



5529 Inventory Data Manager, Release 9.6.07

Northbound Interface Guide

3JL-03001-BRAA-PWZZA

Edition 01

August 2018

Nokia is a registered trademark of Nokia Corporation. Other products and company names mentioned herein may be trademarks or tradenames of their respective owners.

The information presented is subject to change without notice. No responsibility is assumed for inaccuracies contained herein.

© 2018 Nokia.

Contains proprietary/trade secret information which is the property of Nokia and must not be made available to, or copied or used by anyone outside Nokia without its written authorization. Not to be used or disclosed except in accordance with applicable agreements.

Table of contents

1	Preface	13
1.1	Related documentation.....	13
1.2	Conventions used in this guide.....	14
1.2.1	Important information.....	14
1.2.2	Procedures with options or substeps.....	15
1.3	Multiple PDF file search.....	16
Getting started		
2	What's new	19
2.1	What's new in Release 9.6.07	19
2.2	What's new in Release 9.6.05	20
2.3	What's new in Release 9.6.03	21
3	Getting started	23
3.1	General.....	23
3.2	Functionality	23
3.3	Support files.....	24
3.4	Technologies and standards	26
3.4.1	Date and time in MTOSI UTC format	26
5529 IDM Northbound interface		
4	Operations	31
4.1	General.....	31
4.2	Inventory data retrieval.....	31
4.2.1	Web services	32
4.2.2	HTTPS interface	32
4.2.3	Information model.....	32
4.3	MTOSI object names.....	33
4.4	Supported operations	35
4.4.1	exportNetwork	36
4.4.1.1	Specifying a target list of NEs.....	39
4.4.2	getAllManagedElementNames	39
4.4.3	getEquipment	39
4.4.4	getExportedObject.....	40
4.4.5	getInventory.....	40
4.4.5.1	Scoping a getInventory request.....	41
4.4.6	getInventoryIterator	42
4.4.7	getManagedElement	43
4.4.8	getSystemHealthInfo	43
4.4.9	getSystemInfo.....	43
4.4.10	getTP	43
4.4.11	query.....	44
4.4.11.1	Filtering a query request.....	44
4.4.12	queryIterator	45
4.5	Service endpoints	46
4.5.1	Verifying the 5529 IDM service endpoints	47

4.6	Exceptions	47
4.7	Iterative operations in cluster deployments	48
5	JMS notifications	49
5.1	NE inventory updates	49
5.2	Event notifications	50
5.3	Information model	53
5.3.1	SOAP XML messages	54
5.4	JMS client application guidelines	54
5.4.1	System time	54
5.4.2	Properties	54
5.4.3	JMS client application tasks	55
5.4.4	JMS ports	55
5.4.5	JMS application parameters	55
5.4.6	Libraries	56
5.5	Subscription filter	57
5.5.1	JMS property fields	57
5.5.2	MTOSI_EventType values	57
5.5.3	Filter content	58
5.5.3.1	Filter examples	58
5.6	Persistence and durable subscriptions	58
5.6.1	Persistence overview	59
5.6.2	Durable subscriptions	59
5.6.3	Performance impact	60
6	Reference	61
6.1	Purpose	61
6.2	Northbound objects	61
6.2.1	Inventory data retrieval	62
6.2.2	Specifying the value for the meNm attribute in the object FDN	62
6.3	SOAP envelope components	63
6.4	getAllManagedElementNames SOAP envelope	63
6.4.1	Header	64
6.4.2	Body	64
6.5	getExportedObject SOAP envelope	65
6.5.1	Header	65
6.5.2	Body	66
6.6	getInventory SOAP envelope	68
6.6.1	Header	68
6.6.2	Body	70
6.7	getSystemHealthInfo SOAP envelope	71
6.7.1	Header	71
6.7.2	Body	71
6.8	getSystemInfo SOAP envelope	72
6.8.1	Header	72
6.8.2	Body	73
6.9	exportNetwork SOAP envelope	74
6.9.1	Header	74
6.9.2	Body	75
6.10	exportNetwork output file example	76

6.11	query SOAP envelope	81
6.11.1	Header	81
6.11.2	Body	82
6.12	JMS notification SOAP envelope.....	84
6.12.1	Header	85
6.12.2	Body	85
6.13	SOAP envelopes for shortcut operations	87
6.13.1	Header	87
6.13.2	Body	88

List of figures

6	Reference	61
Figure 1	SOAP envelope types	63

List of tables

1	Preface	13
Table 1	Related documentation.....	13
Table 2	Documentation conventions	14
2	What's new	19
Table 3	What's new in Release 9.6.07	19
Table 4	What's new in Release 9.6.05	20
Table 5	What's new in Release 9.6.03	21
3	Getting started	23
Table 6	WSDL locations within the schema files.....	25
4	Operations	31
Table 7	SOAP envelope parent elements	33
Table 8	Object mapping	33
Table 9	Inventory data retrieval operations	35
Table 10	Scope attributes for getInventory.....	41
Table 11	Filtering level and source combinations	45
Table 12	Attribute combination support.....	45
Table 13	5529 IDM service endpoints	46
Table 14	Exceptions	47
5	JMS notifications	49
Table 15	Event description	50
Table 16	Event notification SOAP envelope header	50
Table 17	Card insertion and removal event SOAP envelope body	51
Table 18	ONT creation and deletion event SOAP envelope body	51
Table 19	Successful and failed network export SOAP envelope body.....	52
Table 20	Heartbeat SOAP envelope body	53
Table 21	Subscription filter examples	58
6	Reference	61
Table 22	getAllManagedElementNames SOAP envelope header	64
Table 23	getAllManagedElementNames SOAP XML request elements	65
Table 24	getAllManagedElementNames SOAP XML response elements	65
Table 25	getExportedObject SOAP envelope header	65
Table 26	getExportedObject SOAP XML request elements.....	66
Table 27	getExportedObject SOAP XML response elements	67
Table 28	getInventory SOAP envelope header	69
Table 29	getInventory SOAP XML request elements.....	70
Table 30	getInventoryResponse SOAP XML response elements.....	70
Table 31	getSystemHealthInfo SOAP envelope header	71
Table 32	getSystemHealthInfo SOAP XML response element	72
Table 33	getSystemInfo SOAP envelope header.....	72
Table 34	getSystemInfo SOAP XML response elements.....	73
Table 35	exportNetwork SOAP envelope header.....	74
Table 36	exportNetwork SOAP XML request elements	75
Table 37	exportNetwork SOAP XML response elements.....	76

Table 38	Objects and attributes selected for export for ISAM/FTTB/FTTN	77
Table 39	query SOAP envelope header	81
Table 40	query SOAP XML request elements	82
Table 41	query SOAP XML response elements	84
Table 42	JMS notification SOAP envelope header	85
Table 43	JMS notification SOAP envelope body	85
Table 44	JMS notification SOAP envelope body for exportNetwork	86
Table 45	JMS notification SOAP envelope body for heartbeat	87
Table 46	SOAP envelope header for shortcut operations	87
Table 47	SOAP XML request elements for shortcut operations	88
Table 48	SOAP XML response elements for shortcut operations	89

List of procedures

1	Preface	13
Procedure 1	Example of options in a procedure	15
Procedure 2	Example of substeps in a procedure	15
Procedure 3	To search multiple PDF files for a term	16

1 Preface

The *5529 IDM Northbound Interface Guide* contains information about:

- inventory data retrieval operations
- JMS notifications
- OSS client applications
- NBI objects
- SOAP envelope components

1.1 Related documentation

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

Table 1 **Related documentation**

Customer documentation	Description
5520 Access Management System	
<i>5520 AMS Administrator Guide</i>	Information about administrative functions, including management of client-server communication, users, NE communication, and schedules
<i>5520 AMS Northbound Interface Guide</i>	Information about functions that are supported by the 5520 AMS NBI
<i>5520 AMS User Guide</i>	Information about user functions, including monitoring, fault, and alarm management and performing tasks that are common to the NEs
5520 AMS and 5529 Enhanced Applications	
<i>5520 AMS and 5529 Enhanced Applications Alarm Search Tool</i>	5520 AMS, 5529 Enhanced Applications, and NE alarm descriptions
<i>5520 AMS and 5529 Enhanced Applications Privacy Considerations</i>	Information about the product features that impact privacy and the measures taken to protect such data
<i>5520 AMS Solution Glossary</i>	Terms and acronyms related to the 5520 AMS and 5529 Enhanced Applications
<i>5520 AMS Solution Planning Guide</i>	Information about the system requirements for the installation of the 5520 AMS server and client
5529 Enhanced Applications	
<i>5529 Enhanced Applications Release Notice</i>	Information about updates to the product, software and documentation delivery, known restrictions, and fixed issues
5529 Inventory Data Manager	
<i>5529 IDM Installation, Administration, and User Guide</i>	Information about how to install and configure the 5529 IDM, and perform inventory data management tasks

(1 of 2)

Customer documentation	Description
5529 IDM NE support plug-in attributes guides	Information about the full set of NE object attributes that is supported by the 5529 IDM
HTML files that are provided with the 5529 IDM	Information about the subset of NE object attributes that is supported by the NBI. See the <i>5529 Enhanced Applications Release Notice</i> for information about how to access these files.

