

- IP address of the JMS provider
- keystore and keypass information
- subscriber user name and password
- subscription topic name
- subscription filter (see section 5.5 for more information)

5.4.3 JMS client application tasks

A JMS client application that subscribes to notifications needs to perform the following tasks:

- use the IP address of an application server and port 4447
- get a topic connection factory from the JNDI context
- create a topic connection using the previous factory
- look up the topic of interest in the JNDI context
- from the topic connection, create a topic session object
- create a topic subscriber on the topic read from the JNDI context
- set up a message listener on the topic subscriber
- handle a message that arrives to the topic subscriber.

  If the message is encoded, the client needs to verify that the value of the JMS message type is “Bytes”. If the message is not encoded, the JMS message type will be null.

Note — The 5529 OAD uses a singleton connection factory implementation to enable the JBoss to automatically identify the active application server. It is not necessary to use a getSystemInfo request to identify the interface for JMS subscriptions.

5.4.4 JMS ports

For information about the JMS ports that must be enabled on the firewall, see the *5520 AMS Solution Planning Guide*.

5.4.5 JMS application parameters

This section provides the values of the parameters that a JMS client application uses to successfully connect to the 5529 OAD in order to subscribe to the Fault topic.

The following parameters apply to standalone deployments:

- java.naming.factory.initial=“org.jboss.naming.remote.client.InitialContext Factory”
- java.naming.provider.url=“remote://*5529 OAD_host_IP address*:4447”

The following parameters apply to cluster deployments:

- java.naming.factory.initial="org.jboss.naming.remote.client.InitialContext Factory"
- java.naming.provider.url=use a comma-separated list of entries that follow the "remote://*cluster_node_IP address*:4447" format; for example, "remote://*cluster_node1_IP address*:4447, remote://*cluster_node2_IP address*:4447, remote://*cluster_node3_IP address*:4447"

The following parameters apply to standalone and cluster deployments and are required to establish a connection over SSL between the JMS client application and JMS server:

- java.naming.security.principal=admin
- java.naming.security.credentials=admin
- jboss.naming.client.connect.options.org.xnio.Options.SSL_STARTTLS=true
- jboss.naming.client.connect.options.org.xnio.Options.SASL_POLICY_NOPLAINTEXT=false
- jboss.naming.client.remote.connectionprovider.create.options.org.xnio.Options.SSL_ENABLED=true

For standalone and cluster deployments, the JMS client application also needs to have the following of its system java properties set correctly:

- javax.net.ssl.keyStore
- javax.net.ssl.keyStorePassword
- javax.net.ssl.trustStore
- javax.net.ssl.trustStorePassword

5.4.6 Libraries

The 5529 OAD is delivered with a set of support files intended to help you develop JMS client applications. In addition to the sample JMS client application code, the support files also include the libraries required to compile JMS client applications created in Java.

The security updates implemented by the 5520 AMS to provide SSL support require an interface defined in the axs-mobject-remote-api-*release*.jar library when a JMS subscriber is initialized to receive messages.

The 5520 AMS cluster support requires that the JMS client code reference the following JBoss libraries in its Java classpath:

- axs-encription-app-*release-version*.jar
- axs-mobject-api-*release-version*.jar
- axs-mobject-remote-api-*release-version*.jar
- commons-io-2.4.jar
- jboss-logging-3.3.0.Final.jar
- log4j-1.2.14.jar
- oss-client-utils-1.0.jar
- picketbox-4.9.6.Final.jar
- picketbox-infinispan-4.9.6.Final.jar
- slf4j-simple-1.7.21.jar
- wildfly-client-all.jar
- xbean-2.6.0.jar

where *release* is the 5529 OAD release number and *version* is the software version

The 5529 OAD OSS client .tar file that is listed in the *5529 Enhanced Applications Release Notice* contains the JBoss libraries and the oad-oss-client-*release-version*.jar file.

Caution — Ensure that you install the latest 5529 OAD OSS client and JBoss libraries provided with the 5529 OAD release.

For information about downloading and installing the support files, see the *5529 Enhanced Applications Release Notice*.

5.5 Subscription filter

The JMS provider offers extended filtering capabilities using a complex, SQL-based filtering syntax (SQL-92). After you specify a filter, the OSS client applications receive only the event notifications that match the filter criteria. If you did not specify a filter, the JMS client applications receive all the event notifications sent by the 5529 OAD.

Note 1 — When specifying filter criteria, the filter must include criteria for the EventLossOccured and EventLossCleared notifications in order to receive these notifications from the 5529 OAD.

Note 2 — JMS filtering is not compatible with alarm grouping. If you have configured alarm grouping, JMS filtering is disabled; see the *5529 OAD Installation and Administration Guide*.