(2 of 2)

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
<i>Italics</i>	Identify a variable	<i>hostname</i>
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 task or situation may, or will, cause equipment damage or serious performance problems.



Caution — Caution indicates that the described task 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](#)

[3 Getting started](#)

2 What's new

2.1 What's new in Release 9.6.07

2.2 What's new in Release 9.6.05

2.3 What's new in Release 9.6.03

2.1 What's new in Release 9.6.07

Table 3 describes the 5529 IDM features and enhancements added to the *5529 IDM 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			
NEs with IPv6 address management	<p>The 5529 IDM NBI is updated to support NEs that have IPv6 address management. The following items are updated:</p> <ul style="list-style-type: none"> meNm attribute in the object FDN for SOAP XML request and response operations meNm scope attribute in getInventory operation meNm element in card insertion and removal events meNm element in ONT creation and deletion events JMS notifications (meNm attribute in objectName element) 	01	<p>Section 6.2.2</p> <p>Tables 8, 10, 17, 18, 24, 26, 27, 36, 40, 41, 43, and 47</p>
Parent FN attribute available for network export	<p>The Parent FN attribute for the following ISAM objects is available for network export:</p> <ul style="list-style-type: none"> GPON Current MAC Address GPON VLAN Association DHCP Session 	01	<p>Section 4.4.1 for information about the exportNetwork operation</p> <p><i>5529 IDM Installation, Administration, and User Guide</i> for more information about the Parent FN attribute and export profiles</p>
Parent FN attribute returned by getExportedObject and query operations	<p>The Parent FN attribute for the following ISAM objects is returned by the getExportedObject and query operations:</p> <ul style="list-style-type: none"> GPON Current MAC Address GPON VLAN Association DHCP Session <p>The NBI attribute name for the Parent FN attribute for each object is gponL2VlanPortVlanCurrentMac_parentFN and gponL2DhcpSession_parentFN.</p>	01	<p>Sections 4.4.4 and 4.4.11 for information about the getExportedObject and query operations</p>
Documentation changes			
<i>5520 AMS and 5529 Enhanced Applications Privacy Considerations</i>	Added the <i>5520 AMS and 5529 Enhanced Applications Privacy Considerations</i> document to the list of related documentation sources	01	Table 1

(1 of 2)

Feature/enhancement	Description	Edition	See
Alarm event notifications	Added a note about how the 5529 IDM 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
exportNetwork output file example	Updated the information about the exportNetwork output file example	01	Section 6.10
getInventory operation	The exception that is received when the getInventory operation request contains the unsupported value of BASE_OBJECT for the level attribute is changed from EXCPT_INVALID_INPUT to EXCPT_NOT_IMPLEMENTED.	01	Tables 10 and 14
JMS application parameters	Added the following missing information: <ul style="list-style-type: none"> 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
query operation response	Added a note that indicates that if you modify the objects in the Collected Objects in Inventory view in the 5529 IDM GUI and run the NBI query operation on the objects, the query operation response will not return any data until the changes are taken into account in the next inventory collection	01	Section 4.4.11
Request ID for the exportNetwork operation	The value for the requestId element in the exportNetwork operation response is modified to be unique within a 5520 AMS cluster.	01	Section 4.4.1 Table 37
SOAP version 1.2	Added SOAP version 1.2 to the list of supported technologies and standards that allow the 5529 IDM to easily integrate with OSS client applications. SOAP 1.2 is supported on 5529 IDM R9.6 or later.	01	Section 3.4

(2 of 2)

2.2 What's new in Release 9.6.05

Table [4](#) describes the 5529 IDM features and enhancements added to the *5529 IDM 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			
Display format of Link object name in exportNetwork output file	The display format of Link object names in the exportNetwork output file has changed. In R9.6.03 and earlier, the elements that make up the object name were separated by a space character, but in R9.6.05 or later the object name elements are separated by a colon (:).	01	Section 4.4.1 describes the output file
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 IDM NBI server	01	Tables 6 , 9 , and 13 Sections 4.4.8 and 6.7

(1 of 2)

Feature/enhancement	Description	Edition	See
Documentation changes			
Display format of attribute values in query results and exportNetwork output file	Added information about how an attribute value that is displayed in the exportNetwork output file may not match the attribute value that is displayed in the Inventory Query Results window	01	Section 4.4.1
getInventory operation	Updated the caution note for the getInventory operation to say that you need to use the operation only for individual queries. To perform a complete extraction of a large network, use the exportNetwork operation.	01	Section 4.4.5
query operation filtering examples	Removed the query operation filtering examples that were provided in Chapter 6	01	5529 IDM services page (.zip file that contains NBI operation samples)
Sample NBI operations	Added information about downloading a .zip file that contains samples of all of the 5529 IDM NBI request and response operations from the services page	01	Section 3.3

(2 of 2)

2.3 What's new in Release 9.6.03

Table [5](#) describes the 5529 IDM features and enhancements added to the *5529 IDM 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 3.3 , 4.2.2 , and 4.5.1
JBoss libraries for JMS client application	The name of the <code>idm-oss-client-1_release-version.jar</code> library is changed to <code>idm-oss-client-release-version.jar</code> . The <code>jboss-client.jar</code> library is removed. The following libraries are new: <ul style="list-style-type: none"> <code>axs-encryption-app-release-version.jar</code> <code>jboss-logging-3.3.0.Final.jar</code> <code>picketbox-4.9.6.Final.jar</code> <code>picketbox-infinispan-4.9.6.Final.jar</code> <code>slf4j-simple-1.7.21.jar</code> <code>wildfly-client-all.jar</code> <code>xbean-2.6.0.jar</code> 	01	Section 5.4.6
Login mechanism	Added a note stating that the 5520 AMS implements a rate-limiting login mechanism	01	Section 4.5
Documentation changes			
Exceptions	Removed EXCPT_NOT_FOUND, as it is not supported	01	Table 14

(1 of 2)

Feature/enhancement	Description	Edition	See
getInventory operation	Nokia recommends that OSS client applications use the query operation instead of the getInventory operation for filtered requests of inventory data.	01	Sections 4.4.5 and 4.4.11
getSystemInfo operation	Updated the value for the timestamp element	01	Table 33
HTTPS interface	Added information about the HTTPS interface and the required security certificate for the 5520 AMS server	01	Section 4.2.2
JBoss libraries for JMS client application	The <code>idm-oss-client-1_release-version.jar</code> 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 IDM OSS client .tar file.	01	Section 5.4.6
MTOSI object names	Removed information about ports where the SHDSL span has a -passive- value for the Customer ID attribute, and the port is part of a multi-wire configuration	01	Section 4.3
MTOSI UTC format	Added information about the MTOSI UTC format used to display a date and time in an NBI operation	01	Section 3.4.1 Tables 16 , 17 , 18 , 20 , 22 , 25 , 28 , 34 , 35 , 39 , 42 , 43 , 45 , and 46
NE inventory updates	Added the 7356 ISAM FTTB to list of cards that support notification for card insertion and removal	01	Section 5.1
Object mapping	Removed redundant Table 21, Northbound object names	01	Table 8
query and queryIterator operations	Added the query and queryIterator operations to the table of 5529 IDM service endpoints	01	Table 13
query filtering examples	Updated the query filtering examples to display the new NBI response header following the AXIS2 library upgrade	01	Section 6.12
Service endpoints	Combined the service endpoint information in a new section	01	Section 4.5
Supported version of Java	The supported version of Java is 1.8 as of 5529 IDM R9.6.	01	Section 3.4

(2 of 2)

3 Getting started

3.1 General

3.2 Functionality

3.3 Support files

3.4 Technologies and standards

3.1 General

The 5529 Inventory Data Manager is an application that integrates with the 5520 Access Management System. You can use the 5529 IDM to:

- collect inventory data from the NEs supervised by the 5520 AMS
- execute complex queries on the collected NE inventory data
- report inventory data to OSS client application through the NBI

This document covers only the NBI component of the 5529 IDM. See the *5529 IDM Installation, Administration, and User Guide* for information about collecting inventory data and performing queries using the 5520 AMS client.