Note 3 — JMS filtering is compatible with alarm encoding.

5.5.1 JMS property fields

The 5529 OAD supports the following JMS property fields to which you can apply filters for the event notification subscription:

- MTOSI_EventType
- MTOSI_notificationId
- MTOSI_objectName
- MTOSI_objectType
- MTOSI_osTime
- MTOSI_neTime
- MTOSI_startTime
- MTOSI_SourceFriendlyNameValue
- MTOSI_endTime
- MTOSI_probableCause
- MTOSI_nativeProbableCause
- MTOSI_perceivedSeverity
- MTOSI_serviceAffecting
- MTOSI_rcaiIndicator
- MTOSI_acknowledgeIndication
- MTOSI_X733_EventType

The JMS property fields map to MTOSI notification attributes, which are part of the structure of the notification message.

5.5.2 MTOSI_EventType values

In the 5529 OAD event notification context, the MTOSI_EventType property field can be filtered on the following values:

- Alarm
- Heartbeat
- EventLossOccured
- EventLossCleared

5.5.3 Filter content

A filter is a text string containing properties and logic operators. The following logic operators are supported:

- AND
- OR
- NOT

5.5.3.1 Filter examples

Table 17 describes several examples of filters.

Table 17 **Subscription filter examples**

Filter	Description
`MTOSI_EventType='Alarm'`	Receives only alarms
`MTOSI_EventType='Heartbeat'`	Receives only heartbeat notifications
`MTOSI_X733_EventType='communicationsAlarm'`	Receives only communications alarms

`MTOSI_perceivedSeverity='PS_CRITICAL' OR`
`MTOSI_perceivedSeverity='PS_CLEARED'`
	Receives only critical alarms and clear notifications

`NOT(MTOSI_objectType='OT_EQUIPMENT_HOLDER')`	Receives all notifications except those for an equipment holder

`NOT(MTOSI_objectType='OT_PHYSICAL_TERMINATION_POINT')`
`AND (MTOSI_perceivedSeverity='PS_INDETERMINATE' OR`
`MTOSI_perceivedSeverity='PS_MINOR' OR`
`MTOSI_perceivedSeverity='PS_WARNING')`
	Receives all notifications that are high priority and clear, except those for a physical termination point

`MTOSI_SourceFriendlyNameValue LIKE '%NE System%' OR`
`MTOSI_SourceFriendlyNameValue LIKE '%PON Port%' OR`
`MTOSI_SourceFriendlyNameValue LIKE '%Network Port%' OR`
`MTOSI_SourceFriendlyNameValue LIKE '%GbE Port%'`
	Receives all notifications that have source friendly names that match any of the wildcards. In this example, the wildcards are NE System, PON Port, Network Port, and GbE Port.durable

5.6 Persistence and durable subscriptions

This section describes the JMS message delivery modes supported by the 5529 OAD, and the durable subscriptions.

5.6.1 Persistence overview

The 5529 OAD supports two delivery modes for JMS messages to specify whether messages are lost or can be recovered if the JMS provider fails:

- The NON_PERSISTENT delivery mode does not require the JMS provider to store messages for future recovery if the JMS provider fails. This is the default option.
- The PERSISTENT delivery mode ensures that messages are not lost if the JMS provider fails. Persistent messages are logged to a stable storage area, and can be recovered.

You can configure the JMS Message Delivery Mode parameter in the 5520 AMS GUI to NON_PERSISTENT or PERSISTENT. See the *5529 OAD Installation and Administration Guide* for more information about configuring the 5529 OAD.

5.6.2 Durable subscriptions

With a durable subscription, if the subscriber disconnects (unintentionally), the subscription continues to exist and to hold messages until the application unsubscribes. Messages published while the subscriber is disconnected are received when the subscriber reconnects. The messages that arrive while the subscriber is disconnected are not lost.

To achieve a durable subscription, you need to configure the JMS client application to register with JMS for a durable subscription, in addition to using the PERSISTENT delivery mode in the 5529 OAD.

By default, one durable subscriber is allowed to be subscribed to the Fault topic at one time. You cannot register a durable subscription with the fault topic if the maximum number of durable subscribers is exceeded. Contact your Nokia service representative to change the maximum number of durable subscribers.

Note — You can configure the 5529 OAD to automatically unsubscribe a subscriber when the subscriber disconnects. For an example of how to make that configuration using the 5529 OAD sample client, go to the JmsSubClient.properties of the 5529 OAD sample client and set the unsubscribe attribute to true.

If you experience problems with fault topic subscriptions, contact your Nokia service representative.

A durable subscriber registers a durable subscription with a unique identity that is retained by the JMS provider. Subsequent subscriber objects with the same identity resume the subscription in the state in which it was left by the previous subscriber. If a durable subscription has no active subscriber, the JMS provider retains the messages that are part of the subscription until they are received by the subscription or until they expire.

You need to establish a unique identity for a durable subscriber by setting the following information:

- a client ID for the connection
- a topic and a subscription name for the subscriber

The JMS provider stores the messages published to the topic as if they were messages sent to a queue. If another JMS client application uses the same connection factory and client ID, and the same topic and subscription name, the subscription is reactivated, and the JMS provider delivers the messages that were published while the subscriber was inactive, and that have not yet expired.

If a durable subscriber is disconnected and the threshold of messages in the topic queue is reached, the 5529 OAD logs a warning message and continues forwarding messages to other subscribers.

When a durable subscriber becomes disconnected, Nokia recommends that the subscriber try to subscribe again using the same ID or unsubscribe existing subscribers if the maximum number of durable subscribers is exceeded. You can display the list of durable subscribers and unsubscribe durable subscribers in the 5520 AMS GUI; see the *5529 OAD Installation and Administration Guide*.

If the subscription is active and the durable client is not connected, the JMS messages expire in 15 minutes by default. If you need to change the default value, contact your Nokia support representative.

If a JMS client with a durable subscription fails to subscribe, check whether there is an existing durable subscription and remove it.

5.6.3 Performance impact

Using the persistence message delivery mode and providing support for durable subscriptions can increase the storage overhead and decrease performance. The performance and scalability numbers are not yet available.

6 SNMP trap forwarding

6.1 SNMP trap forwarding overview

The 5529 OAD also supports the forwarding of alarms as SNMP traps to OSS client applications. The alarms that the 5529 OAD forwards as SNMP traps can originate in the 5520 AMS, the NEs, and the installed plug-in applications such as the 5529 OAD itself and the 5529 IDM.

The 5529 OAD supports forwarding alarms as SNMPv2 traps. The 5529 OAD SNMP trap forwarder sends alarm traps as UDP packets.

The SNMP trap forwarding has the following limitations:

- no guarantee of alarm information delivery
- no buffering in case of loss of communication between the OSS and the 5529 OAD
- no ability to retrieve a list of current active alarms
- no capability to acknowledge alarms
- the sysUpTime attribute has a maximum value of 4 294 967 295 (497 days). After this maximum value is reached, the value will restart at zero.

However, the SNMP trap forwarding is an alarm dispatching solution that suits small networks with low alarm rates.

6.2 Configuration

The 5529 OAD can send alarm traps to multiple SNMP trap targets. You need to enter the parameters for each SNMP trap target, and, if required, configure the SNMP trap filtering using the OAD NBI Settings window on the 5520 AMS GUI, as described in the *5529 OAD Installation and Administration Guide*.

6.2.1 Parameters

For each SNMP trap target, you need to configure the following parameters:

- the port number to which the 5529 OAD sends the trap
- the IP address of the target
- the community name to be sent with the trap

If any of the parameters that need to be entered for an SNMP trap target are missing, the 5529 OAD does not forward traps to that target. These parameters are set using the OAD NBI Settings page on the 5520 AMS GUI, as described in the *5529 OAD Installation and Administration Guide*.

The 5529 OAD logs any invalid entries for an SNMP target when the SNMP trap forwarder configuration is saved. A configuration entry is invalid if some of the parameters for an SNMP trap target are not configured.

6.2.2 SNMP trap filtering

The 5529 OAD supports the basic filtering of SNMP traps. You can configure the SNMP trap filtering parameters in the 5520 AMS GUI. The following SNMP trap filtering parameters are available for configuration:

- Last Acknowledged User ID
- EMS Domain
- Acknowledged
- Specific Problem
- Repair Actions
- Alarm Severity
- Asam Alarm ServAffType
- Alarm Category
- Alarm Domain
- Probable Cause

For more information, see the *5529 OAD Installation and Administration Guide*.

6.3 SNMP client application

You need to create SNMP client applications to receive SNMP alarm traps, and to display the alarm information in a readable format that identifies the alarm attributes and their values. You need to integrate the 5529 OAD MIBs into your client application to correctly interpret the alarm information contained in the SNMP traps.

6.4 MIBs

The proprietary OAD-AMS-ALARM MIB has been developed to assist your SNMP client applications in processing SNMP alarm traps. The MIB defines all the alarm attributes that are present in alarm traps.

The OAD-AMS-ALARM MIB and other supporting MIBs (for example, RFC-1212, RFC1155-SMI, SNMPv2-COMMUNITY-MIB, SNMPv2-MIB, SNMPv2-SMI, and SNMPv2-TC) are supplied with the 5529 OAD load. After installing the 5529 OAD, you can find the MIBs on the 5520 AMS server in the following location:

$AMS_LOCAL_DATA_DIR/ams/local/repository/app-oad-*version_release-revision*/docs

where *version_release-revision* is the version, release, and revision of the 5529 OAD software. An example of *version_release-revision* is 1.0_9.6.07-999000.