You can use the 5529 IDM with NBI OSS client application only if you purchased the license to enable the NBI functionality. See the *5529 IDM Installation, Administration, and User Guide* for information about 5529 IDM licenses.

3.2 Functionality

The 5529 IDM NBI component facilitates the integration of inventory data reports into OSS client applications. The 5529 IDM simplifies the inventory data reporting, and allows the OSS client applications to request and receive inventory data about the NEs supported by the 5520 AMS.

The 5529 IDM 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 with current inventory data
- a JMS interface, which allows OSS client applications to receive inventory updates about the creation and deletion of cards and ONTs, and heartbeat event notifications by way of a JMS provider

3.3 Support files

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

- sample code for both HTTP/S and JMS client applications
- the 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 IDM WSDL and XSD schema source files to design the OSS client applications. The schema documentation for the activated 5529 IDM is available at the following URLs:

- <https://host:8443/idm/services>
- <http://host:8080/idm/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 [4.2.2](#) 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 username and password of a 5520 AMS user with the IDM NBI role. You can download the schema files individually or as a .zip file. 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 IDM NBI request and response operations.



Note — You cannot open multiple different NBI schema documentation pages that belong to different 5529 Enhanced Applications in the same web browser. If you attempt to do so, you will 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
getAllManagedElementNames getManagedElement getTP	ManagedElementMgrSOAP.wsdl	schema/tmf854/wsdl/binding/ soap_http/ManagedElementMgrSOAP.wsdl
getEquipment	EquipmentInventoryMgrSOAP.wsdl	schema/tmf854/wsdl/binding/ soap_http/EquipmentInventoryMgrSOAP.wsdl
getInventory getInventoryIterator getSystemHealthInfo getSystemInfo query queryIterator	InventoryRetrievalMgrExtnsSOAP.wsdl	schema/alu/wsdl/binding/soap_http/ InventoryRetrievalMgrExtnsSOAP.wsdl
exportNetwork getExportedObject	NetworkExportMgrSOAP.wsdl	schema/alu/wsdl/binding/soap_http/ NetworkExportMgrSOAP.wsdl

3.4 Technologies and standards

The 5529 IDM software architecture is built on open interoperable technologies such as SOAP and XML, the Java and J2EE framework, multi-tier layering, and web service interfaces. The use of standard interfaces allows the 5529 IDM 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 IDM, you need to be familiar with the following technologies and standards:

- XML 1.0
- XML Schema (XSD) 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)

See the following Web sites for more 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.ietf.org/rfc/rfc2616.txt> (HTTP/S 1.1)
- <http://www.tmfforum.org> (TMF and MTOSI)
- <http://www.oracle.com/technetwork/java/jms/index.html> (JMS 1.1)

3.4.1 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 IDM 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

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 IDM Northbound interface

[4 Operations](#)

[5 JMS notifications](#)

[6 Reference](#)

4 Operations

4.1 General

4.2 Inventory data retrieval

4.3 MTOSI object names

4.4 Supported operations

4.5 Service endpoints

4.6 Exceptions

4.7 Iterative operations in cluster deployments

4.1 General

The 5529 IDM NBI component retrieves inventory data from the 5529 IDM repository. Before using the 5529 IDM NBI component, ensure that the repository contains NE inventory data collected from your network. See the *5529 IDM Installation, Administration, and User Guide* for information about collecting inventory data.

The structure of the inventory data that the 5529 IDM NBI component reports depends on the NE objects and object attributes stored in the 5529 IDM repository. The 5529 IDM reports inventory data only for the NE objects and object attributes collected in the repository.

The 5529 IDM NBI component requires that the appropriate 5529 IDM NE support plug-ins be installed on the 5520 AMS server.

4.2 Inventory data retrieval

This section provides general information about the retrieval of inventory data in the 5529 IDM NBI architecture.

4.2.1 Web services

The 5529 IDM architecture supports the inventory data retrieval operations performed by OSS client applications. A web services server is the interface between the OSS and the 5529 IDM. The inventory data retrieval operations include synchronous RPC-type transactions, in which the data types and the request/response messages are strictly defined, and asynchronous transactions, such as the exportNetwork operation.

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

To send requests to the 5529 IDM using web services, an OSS client application must specify a user ID and password as part of the HTTP/S header. The 5529 IDM uses the 5520 AMS user management mechanism.

See the *5529 IDM Installation, Administration, and User Guide* for information about 5529 IDM users, synchronizing the inventory database, and the NE equipment in your network. See the *5520 AMS Administrator Guide* for information about managing users.

4.2.2 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.

4.2.3 Information model

To retrieve inventory data information, you need to create OSS client applications. An OSS client application incorporates an inventory data retrieval request wrapped in a SOAP envelope. The 5529 IDM receives the request, retrieves the inventory data from the repository, and sends a SOAP XML response message that contains inventory data to the OSS.

The SOAP envelope for the XML request/response includes 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 6 for more information about the SOAP XML request and response messages supported by the 5529 IDM NBI operations.

4.3 MTOSI object names

The 5529 IDM retrieves the 5520 AMS inventory data from the repository, maps the data to the appropriate MTOSI-compliant object names, and sends the inventory data to the OSS client applications.

The MTOSI object names to which the 5529 IDM maps inventory data are defined according to the network object containment hierarchy. Table 8 lists the inventory data mapping to the supported MTOSI objects, and describes the MTOSI object naming convention.



Note — Consult the schema documentation for objects supported by the 5529 IDM NBI, including logical objects, and for formats used for MTOSI object names. When you open the schema documentation, click on a URL under OSS Plugin Attributes Documentation (for example, click on iSAM_5.4) to display the attributes for MTOSI objects for an NE type and release. See section 3.3 for information about how to access the schema documentation.

Table 8 Object mapping

5520 AMS object	MTOSI object	MTOSI naming convention ⁽¹⁾ ⁽²⁾
Management domain (5520 AMS)	ManagedDomain	<mdNm>MD_name</mdNm>
Node	ManagedElement	<mdNm>MD_name</mdNm> <meNm>NE_name_or_IP_address</meNm>
Rack	EquipmentHolder	<mdNm>MD_name</mdNm> <meNm>NE_name_or_IP_address</meNm> <ehNm>/rack=rackNr</ehNm> ⁽³⁾
Subrack or shelf	EquipmentHolder	<mdNm>MD_name</mdNm> <meNm>NE_name_or_IP_address</meNm> <ehNm>/rack=rackNr/shelf=subrackNr</ehNm> ⁽³⁾

(1 of 2)

5520 AMS object	MTOSI object	MTOSI naming convention ⁽¹⁾ ⁽²⁾
Slot	EquipmentHolder	<code><mdNm>MD_name</mdNm></code> <code><meNm>NE_name_or_IP_address</meNm></code> <code><ehNm>/rack=rackNr/shelf=subrackNr/slot=slotNr</ehNm></code> ⁽³⁾
Card	Equipment	<code><mdNm>MD_name</mdNm></code> <code><meNm>NE_name_or_IP_address</meNm></code> <code><ehNm>/rack=rackNr/shelf=subrackNr/slot=slotNr</ehNm></code> ⁽³⁾ <code><eqNm>cardNr</eqNm></code>
Port	PhysicalTermination Point	<code><mdNm>MD_name</mdNm></code> <code><meNm>NE_name_or_IP_address</meNm></code> <code><ptpNm>/rack=rackNr/shelf=subrackNr/slot=slotNr/port=portNr</ptpNm></code> ⁽³⁾
ONT	EquipmentHolder	<code><mdNm>MD_name</mdNm></code> <code><meNm>NE_name_or_IP_address</meNm></code> <code><ehNm>/rack=rackNr/shelf=subrackNr/slot=slotNr/port=portNr/remote_unit=remote_unitNr</ehNm></code> ⁽³⁾
ONT slot	EquipmentHolder	<code><mdNm>MD_name</mdNm></code> <code><meNm>NE_name_or_IP_address</meNm></code> <code><ehNm>/rack=rackNr/shelf=subrackNr/slot=slotNr/port=portNr/remote_unit=remote_unitNr/slot=slotNr</ehNm></code> ⁽³⁾
ONT card	Equipment	<code><mdNm>MD_name</mdNm></code> <code><meNm>NE_name_or_IP_address</meNm></code> <code><ehNm>/rack=rackNr/shelf=subrackNr/slot=slotNr/port=portNr/remote_unit=remote_unitNr/slot=slotNr</ehNm></code> ⁽³⁾ <code><eqNm>cardNr</eqNm></code>
ONT port	PhysicalTermination Point	<code><mdNm>MD_name</mdNm></code> <code><meNm>NE_name_or_IP_address</meNm></code> <code><ptpNm>/rack=rackNr/shelf=subrackNr/slot=slotNr/port=portNr/remote_unit=remote_unitNr/slot=slotNr/port=portNr</ptpNm></code> ⁽³⁾
Voice user port	FloatingTermination Point	<code><mdNm>MD_name</mdNm></code> <code><meNm>NE_name_or_IP_address</meNm></code> <code><ftpNm>/rack=rackNr/shelf=subrackNr/slot=slotNr/voiceUserPort=portNr</ftpNm></code> ⁽³⁾

(2 of 2)

Notes

- ⁽¹⁾ The formats for MTOSI object names are described in the schema documentation delivered with the 5529 IDM. See section 3.3 for information about how to access the schema documentation.
- ⁽²⁾ *MD_name* is the managed domain name, *NE_name_or_IP_address* is the name or IP address of the NE, *rackNr* is the rack number, *subrackNr* is the shelf number, *portNr* is the port number, *remote_unitNr* is the remote unit number, *slotNr* is the slot number, and *cardNr* is the card number.
- ⁽³⁾ In these MTOSI object names, *rackNr*, *subrackNr*, *portNr*, *remote_unitNr*, *slotNr* (of remote unit), and *cardNr* represent a numeric value, and *slotNr* represents an alphanumeric string; for example LT4.

See section 6.2 for more information about the MTOSI objects and the naming conventions used to specify objects and object types in the SOAP body.

4.4 Supported operations

Table 9 describes the inventory data retrieval operations that are supported by the 5529 IDM NBI component. All operation requests and responses are structured as SOAP envelopes.

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.

Table 9 Inventory data retrieval operations

Operation	Description	Response
exportNetwork	Requests inventory data according to an export profile. The profile must be created before the operation can be performed.	A successful exportNetwork operation returns a request ID, posts an event to the JMS topic, and creates an output file in .csv format.
getAllManagedElementNames	Requests the names of all supported NEs supervised by the 5520 AMS, and collected by the 5529 IDM	A successful getAllManagedElementNames operation returns the names of all the supported NEs supervised by the 5520 AMS that are present in the 5529 IDM repository. The inventory data is returned in an MTOSI-compliant getAllManagedElementNamesResponse structure.
getEquipment	Requests inventory data for a rack, shelf (subrack), LT slot, LT card, ONT, ONT slot, or ONT card	A successful getEquipment operation returns a message containing inventory data for the specified object in an MTOSI-compliant getEquipmentResponse structure.
getExportedObject	Requests the latest data view for an object, which is exported using a specific export profile, in the 5529 IDM database or network	A successful getExportedObject operation returns an ordered list of attribute values for an object according to the given profile.
getInventory	Requests inventory data for an NE and its objects, or for a subset of NE objects according to your filtering criteria	A successful getInventory operation returns the first batch of the requested NE MTOSI-compliant objects in a getInventoryResponse structure.
getInventoryIterator	Requests inventory data for an NE in batches, according to a batch size specified in the header of a previous getInventory request	A successful getInventoryIterator operation returns the subsequent batch of NE MTOSI-compliant objects requested by a getInventory operation in a getInventoryResponse structure.
getManagedElement	Requests first level only inventory data for an NE	A successful getManagedElement operation returns a message containing inventory data for the specified NE in an MTOSI-compliant getManagedElementResponse structure. Only the managedElement information is returned.
getSystemHealthInfo	Requests the availability of the 5529 IDM NBI server	A successful getSystemHealthInfo operation returns a message containing an XML namespace; for example, xmlns="alu.v1".
getSystemInfo	Requests information about the 5529 IDM and the installed NE support plug-ins	A successful getSystemInfo operation returns a message containing information about the 5529 IDM application and the installed NE support plug-ins.
getTP	Requests inventory data for an LT port (PON port), Ethernet SHub port, or ONT port	A successful getEquipment operation returns a message containing inventory data for the specified object in an MTOSI-compliant getTPResponse structure.

(1 of 2)

Operation	Description	Response
query	Requests inventory data for all of the objects beneath an object or a set of objects of the same type	A successful query operation returns the first batch of the objects beneath the specified object (including the specified object).
queryIterator	Requests inventory data, in batches, according to a batch size specified in the header of a previous query request, for all of the objects beneath an object or a set of objects of the same type	A successful queryIterator operation returns the next batch of the objects that were requested by query object.

(2 of 2)



Note — The objects and attributes for which the following operations can be used are limited in scope:

- `getAllManagedElementNames`
- `getEquipment`
- `getInventory`
- `getInventoryIterator`
- `getManagedElement`
- `getTP`

The 5529 IDM NBI schema documentation provides a list of the limited scope of objects and attributes that can be used with these operations for each NE type and release. See section [3.3](#) for information about how to access the schema documentation. From the schema documentation main page, click on the NE URL links in the OSS Plugin Attributes Documentation section to view the lists.

The following operations can be used for all objects and attributes:

- `exportNetwork`
- `getExportedObject`

4.4.1 exportNetwork

When an OSS sends an `exportNetwork` request to the 5529 IDM NBI, the 5529 IDM extracts NE inventory data from the 5529 IDM database and writes the information to a comma-separated value (.csv) output file. The values in the output file are in UTF-8 format.

The objects and attributes included in the output file are defined by an export profile. See the *5529 IDM Installation, Administration, and User Guide* for information about creating export profiles.

The output file is saved on the server to the `$AMS_EXTERNAL_SHARED_DATA_HOME/idm/queries/IDMQueryResult` directory.

The exportNetwork operation returns a request ID right away. Following a successful network export operation, the 5529 IDM also posts an event to the JMS topic. The request ID, included in the .csv file name, makes it easier to match the output file with the request because it uniquely identifies the request within a 5520 AMS cluster. For information about the request ID returned in the operation response, see Table 37.

The output file name format is as follows:

username-networkexport-profilename-YYYY_MM_DD_HH_MM_SS-requestID.csv,
where:

- *username* is the user name you used to log in to the 5520 AMS
- *profilename* is the name of the export profile specified in the operation
- *YYYY_MM_DD_HH_MM_SS* is the date and time the file was created
- *requestID* is the ID number assigned to the exportNetwork task by the 5520 AMS

The content of the output file is organized in columns that read from left to right: NE name, object type, object name, and attribute values. The rows in the output file are ordered by NE name and object type.

The object type depends on the NE type and release, and on the objects selected for export. See the 5529 IDM NE support plug-in attributes guides for information about the object types for each NE type and release. The object name contains the full name of the object; for example, if the object is a line card, the object name contains the NE name, rack number, subrack number, and slot number.

The order of the attribute values in the output file is determined by the order of attribute values in the export profile. You determine the order of the attribute values when you create the export profile.

When the Enable Header Information In Bulk Network Export Operation check box is selected in the 5529 IDM Query Settings, header information for each object included in the export profile is added to the output file. The header information format is # NE Name, Object Type, Object Name, *Attribute Name*, where the *Attribute Name* fields are defined by the export profile. The header information is displayed in the row that precedes each row or set of rows grouped by NE name and object type.



Note — An attribute value that is displayed in the exportNetwork output file may not match the attribute value that is displayed in the Inventory Query Results window. The value in the Inventory Query Results window is in a user-friendly format. The value in the exportNetwork output file is in a non-user-friendly format.

For the attributes that display values in a non-user-friendly format in the exportNetwork output file, the attribute meta data that is defined for them is more complex than a simple value or list of values. The meta data can be a mix of different types or lengths of values. For this reason, the user-friendly value format is not supported for these attributes in the exportNetwork output file.

For example (ISAM NE):

- For the PON Loss Behavior attribute (NGPON2 Ethernet Port object), the user-friendly value is Don't Care (Does Not Send OMCI). The non-user-friendly value is noOmci.
- For the VSI ID attribute (BGP object), the user-friendly value is {Type=AS Number(2Bytes):Assigned Number, Value=0:0}. The non-user-friendly value is {type=type0, value=0:0}.