The 5529 OAD provides a harmonized alarm stream for an access network that includes multiple NE types, and the alarm trap format is based on standard MIBs. The forwarding of alarms as traps is independent of the supported 5529 OAD NE plug-ins. As a result, you do not need to load MIBs for each NE type.

Note — The alarmId attribute in the 5529 OAD MIBs is a unique identifier that refers to an individual alarm; for example, multiple alarms of the same type have multiple alarmId attributes.

6.5 SNMP trap example

The following is an SNMP trap example.

```
2012-09-14 11:38:48 ubuntu101 [UDP:
[192.168.95.101]:56877->[192.168.95.102]]:

SNMPv2-MIB::sysUpTime.0 = Timeticks: (1019122) 2:49:51.22

SNMPv2-MIB::snmpTrapOID.0 = OID:

OAD-AMS-ALARM-MIB::amsAlarmMIBNotifications.0.1

OAD-AMS-ALARM-MIB::oadAmsAlarmMib.1.4.0 = STRING: "oadadm"

OAD-AMS-ALARM-MIB::specificProblem = STRING: Trap Manager Registration
Failure

OAD-AMS-ALARM-MIB::repairActions = STRING: "Launch the command Resync Alarms
and State for that NE. If the problem persists, stop and restart supervision
of the NE."

OAD-AMS-ALARM-MIB::emsDomain = STRING: "AMS"

OAD-AMS-ALARM-MIB::sourceFriendlyName = STRING: Agent:GPON48:IACM

OAD-AMS-ALARM-MIB::alarmSeverity = INTEGER: major(4)

OAD-AMS-ALARM-MIB::notificationId = STRING: 80015

OAD-AMS-ALARM-MIB::alarmCategory = INTEGER: equipment(4)

OAD-AMS-ALARM-MIB::asamAlarmServAffType = INTEGER: notServiceAffecting(2)
```

```
OAD-AMS-ALARM-MIB::alarmDomain = STRING: NE

OAD-AMS-ALARM-MIB::filterId = INTEGER: -1

OAD-AMS-ALARM-MIB::raisedTimeStamp = STRING: Fri Sep 14 11:38:48 ICT 2012

OAD-AMS-ALARM-MIB::acknowledged = INTEGER: false(2)

OAD-AMS-ALARM-MIB::probableCause = STRING: Synchronization Problem

OAD-AMS-ALARM-MIB::neIpAddress = STRING: 172.21.222.101
```

7 Reference

7.1 Purpose

This chapter provides reference information to support the 5529 OAD developer tasks.

7.2 MTOSI objects

The SOAP body defines each MTOSI name as a sequence of name components, where each component is presented in a relative position with respect to the parent element. Each element name corresponds to one of the components of the hierarchical name, and the element value contains the value of the specified hierarchical name component.

All of the alarm events that are generated by the 5520 AMS or 5523 AWS are mapped to their closest parent objects on the node. The supported parent objects are node, rack, shelf, slot, card, and port. These objects are mapped in MTOSI as listed in Table 8.

A 5520 AMS alarm that fails to map to the corresponding parent object is mapped to the node level.

7.3 Alarm encoding

You can use alarm encoding to compress alarm information. Each alarm XML string is sent as a compressed byte array. If a message is encoded, the JMS message type is “Bytes”. For information about configuring alarm encoding, see the *5529 OAD Installation and Administration Guide*.

Alarm encoding is compatible with JMS filtering because only the JMS payload is encoded. JMS filtering is based on the header. You can use both alarm encoding and alarm grouping. However, alarm grouping is not compatible with JMS filtering; therefore, you will not be able to use subscriber filters if you use both encoding and grouping. See section 5.5 for more information about JMS filtering.

The sample OSS client is compatible with any combination of grouping and encoding. No code changes are required if the grouping or encoding parameters are changed.

7.4 Alarm information mapping

Table 18 lists the mapping of 5520 AMS alarm field names in the Alarm perspective to 5529 OAD attributes in SOAP XML responses. See also Table 20 for information about attribute types and values.