The exportNetwork operation supports password-based authentication and key-based authentication when SFTP is selected as the file transfer method. When key-based authentication is being used for SFTP, the OSS should not send a password tag in the exportNetwork request.

When the Prepend Site Name to Bulk Export File Name check box is selected in the 5529 IDM Query Settings, exported files have the site name prepended to them. See the *5529 IDM Installation, Administration, and User Guide* for information about configuring the 5529 IDM Query Settings.

See section 6.10 for an example of an exportNetwork output file. See the 5529 IDM NE support plug-in attributes guides to match the 5529 IDM NBI attribute names and values to the 5529 IDM GUI attribute names and values.

See section 6.9 for information about the elements that are present in the header and body of the exportNetwork request and response messages.

The exportNetwork operation is an Nokia proprietary operation.

4.4.1.1 Specifying a target list of NEs

The exportNetwork operation supports a target list of NEs. If you specify one or more NEs in the exportNetwork request operation, the NBI exports inventory data for only the NEs in the list. If you do not specify any NEs, inventory data is exported for all NEs managed by the 5520 AMS. Using an NE target list can help reduce the size of the output file and minimize the impact to performance. In the 5529 IDM NBI settings, you can specify the maximum number of NEs that can be included in the target list.

When at least one NE is included in the target list, you can send multiple exportNetwork operations in parallel. In the 5529 IDM NBI settings, you can specify the maximum number of operations that can be sent in parallel.

See the *5529 IDM Installation, Administration, and User Guide* for more information about the values of the 5529 IDM NBI settings and the procedure to configure the settings.

4.4.2 getAllManagedElementNames

An OSS client application uses the getAllManagedElementNames operation to retrieve the names of all supported NEs supervised by a 5520 AMS that are present in the 5529 IDM repository. The getAllManagedElementNames request must specify the name of the 5520 AMS as the ManagementDomain object name. You can configure the name of the 5520 AMS in the 5529 IDM NBI settings; see the *5529 IDM Installation, Administration, and User Guide* for more information.

The 5529 IDM supports the getAllManagedElementNames operation as defined in the tmf854 v1 namespace.

See section 6.4 for information about the elements that are present in the header and body of the getAllManagedElementNames request and response messages.

4.4.3 getEquipment

An OSS client application uses the getEquipment operation to request inventory data for a rack, shelf (subrack), LT slot, LT card, ONT, ONT card, NT card, NT slot, ACU_NTIO card, ACU_NTIO slot, applique card, or applique slot. Instead of using the getInventory command to search for the object, use the getEquipment shortcut to reduce lookup time. The request message needs to specify the object FDN. A successful getEquipment operation retrieves the object inventory data, and returns it within a getEquipmentResponse SOAP XML message.

The 5529 IDM supports the MTOSI-compliant getEquipment operation as defined in the tmf854 v1 namespace.

See section 6.13 for information about the elements that are present in the header and body of the getEquipment request and response messages.

4.4.4 getExportedObject

An OSS client application uses the `getExportedObject` operation to retrieve the latest view of an object, exported using a specific export profile, in the 5529 IDM database or the network. The response is an ordered list of attribute values for the object that is specified in the export profile.

For a successful `getExportedObject` operation, the user needs to provide the FDN of the target object (as specified by the object ID information in the 5529 IDM NE support plug-in attributes guides), object type, and profile name used for network export. The user also needs to provide a flag to indicate whether the object details are retrieved from the 5529 IDM database or the network. You can retrieve data for up to 100 objects in a single operation. See the *5529 IDM Installation, Administration, and User Guide* for information about creating export profiles.

See the 5529 IDM NE support plug-in attributes guides to match the 5529 IDM NBI attribute names and values to the 5529 IDM GUI attribute names and values.

See section 6.5 for information about the elements that are present in the header and body of the `getExportedObject` request and response messages.

4.4.5 getInventory

An OSS client application uses the `getInventory` operation to request NE inventory data from the 5529 IDM repository. A successful `getInventory` operation retrieves the NE inventory data, and returns a batch of it within a `getInventoryResponse` SOAP XML message. If there is more inventory data than the batch can contain, the OSS client application needs to use `getInventoryIterator` operations to retrieve the remaining data.



Caution — Use the `getInventory` operation only for individual queries. To perform a complete extraction of a large network, use the `exportNetwork` operation, as described in section 4.4.1.



Note — Nokia recommends that OSS client applications use the query operation instead of the `getInventory` operation for filtered requests of inventory data. The query operation is described in section 4.4.11.

You can view and configure the maximum number of `getInventory` operations executed in parallel using the Inventory Manual Collection setting in the Application Server Object Details view on the 5520 AMS GUI; see the *5529 IDM Installation, Administration, and User Guide* for information about the Inventory Manual Collection setting.

The `getInventory` request uses a filter element to specify the scope of the operation; see section 4.4.5.1.

The 5529 IDM supports the proprietary `getInventory` operation as defined in the `alu v1` namespace.

See section 6.6 for information about the elements that are present in the header and body of the getInventory request and response messages.

4.4.5.1 Scoping a getInventory request

You can use the 5529 IDM to specify the scope of a getInventory request, and to filter the inventory data on the basis of object attributes. The body of a getInventory request that uses scoping contains a scopeList container element.



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

Table 10 describes the getInventory scope elements implemented in the 5529 IDM NBI.



Note — XPATH queries are not supported.

Table 10 Scope attributes for getInventory

Attribute	Description	Values
scopeList	Container element for the scope attributes	—
scope	Container element for the baseObject and level attributes	—
baseObject	Container element for the attributes identifying the object that is the scope of the request: <ul style="list-style-type: none"> • mdNm • meNm • ehNm • eqNm • ptpNm • ftpNm • prNm • bgNm • vlanNm 	—
mdNm	Name of the managed domain; that is, the name of the 5520 AMS that manages the NE	Alphanumeric string For example: AMS12
meNm	Name or IP address of the managed element; that is, the name or IP address of the NE supervised by the 5520 AMS	Alphanumeric string For example: ISAM34 or IP address in the IPv4 or IPv6 format

(1 of 2)

Attribute	Description	Values
level	Specifies the query scope level	<p>The values are:</p> <ul style="list-style-type: none">• FIRST_LEVEL_ONLY (only the immediate children of the identified baseObject in the scope are considered; this value is supported with and without the filter, and with any value as a baseObject)• WHOLE_SUBTREE (only the identified baseObject in the scope and all the objects contained by it are considered; this value is not supported with a filter, and is supported only if the baseObject is referencing an ME)• BASE_OBJECT (only the identified baseObject in the scope is considered; this value is not supported by the 5529 IDM)

(2 of 2)

4.4.6 getInventoryIterator

An OSS client application uses the `getInventoryIterator` operation to request the next batch of inventory data requested by a previous `getInventory` operation. A successful `getInventoryIterator` operation returns the next batch of inventory data from the 5529 IDM repository. The OSS client application needs to send subsequent `getInventoryIterator` requests until a response indicates that it contains the final batch of inventory data.

The `getInventoryIterator` operation returns the same object type as the `getInventory` operation: a `getInventoryResponse` SOAP XML message.

The header of the `getInventoryResponse` message indicates the batch sequence number, and if that batch is the last in the sequence or not.

The 5529 IDM correlates the current `getInventoryIterator` operation request with the last `getInventory` operation request using the `iteratorReferenceURI` element in the MTOSI header. Each `getInventoryIterator` operation request contains the same `iteratorReferenceURI` value as the value in the initial `getInventory` operation response. See Table 28 for more information about the `iteratorReferenceURI` element.

The 5529 IDM supports the proprietary `getInventoryIterator` operation as defined in the `alu v1` namespace.

See section 6.6 for information about the elements that are present in the header and body of the `getInventory` request and response messages.

4.4.7 getManagedElement

An OSS client application uses the getManagedElement operation to request inventory data for a supervised NE. Instead of using the getInventory command to search for the NE, use the getManagedElement shortcut to reduce lookup time. The request message needs to specify the managed element name. A successful getManagedElement operation retrieves the NE inventory data, and returns it within a getManagedElementResponse SOAP XML message.

The 5529 IDM supports the MTOSI-compliant getManagedElement operation as defined in the tmf854 v1 namespace.

See section 6.13 for information about the elements that are present in the header and body of the getManagedElement request and response messages.

4.4.8 getSystemHealthInfo

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

See section 6.7 for information about the elements that are present in the header and body of the getSystemHealthInfo request and response messages.

4.4.9 getSystemInfo

An OSS client application uses the getSystemInfo operation to request information about the 5529 IDM application, and the installed 5529 IDM NE support plug-ins. The getSystemInfo operation request contains no input attributes. A successful getSystemInfo operation retrieves information about the 5529 IDM installation, and returns it within a getSystemInfoResponse SOAP XML message.

The 5529 IDM getSystemInfo operation is an Nokia proprietary operation.

See section 6.8 for information about the elements that are present in the header and body of the getSystemInfo request and response messages.

4.4.10 getTP

An OSS client application uses the getTP operation to request inventory data for an LT port (PON port), Ethernet SHub port, or ONT port. Instead of using the getInventory command to search for the object, use the getTP shortcut to reduce lookup time. The request message needs to specify the object FDN. A successful getTP operation retrieves the PTP or FTP inventory data, and returns it within a getTPResponse SOAP XML message.

The 5529 IDM supports the MTOSI-compliant getTP operation as defined in the tmf854 v1 namespace.

See section 6.13 for information about the elements that are present in the header and body of the getTP request and response messages.

4.4.11 query

An OSS client application uses the query operation to request inventory data for:

- an object or a set of objects of the same type
- an object that matches an object type
- a set of objects of the same type that match an object type
- all of the objects of the same type in an NE that match a list of object attribute values

An OSS client application uses this operation to request inventory data from the 5529 IDM repository. A successful query operation retrieves the inventory data, and returns a batch of it within a query SOAP XML message. If there is more inventory data than a batch size, the OSS client application needs to use queryIterator operations to retrieve the remaining data.



Note — If you add or remove an object attribute in the Collected Objects in Inventory view in the 5529 IDM GUI and run the NBI query operation, the query on the modified object will not return any data. You need to perform an inventory data collection on your network to make the modified object configuration available for query. See the *5529 IDM Installation, Administration, and User Guide* for information about selecting objects and attributes for inventory data collection.

See section 6.11 for information about the elements that are present in the header and body of the query request and response messages.

4.4.11.1 Filtering a query request

You can use the 5529 IDM to filter the inventory data for a query request. The body of a query request that uses filtering can contain the following filters:

- logical operators: AND and OR
- relational operators: EQUALS, GREATER, GREATER_EQUAL, LESS, LESS_EQUAL, and NOT_EQUAL

Filtering is only applicable when all of the objects specified in the baseObjectList attribute are of the same type; see Table 40 for information about the baseObjectList attribute.

Table 11 lists the filtering level and source combinations that are supported.

Table 11 Filtering level and source combinations

Level	Source	
	AMS	NETWORK
WHOLE_SUBTREE	Supported	Not supported
BASE_OBJECT	Supported	Supported

An EXCPT_INVALID_INPUT exception is raised when the unsupported combination is sent by the OSS; see Table 14 for more information.

Table 12 lists the attribute combinations in a query operation and indicates whether each combination is supported.

Table 12 Attribute combination support

Base object list	Level	Source	Query operation supported	Object type filtering supported	Attribute value filtering supported
Base object	WHOLE_SUBTREE	AMS	Yes	Yes	Yes
	WHOLE_SUBTREE	NETWORK	No	No	No
	BASE_OBJECT ⁽¹⁾	AMS	Yes	No	Yes
	BASE_OBJECT ⁽¹⁾	NETWORK	Yes	No	Yes
Whole network	WHOLE_SUBTREE	AMS	Yes	Yes	Yes
	WHOLE_SUBTREE	NETWORK	No	No	No

Note

⁽¹⁾ Filtering is only supported when all of the objects in the basicObjectList are of the same object type.

4.4.12 queryIterator

An OSS client application uses the queryIterator operation to request the next batch of inventory data requested by a previous query operation. A successful queryIterator operation returns the next batch of inventory data from the 5529 IDM repository. The OSS client application needs to send subsequent queryIterator requests until a response indicates that it contains the final batch of inventory data.

The queryIterator operation returns the same object type as the query operation: a queryResponse SOAP XML message.

The header of the queryResponse message indicates the batch sequence number, and whether that batch is the last in the sequence.

4.5 Service endpoints

The OSS client applications need to send the inventory data requests to specific 5529 IDM service endpoints, depending on the operation.

Table 13 lists the 5529 IDM service endpoints.



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 13 5529 IDM service endpoints

Operation	Service endpoint links	
	Standalone deployment ⁽¹⁾	Cluster deployment ⁽²⁾
exportNetwork getExportedObject	https://host:8443/idm/services/NetworkExportMgr http://host:8080/idm/services/NetworkExportMgr	https://load-balancer_IP:8443/idm/services/NetworkExportMgr http://load-balancer_IP:8080/idm/services/NetworkExportMgr
getAllManagedElementNames getManagedElement getTP	https://host:8443/idm/services/ManagedElementMgr http://host:8080/idm/services/ManagedElementMgr	https://load-balancer_IP:8443/idm/services/ManagedElementMgr http://load-balancer_IP:8080/idm/services/ManagedElementMgr
getEquipment	https://host:8443/idm/services/EquipmentInventoryMgr http://host:8080/idm/services/EquipmentInventoryMgr	https://load-balancer_IP:8443/idm/services/EquipmentInventoryMgr http://load-balancer_IP:8080/idm/services/EquipmentInventoryMgr
getInventory getInventoryIterator getSystemHealthInfo query queryIterator	https://host:8443/idm/services/InventoryRetrievalMgrExtns http://host:8080/idm/services/InventoryRetrievalMgrExtns	https://load-balancer_IP:8443/idm/services/InventoryRetrievalMgrExtns http://load-balancer_IP:8080/idm/services/InventoryRetrievalMgrExtns
getSystemInfo	https://host:8443/idm/services/InventoryRetrievalMgrExtns http://host:8080/idm/services/InventoryRetrievalMgrExtns	

Notes

⁽¹⁾ *host* is the IP address or host name of the application server

⁽²⁾ *load-balancer_IP* is the IP address of the load balancer

4.5.1 Verifying the 5529 IDM service endpoints

Before sending requests, it is good practice to verify the 5529 IDM 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 know the user ID and password of a user with the Inventory NBI - Administrator function. See the *5529 IDM Installation, Administration, and User Guide* for information about the 5529 IDM user functions and roles. See the *5520 AMS Administrator Guide* for information about managing user functions and roles.

4.6 Exceptions

Table 14 describes the exceptions that the 5529 IDM NBI component raises for the inventory data retrieval operations.

Table 14 Exceptions

Exception	Description
EXCPT_ACCESS_DENIED	Raised if the 5529 IDM license is invalid.
EXCPT_CAPACITY_EXCEEDED	Raised if the 5529 IDM receives a request for a new operation while the maximum number of concurrent operations are running.
EXCPT_COMM_FAILURE	Raised if the connection to the 5529 IDM is lost before the names of all managed element objects are returned to the OSS client application.
EXCPT_ENTITY_NOT_FOUND	Raised if a request specifies an NE type that is not supported, an NE that is not present anymore in the 5529 IDM repository, or an incorrect NE name.
EXCPT_INTERNAL_ERROR	Raised if the 5529 IDM encounters a non-specific OS internal failure, or an unexpected runtime error
EXCPT_INVALID_FILTER_DEFINITION	Raised if the 5529 IDM encounters an invalid filter in the query or queryIterator operation
EXCPT_INVALID_INPUT	Raised if an attribute value in the SOAP XML request is invalid (wrong format or out-of-range value). The reason provided in the exception body highlights the attribute that caused the error, and suggests values for the attribute.
EXCPT_NOT_IMPLEMENTED	Raised if the 5529 IDM does not support an attribute or attribute value specified in a request. For example: the getAllManagedElementNames operation does not support a multiple batch response.
EXCPT_UNABLE_TO_COMPLY	The request is valid, but the 5529 IDM was unable to comply for a different reason than those specified here.

4.7 Iterative operations in cluster deployments

In cluster deployments where a load balancer is used, 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 `getInventoryIterator` operation are processed by the same 5529 IDM 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 describes how the cookie persistence method works in a cluster deployment:

- The load balancer receives a `getInventory` operation request and directs it to a cluster node.
- The cluster node receives the `getInventoryIterator` 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 that is 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 `getInventoryIterator` operation must include the cookie, so that the load balancer will forward the request to the same cluster node.