Table 18 **5520 AMS alarm field mapping to 5529 OAD SOAP XML attributes**

5520 AMS alarm field	5529 OAD SOAP XML attribute
Acknowledged	acknowledgeIndication
Acknowledged by	lastAcknowledgedUserId
Acknowledged on	lastAcknowledgedTimeStamp
Acknowledged via IP Address	lastAcknowledgedInetAddress
Additional Info/Text	additionalText
Assigned by	— [1]
Assigned on	— [1]
Assigned to	userId
Assigned via IP Address	lastChangedUserIdInetAddress
Category	X733_EventType
Cleared Time	clearedTimeStamp
Domain	alarmDomain
Event Time	neTime
Filter Id	filterId
Last Modification Time	lastModificationTimeStamp
Mnemonic	probableCauseMnemonic
NE Alarm Type	neAlarmType
NE Group	neGroup
NE IP Address	neIpAddress
NE Name	objectName

(1 of 2)

5520 AMS alarm field	5529 OAD SOAP XML attribute
Notes	notes
Probable Cause	nativeProbableCause
Proposed Repair Actions	proposedRepairAction
Service Affecting	serviceAffecting
Severity	perceivedSeverity
Source Name	aliasValue of the alias, within the aliasList, with SOURCE_FRIENDLY_NAME or REPORTING_FOR as the aliasName
Specific Problem	specificProblem
Threshold	filterThreshold
TL1 cause	tl1Cause
Filter Type	filterType
Window	filterWindow

(2 of 2)

Note

[1] The 5529 OAD SOAP XML message does not contain an equivalent attribute.

See the Release Notice for the NE support plug-in for information about the mapping of:

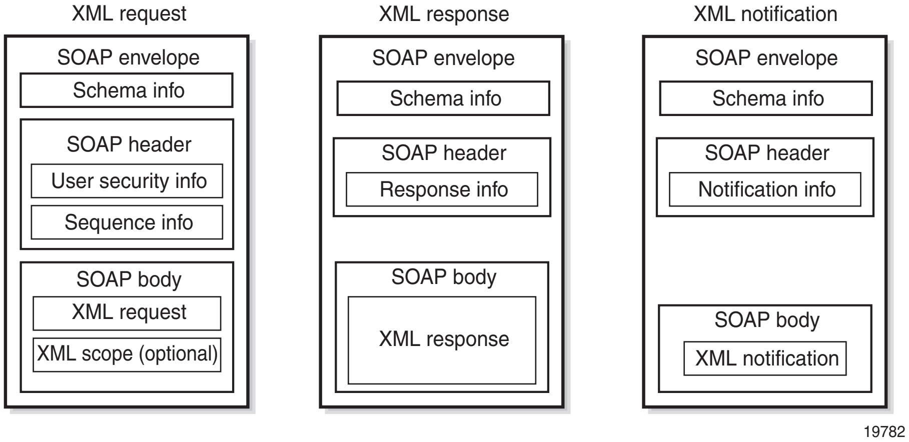
- 5529 OAD native probable cause, specific problem, and MTOSI probable cause attributes to 5520 AMS alarm IDs
- alarm IDs to the 5529 OAD native probable cause, specific problem, and MTOSI probable cause attributes

7.5 SOAP envelope components

Note — In this document, the tables of SOAP envelope body elements may not list the elements in the same order as they are listed in the actual SOAP XML requests and responses in the OSS. See the schema documentation for the correct order of the elements.

Figure 1 shows the SOAP envelope types and their content.

Figure 1 **SOAP envelope types**



Note — If alarm grouping is used, the SOAP header is empty. The SOAP body contains one or more units; each unit contains a header and body pair.

7.5.1 MTOSI header

Table 19 describes the most commonly used attributes in a standard SOAP envelope header supported by the 5529 OAD, and indicates the attribute relevance to each type of SOAP envelope (request, response, and notification). See the TMF MTOSI documentation for information about all SOAP header attributes defined in MTOSI.

Table 19 **MTOSI header attributes**

Attributes	Description	Req	Resp	Notif	Values
activityName	Activity (operation or notification) name	Yes	Yes	Yes	*Supported operation name*; for example, getActiveAlarmsCount
msgName	Message name in the WSDL file	Yes	Yes	Yes	*Supported operation name*; for example, getActiveAlarmsCount
msgType	Message type	Yes	Yes	Yes	REQUEST RESPONSE NOTIFICATION ERROR
senderURI	Application sending the message	Yes	Yes	Yes	Character string
destinationURI	Destination for the message	Yes	Yes	Yes	Character string (request/response) topic/Fault (notification)

(1 of 2)

Attributes	Description	Req	Resp	Notif	Values
activityStatus	Status of the response operation	No	Yes	No	SUCCESS FAILURE WARNING [1]
correlationId	Identifier to correlate request and response message	Yes	Yes	No	0 1
communicationPattern	Sets the communication pattern	Yes	Yes	Yes	SimpleResponse MultipleBatchResponse Notification
communicationStyle	Sets the communication style	Yes	Yes	Yes	RPC MSG
requestedBatchSize	Logical size of the batch for a multi-response communication pattern	Yes	Yes	No	Integer
batchSequenceNumber	Identifies the batch sequence number in a multiple response communication pattern	No	Yes	No	Integer
batchSequenceEndOfReply	Indicates the end of a batch response when the value is true	No	Yes	No	true false
iteratorReferenceURI	Describes the data retrieval iterator	No	Yes	No	Yes
timestamp	Date and time when the message was created	Yes	Yes	Yes	Date and time in MTOSI UTC format: *yyyyMMddHHmmss*.*S*[Z|{+|-}*HHMm*] See section 3.5.1 for more information about the MTOSI UTC format.