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

5 JMS notifications

5.1 NE inventory updates

5.2 Event notifications

5.3 Information model

5.4 JMS client application guidelines

5.5 Subscription filter

5.6 Persistence and durable subscriptions

5.1 NE inventory updates

The 5529 IDM receives notifications of the following inventory events from the 5520 AMS:

- card insertion
- card removal
- ONT creation
- ONT deletion
- network export using the NBI

Notification for card insertion and removal is supported for the following NE types:

- 7302 ISAM/7330 ISAM FTTN
- 7342 ISAM FTTU
- 7356 ISAM FTTB
- 7360 ISAM FX
- 7363 ISAM MX
- GENBAND G6

Notification for ONT creation and deletion is supported for the following NE types:

- 7302 ISAM/7330 ISAM FTTN
- 7342 ISAM FTTU
- 7360 ISAM FX

The 5529 IDM translates the event information into an MTOSI 1.1 structure, and sends it to subscribed JMS client applications by way of a JMS provider.

The 5529 IDM also sends heartbeat notifications to inform the client applications that the event notification service is up and running.

5.2 Event notifications

Table 15 describes the event notifications that the 5529 IDM forwards to JMS client applications.



Note — The 5529 IDM 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.

Table 15 Event description

Event	Type	Description	See
Card insertion	ObjectCreation	Event that the 5529 IDM sends to an OSS client to indicate that a card has been inserted on an NE.	Tables 16 and 17
Card removal	ObjectDeletion	Event that the 5529 IDM sends to an OSS client to indicate that a card has been removed from an NE.	Tables 16 and 17
ONT creation	ObjectCreation	Event that the 5529 IDM sends to an OSS client to indicate that an ONT has been created on an NE.	Tables 16 and 18
ONT deletion	ObjectDeletion	Event that the 5529 IDM sends to an OSS client to indicate that an ONT has been deleted from an NE.	Tables 16 and 18
Successful network export	NetworkExportComplete	Event that the 5529 IDM sends to an OSS client to indicate that a network export has been successfully completed.	Tables 16 and 19
Failed network export	NetworkExportFailed	Event that the 5529 IDM sends to an OSS client to indicate that a network export has been performed and an error has been found.	Tables 16 and 19
Heartbeat	Heartbeat	Event that the 5529 IDM sends to an OSS client to indicate that the notification service is running.	Tables 16 and 20

Table 16 Event notification SOAP envelope header

Element	Description	Type or value
activityName	Activity (operation or notification) name	notify
msgName	Message name in the WSDL file	notify
msgType	The message type	NOTIFICATION
senderURI	Application sending the message	Alphanumeric string
destinationURI	Destination for the message	Alphanumeric string
communicationPattern	Message communication pattern	Notification
communicationStyle	Message communication style	MSG
timestamp	Date and time when the message was created	Date and time in MTOSI UTC format: <code>yyyyMMddHHmmss.S[Z {+ -}HHMm]</code> See section 3.4.1 for more information about the MTOSI UTC format.

Table 17 Card insertion and removal event SOAP envelope body

Element	Description	Type or value	Applies to card	
			Insertion	Removal
topic	Event notification topic	topic/Inventory	Yes	Yes
message	Container element for the event notification	—	Yes	Yes
ObjectCreation	Container element for the object creation event notification	—	Yes	No
ObjectDeletion	Container element for the object creation event notification	—	No	Yes
eventInfo	Container element for the event information	—	Yes	Yes
notificationId	Notification ID	Alphanumeric string	Yes	Yes
objectName	Container element for the object name attributes	—	Yes	Yes
mdNm	Name of the managed domain—the name of the 5520 AMS that manages the NE	Alphanumeric string For example: AMS:12	Yes	Yes
meNm	Name or IP address of the managed element—the name or IP address of the NE supervised by the 5520 AMS	Alphanumeric string For example: ISAM34 or IP address in the IPv4 or IPv6 format	Yes	Yes
ehNm	Equipment holder name	For example: /rack=1/shelf=1/slot=1	Yes	Yes
eqNm	Equipment name	Alphanumeric string For example: 1	Yes	Yes
vendorExtensions	Container object for vendor extensions attributes	—	Yes	No
package	Container object for a set of NameAndStringValue elements	—	Yes	No
NameAndStringValue	Container object for the attribute name and value	—	Yes	No
name	Attribute name	Alphanumeric string	Yes	No
value	Attribute value	Alphanumeric string	Yes	No
objectType	Object type	OT_EQUIPMENT	No	Yes
osTime	Date and time of the notification creation	Date and time in MTOSI UTC format: yyyyMMddHHmmss.S[Z {+ -}HHMM] See section 3.4.1 for more information about the MTOSI UTC format.	No	Yes

Table 18 ONT creation and deletion event SOAP envelope body

Element	Description	Type or value	Applies to ONT	
			Creation	Deletion
topic	Event notification topic	topic/Inventory	Yes	Yes

(1 of 2)

Element	Description	Type or value	Applies to ONT	
			Creation	Deletion
message	Container element for the event notification	—	Yes	Yes
ObjectCreation	Container element for the object creation event notification	—	Yes	No
ObjectDeletion	Container element for the object creation event notification	—	No	Yes
eventInfo	Container element for the event information	—	Yes	Yes
notificationId	Notification ID	Alphanumeric string	Yes	Yes
objectName	Container element for the object name attributes	—	Yes	Yes
mdNm	Name of the managed domain—the name of the 5520 AMS that manages the NE	Alphanumeric string For example: AMS:12	Yes	Yes
meNm	Name or IP address of the managed element—the name or IP address of the NE supervised by the 5520 AMS	Alphanumeric string For example: ISAM34 or IP address in the IPv4 or IPv6 format	Yes	Yes
ehNm	Equipment holder name	For example: /rack=1/shelf=1/slot=LT3/port=1/rem ote_unit=9	Yes	Yes
objectType	Object type	OT_EQUIPMENT HOLDER	Yes	Yes
osTime	Date and time of the notification creation	Date and time in MTOSI UTC format: yyyyMMddHHmmss.S[Z {+ -}HHMM] See section 3.4.1 for more information about the MTOSI UTC format.	Yes	Yes
vendorExtensions	Container object for vendor extensions attributes	—	Yes	No
package	Container object for a set of NameAndStringValue elements	—	Yes	No
NameAndStringValue	Container object for the attribute name and value	—	Yes	No
name	Attribute name	Alphanumeric string	Yes	No
value	Attribute value	Alphanumeric string	Yes	No

(2 of 2)

Table 19 Successful and failed network export SOAP envelope body

Element	Description	Type or value	Applies to	
			Successful	Failed
topic	Event notification topic	topic/Inventory	Yes	Yes
message	Container element for the event notification	—	Yes	Yes
VendorNotification	Container element for the event notification	—	Yes	Yes

(1 of 2)

Element	Description	Type or value	Applies to	
			Successful	Failed
notificationId	Notification ID	Alphanumeric string	Yes	Yes
vendorNotificationType	Container element for the event information	NetworkExportComplete	Yes	No
		NetworkExportFailed	No	Yes
vendorExtensions	Container object for vendor extensions attributes	—	Yes	Yes
package	Container object for a set of NameAndStringValue elements	—	Yes	Yes
NameAndStringValue	Container object for the attribute name and value	—	Yes	Yes
name	Attribute name	Alphanumeric string	Yes	Yes
value	Attribute value	Alphanumeric string	Yes	Yes

(2 of 2)

Table 20 Heartbeat SOAP envelope body

Element	Description	Type or value
topic	Event notification topic	topic/Inventory
message	Container element for the event notification	—
Heartbeat	Container element for the event notification	—
notificationId	Notification ID	Alphanumeric string
objectType	Object type	OT_OS
objectName	Container element for the object name attributes	—
osNm	Object name	<osNm>Alcatel-Lucent/IDM</osNm>
osTime	Date and time of the notification creation	Date and time in MTOSI UTC format: yyyyMMddHHmmss.S[Z {+ -}HHMm] See section 3.4.1 for more information about the MTOSI UTC format.

5.3 Information model

The 5529 IDM architecture supports event notifications by way of a JMS provider established between the OSS and the 5529 IDM. 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 IDM 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 7.

See chapter 6 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.

5.4 JMS client application guidelines

To receive event notifications from the 5529 IDM through the JMS provider, you need to create a JMS client application to subscribe to the Inventory topic. Because the 5529 IDM 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 IDM 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

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.

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 describes the parameter values that a JMS client application uses to connect to the 5529 IDM in order to subscribe to the Inventory topic.