(2 of 2)

Note

[1] WARNING is not supported.

7.5.2 MTOSI body attributes in alarm response and notification

Table 20 describes the supported attributes in a standard SOAP envelope body supported by the 5529 OAD for alarm response and notification messages.

Table 20 **MTOSI body attributes for alarm response and notification**

Attribute	Description	Type or value
eventInfo	Container element for the information associated with the event	EventInformation_T
notificationId	Event notification identifier	Unique value consisting of a prefix and a number (5520 AMS alarm ID) separated by a colon (see Table 16)

(1 of 5)

Attribute	Description	Type or value
objectName	Full distinguished name of the object reporting the notification The objectName reflects the name of the shelf and not that of the rack when an alarm is raised on the shelf (subrack).	NamingAttributes_T Character string
objectType	Type of the object identified by the objectName. See Table 8.	ObjectType_T
osTime	Date and time at which the event was reported by the OS system	ITU-Time_T Date and time in MTOSI UTC format: *yyyyMMddHHmmss.S*[Z|{+|-}*HHMm*] See section 3.5.1 for more information about the MTOSI UTC format.
neTime	Date and time at which the alarm was raised	ITU-Time_T Date and time in MTOSI UTC format: *yyyyMMddHHmmss.S*[Z|{+|-}*HHMm*] See section 3.5.1 for more information about the MTOSI UTC format.
isClearable	Boolean attribute indicating if the alarm or event is clearable	true or false The default is false.
aliasNameList	Container element for a list of aliases for the network entity	AliasNameList_T
alias	Container element for the following attributes: • aliasName • aliasValue	—
aliasName	Alias name of the network entity	AliasName_T The values are: • SOURCE_FRIENDLY_NAME (if the objectName is the name of the object impacted by the alarm) (1) • REPORTING_FOR (if the objectName is the name of a parent of the object impacted by the alarm) (1) • NE_ALIAS_NAME
aliasValue	Alias value of the network entity	AliasValue_T The values are: • Source-friendly name of the object impacted by the alarm • Configured alias name of the NE that contains the object impacted by the alarm
layerRate	Layer to which the alarm or event is relevant	LayerRate_T The default is PROP_UNKNOWN
probableCause	Probable cause of the error. Container element for the following attributes: • ru • contra • type	ProbableCause_T

(2 of 5)

Attribute	Description	Type or value
ru	Boolean attribute indicating the location of the network resource to which the probable cause is associated	The options are true or false. The value is always false.
contra	Boolean attribute used to distinguish the direction of the signal-related alarm	The options are true or false. The value is always false.
type	Probable cause specific value	ProbableCause_T
nativeProbableCause	Probable cause as indicated on the OS interface	Character string
additionalText	More information about the alarm (for example, “Unit is mismounted”). The attribute is for display only and should not be used by the OSS to derive a logical decision.	Character string
perceivedSeverity	Severity of the alarm or event A value of PROP_UPDATE is provided for the perceivedSeverity attribute if the Show Alarm Details in Alarm Change Notifications parameter of the 5529 OAD NBI settings is enabled and an alarm change event occurs. See the *5529 OAD Installation and Administration Guide* for information about 5529 OAD NBI settings.	PerceivedSeverity_T The values are: • PS_INDETERMINATE • PS_CRITICAL • PS_MAJOR • PS_MINOR • PS_WARNING • PS_CLEARED • PROP_UPDATE (notification of an update to an existing alarm)
serviceAffecting	Indicates whether the alarm has affected service	ServiceAffecting_T The values are: • SA_UNKNOWN • SA_SERVICE_AFFECTING • SA_NON_SERVICE_AFFECTING
rcaiIndicator	Boolean attribute providing a root cause alarm indication	The options are true or false. The value is always false.
acknowledgeIndication	Acknowledge indication state	AcknowledgeIndication_T The values are: • AI_EVENT_UNACKNOWLEDGED • AI_EVENT_ACKNOWLEDGED
X733_EventType [(2)]	One of the five basic categories specified in ITU-T X.733	The values are: • communicationsAlarm • environmentalAlarm • equipmentAlarm • processingErrorAlarm • qualityofServiceAlarm
X733_SpecificProblems [(2)]	Further refinements to the probable cause of the alarm	SpecificProblemList_T
X733_ProposedRepairActions [(2)]	Used if the cause is known and the system that is being managed can suggest one or more solutions	ProposedRepairActionList_T
vendorExtensions	Specific container element for vendor extensions of the alarm response or notification object	AlarmExt_T [(3)]

(3 of 5)

Attribute	Description	Type or value
clearedTimeStamp	Date and time at which the alarm was cleared	ITU-Time_T Date and time in MTOSI UTC format: *yyyyMMddHHmmss.S*[Z|{+|-}*HHMm*] See section 3.5.1 for more information about the MTOSI UTC format.
lastModificationTimeStamp	Date and time at which the alarm was last modified.	ITU-Time_T Date and time in MTOSI UTC format: *yyyyMMddHHmmss.S*[Z|{+|-}*HHMm*] See section 3.5.1 for more information about the MTOSI UTC format.
filterId	The ID for spatial and/or temporal alarms as configured in the 5520 AMS	Character string
filterThreshold	The threshold value for temporal or spatial alarms as configured in the 5520 AMS	Character string
filterType	Type of the alarm (spatial or temporal)	Character string The values are: • spatial • temporal
filterWindow	Time duration for temporal or spatial alarms as configured in the 5520 AMS	Character string
userId	User ID of the user to whom the alarm is assigned	Character string
notes	Notes added by the 5520 AMS user to provide additional information	Character string
alarmDomain	Domain name of the alarm	Character string
neAlarmType	The NE alarm type and ID	Character string
neGroup	The name of the group that the NE on which the alarm is raised belongs to	Character string
neType	The type of the NE on which the alarm is raised	Character string
probableCauseMnemonic	The mnemonic used to indicate the alarm probable cause type	Character string
lastAcknowledgedTimeStamp	Date and time at which the alarm was last acknowledged	ITU-Time_T Date and time in MTOSI UTC format: *yyyyMMddHHmmss.S*[Z|{+|-}*HHMm*] See section 3.5.1 for more information about the MTOSI UTC format.
lastAcknowledgedUserId	User ID of the user who acknowledged the alarm	Character string
lastAcknowledgedInetAddress	IP address of the 5520 AMS client used by the user who acknowledged the alarm	IP address
lastChangedUserIdInetAddress	IP address of the 5520 AMS client used by the user who assigned the alarm	IP address
tl1Cause	TL1 Alarm Condition of the alarm	Character string

(4 of 5)

Attribute	Description	Type or value
neIpAddress	IP address of the NE on which the alarm is raised, modified, or cleared.	IP address The IPv4 and IPv6 address formats are supported.

(5 of 5)

Notes

[1] This attribute represents the alarm source name. The format is described in the *5520 AMS and 5529 Enhanced Applications Alarm Search Tool*.

[2] The X733_EventType attribute is not provided when the category of the alarm received by the 5529 OAD from the 5520 AMS is specified in ITU-T X.736. The X733_SpecificProblems and X733_ProposedRepairActions attributes are provided when the alarm category is specified in ITU-T X.736.

[3] MTOSI leaves the type of vendorExtensions attribute for an alarm open; that is, it does not provide a specific type. An OSS client application that implements this MTOSI attribute must specify a vendorExtensions type. The 5529 OAD uses the PackageOfNamedTypeAndValues_T type as the alarm vendorExtensions type. Any OSS client application that specifies the vendorExtensions attribute must implement the correct attribute type to be able to process it. The MTOSI alarm vendorExtensions attribute is of type AlarmExt_T (that is, of any type), but the actual type is PackageOfNamedTypeAndValues_T, as specified in the alu.v1 namespace.

7.5.3 MTOSI body attributes for event notification

Table 21 describes the attributes present in the MTOSI body for event notification.

Table 21 **MTOSI body attributes for event notifications**

Attribute	Description	Heartbeat	EventLossOccured	EventLossCleared
notificationId	Event notification identifier Unique value consisting of a prefix and a number separated by a colon (see Table 16)	Yes	Yes	Yes
objectType	OT_OS (see Table 8)	Yes	Yes	Yes
objectName	Container element for the object name	Yes	Yes	Yes
osNm	Full distinguished name of the object reporting the notification. The value is a string following the naming conventions defined in NamingAttributes_T.	Yes	Yes	Yes
osTime	Date and time that the heartbeat notification was sent [1]	Yes	No	No
startTime	Date and time that the event loss occurred [1]	No	Yes	No
endTime	Date and time that the event loss was cleared [1]	No	No	Yes

Note

[1] The date and time in MTOSI UTC format is *yyyyMMddHHmmss.S*[Z|{+|-}*HHMm*]; see ITU-Time_T.

7.5.4 getSystemInfo SOAP envelope

Table 22 describes the elements used in the getSystemInfo SOAP envelope header, and indicates the element relevance to each type of SOAP XML message (request or response).

Table 22 **getSystemInfo SOAP envelope header**

Element	Description	Req	Resp	Values
activityName	Activity (operation) name	Yes	Yes	getSystemInfo
msgName	Message name in the WSDL file	Yes	Yes	getSystemInfo
msgType	Message type	Yes	Yes	REQUEST RESPONSE ERROR
senderURI	Application sending the message	Yes	Yes	Alphanumeric string
destinationURI	Destination of the message	Yes	Yes	Alphanumeric string
activityStatus	Status of the response operation	No	Yes	SUCCESS FAILURE WARNING
communicationPattern	Message communication pattern	Yes	Yes	SimpleResponse
communicationStyle	Message communication style	Yes	Yes	RPC
timestamp	Date and time when the message was created	Yes	Yes	Date and time format: *dd*/*MM*/*yyyy HH*:*mm*:*ss* (not MTOSI UTC format) where *dd* is the day *MM* is the month *yyyy* is the year *HH* is the hour *mm* is the minute *ss* is the second

Table 23 describes the elements used in the SOAP envelope body of a getSystemInfo response message.

The SOAP envelope body of a getSystemInfo request message contains only the operation name, and no attributes.

Note — Table 23 lists only the elements that are supported by the 5529 OAD for the operation. There may be additional standard elements that are displayed in the WSDL file, but those elements are not used nor supported by the 5529 OAD.

Table 23 getSystemInfo SOAP XML response elements

Element	Description	Value
getSystemInfoResponse	Container element for the system information	—
systemTime	The current system date and time on the local host (the host that serves the request)	Date and time in MTOSI UTC format: *yyyyMMddHHmmss.S*[Z|{+|-}*HHMm*] See section 3.5.1 for more information about the MTOSI UTC format.
versionInfo	Container element for application version information	—
version	5529 OAD application version	Alphanumeric string; for example, 9.6.07
timestamp	The date and time that the 5529 OAD software load was produced	Date and time in MTOSI UTC format: *yyyyMMddHHmmss.S*[Z|{+|-}*HHMm*] See section 3.5.1 for more information about the MTOSI UTC format.
vendorExtensions	Container element for vendor extensions of the getSystemInfoResponse message	—
package	Container element for the master application server information	—
NameAndStringValue	Container element for the IP address of the master application server	—
tmf854:name	Element indicating that this information is about the master application server	masterServer
tmf854:value	IP address of the master application server	IP address in the following format: four decimal fields separated by periods

7.5.5 getSystemHealthInfo SOAP envelope

Table 24 describes the elements used in the getSystemHealthInfo SOAP envelope header, and indicates the element relevance to each type of SOAP XML message (request or response).

Table 24 getSystemHealthInfo SOAP envelope header

Element	Description	Req	Resp	Values
activityName	Activity (operation) name	Yes	Yes	getSystemHealthInfo
msgName	Message name in the WSDL file	Yes	No	getSystemHealthInfo
		No	Yes	getSystemHealthInfoResponse
msgType	Message type	Yes	Yes	REQUEST RESPONSE ERROR
senderURI	Application sending the message	Yes	Yes	Alphanumeric string
destinationURI	Destination of the message	Yes	Yes	Alphanumeric string

(1 of 2)

Element	Description	Req	Resp	Values
activityStatus	Status of the response operation	No	Yes	SUCCESS FAILURE WARNING
communicationPattern	Message communication pattern	Yes	Yes	SimpleResponse
communicationStyle	Message communication style	Yes	Yes	RPC
timestamp	Date and time when the message was created	Yes	Yes	Date and time in MTOSI UTC format: *yyyyMMddHHmmss.S*[Z|{+|-}*HHMm*] See section 3.5.1 for more information about the MTOSI UTC format.

(2 of 2)

Table 25 describes the elements used in the SOAP envelope body of a getSystemHealthInfo response message.

The SOAP envelope body of a getSystemHealthInfo request message contains only the operation name, and no attributes.

Table 25 **getSystemHealthInfo SOAP XML response elements**

Element	Description	Value
getSystemHealthInfoResponse	The XML namespace; for example, xmlns=“alu.v1”	Alphanumeric string

7.6 Event timestamp

The OSS client applications get the timestamp of various events from the following sources:

- osTime attribute of the Alarm and Heartbeat event notifications
- startTime attribute of the EventLossOccured event notification
- endTime attribute of the EventLossCleared event notification

Customer document and product support

Customer documentation

Customer Documentation Welcome Page

Technical Support

Product Support Portal

Documentation feedback

Customer Documentation Feedback