The following parameters apply to standalone deployments:

- `java.naming.factory.initial`=“org.jboss.naming.remote.client.InitialContextFactory”
- `java.naming.provider.url`=“remote://*host*:4447”

where *host* is the 5529 IDM host IP address

The following parameters apply to cluster deployments:

- `java.naming.factory.initial`=“org.jboss.naming.remote.client.InitialContextFactory”
- `java.naming.provider.url`=use a comma-separated list of entries that follow the “remote://*IP address*:4447” format; for example, “remote://*IP address*1:4447, remote://*IP address*2:4447, remote://*IP address*3:4447”

where *IP address**xx* is the IP address of a cluster NE

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 IDM 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:

- | | |
|--|---|
| • <code>axs-encryption-app-<i>release-version</i>.jar</code> | • <code>picketbox-4.9.6.Final.jar</code> |
| • <code>axs-mobject-api-<i>release-version</i>.jar</code> | • <code>picketbox-infinispan-4.9.6.Final.jar</code> |
| • <code>axs-mobject-remote-api-<i>release-version</i>.jar</code> | • <code>slf4j-simple-1.7.21.jar</code> |
| • <code>jboss-logging-3.3.0.Final.jar</code> | • <code>wildfly-client-all.jar</code> |
| • <code>log4j-1.2.14.jar</code> | • <code>xbean-2.6.0.jar</code> |

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

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



Caution — Ensure that you install the latest 5529 IDM OSS client and JBoss libraries that are provided with this 5529 IDM 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 IDM.

5.5.1 JMS property fields

The 5529 IDM 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_vendorNotificationType
- endTime
- reason
- requestId

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 IDM event notification context, the MTOSI_EventType property field can be filtered on the following values:

- AttributeValueChange
- Heartbeat
- ObjectCreation

- ObjectDeletion
- VendorNotification (includes NetworkExport events)

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 21 describes several examples of filters.

Table 21 Subscription filter examples

Filter	Description
MTOSI_EventType='Heartbeat'	Receives only heartbeat notifications
filter=MTOSI_EventType='VendorNotification' AND MTOSI_vendorNotificationType='NetworkExportFail'	Receives only messages that satisfy all of the conditions: <ul style="list-style-type: none">• event type is VendorNotification• vendorNotificationType is NetworkExportFail
filter=MTOSI_EventType='Heartbeat' OR MTOSI_EventType='VendorNotification'	Receives messages that satisfy one of the following two conditions: <ul style="list-style-type: none">• event type is Heartbeat• event type is VendorNotification
filter=NOT MTOSI_EventType='Heartbeat'	Receives all messages except messages that have the Heartbeat event type

5.6 Persistence and durable subscriptions

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

5.6.1 Persistence overview

The 5529 IDM 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 set the JMS message delivery mode in the 5520 AMS GUI; see the *5520 AMS Administrator Guide* for more information. See the *5529 IDM Installation, Administration, and User Guide* for more information about configuring the 5529 IDM.

5.6.2 Durable subscriptions

With a durable subscription, the subscriber can disconnect (unintentionally), but 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.

In order 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 IDM. A durable subscription can have only one active subscriber at a time.

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 of 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 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.

5.6.3 Performance impact

The reliability provided by the 5529 IDM, using the persistence message delivery mode and providing support for durable subscriptions, can increase the storage overhead and decrease performance.

6 Reference

6.1 Purpose

6.2 Northbound objects

6.3 SOAP envelope components

6.4 getAllManagedElementNames SOAP envelope

6.5 getExportedObject SOAP envelope

6.6 getInventory SOAP envelope

6.7 getSystemHealthInfo SOAP envelope

6.8 getSystemInfo SOAP envelope

6.9 exportNetwork SOAP envelope

6.10 exportNetwork output file example

6.11 query SOAP envelope

6.12 JMS notification SOAP envelope

6.13 SOAP envelopes for shortcut operations

6.1 Purpose

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

6.2 Northbound 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 XML 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.

Table 8 lists the mapping of 5520 AMS objects to MTOSI-compliant object names.



Note — When a data string is displayed in the GUI or exported to a CSV file, the number of characters in the string is limited to 21844. If the data string is longer than 21844 characters, only the first 21844 characters of the string are displayed and the remainder of the string is truncated.

6.2.1 Inventory data retrieval

The inventory data for each object is returned in an `inventoryObject` element contained in a response message. There are two types of objects:

- physical objects (for example, nodes, racks, shelves, slots, and ports)
- logical objects (for example, profiles, VLANs, and bonding groups)

The object inventory data in a response message is ordered as follows:

- physical objects
- logical objects
 - profiles
 - VLANs
 - bonding groups

The sequence of the profiles depends on the type of NE that is being inventoried.

6.2.2 Specifying the value for the `meNm` attribute in the object FDN

You can specify whether the value for the `meNm` attribute in the object FDN is the NE name or the IP address of the NE. By default, the 5529 IDM NBI expects that the `meNm` attribute value is the NE name. You can modify the 5529 IDM NBI settings to specify that the `meNm` attribute value is the IP address of the NE, using the `Support NE IP instead of NE name` parameter.

If you specify IP address in the 5529 IDM NBI settings, the 5529 IDM NBI expects that the `meNm` attribute value is the IP address of the NE. When an operation is run, the 5529 IDM NBI retrieves the NE name from the 5520 AMS using the IP address, and then continues processing the operation. In the operation response, the IP address of the NE is displayed instead of the NE name in the object FDN. The IPv4 and IPv6 address formats are supported.

The functionality applies to the following 5529 IDM NBI operations:

- exportNetwork
- getAllManagedElementNames
- getEquipment
- getExportedObject
- getInventory/getInventoryIterator
- getManagedElement
- getTP
- query/queryIterator

See the *5529 IDM Installation, Administration, and User Guide* for more information about the 5529 IDM NBI settings.

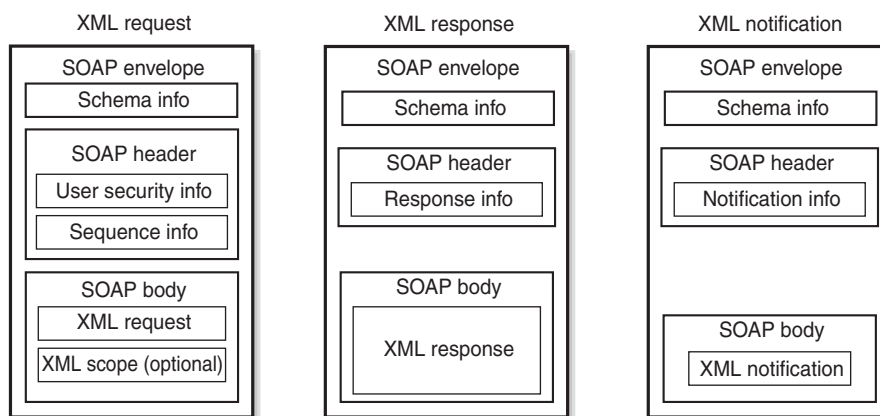
6.3 SOAP envelope components



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 types of SOAP envelopes and their content.

Figure 1 SOAP envelope types



19782

6.4 getAllManagedElementNames SOAP envelope

This section provides reference information about the standard SOAP envelope for `getAllManagedElementNames` and `getAllManagedElementNamesResponse` messages.

6.4.1 Header

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

Table 22 `getAllManagedElementNames` SOAP envelope header

Element	Description	Request	Response	Values
activityName ⁽¹⁾	Activity (operation or notification) name	Yes	Yes	<code>getAllManagedElementNames</code>
msgName ⁽¹⁾	Message name in the WSDL file	Yes	Yes	<code>getAllManagedElementNames</code> <code>getAllObjectNamesResponse</code>
messageType	Message type	Yes	Yes	REQUEST RESPONSE ERROR
senderURI ⁽¹⁾	Application sending the message	Yes	Yes	Character string
destinationURI ⁽¹⁾	Destination for the message	Yes	Yes	Character string
correlationId	Arbitrary number assigned to the request to correlate it with the response and differentiate it from other requests sent by the OSS client	Yes	Yes	Integer
activityStatus	Status of the response operation	No	Yes	SUCCESS FAILURE WARNING
communicationPattern	Message communication pattern	Yes	Yes	<code>SimpleResponse</code>
communicationStyle	Message communication style	Yes	Yes	MSG RPC
timestamp	Date and time when the message was created	Yes	Yes	Date and time in MTOSI UTC format: <code>yyyyMMddHHmmss.S[Z {+ -}HHMm]</code> See section 3.4.1 for more information about the MTOSI UTC format.

Note

- ⁽¹⁾ A value for this element is required in the NBI operation request, but the value is not validated by the 5529 IDM. Any value can be added for this element and the operation can still be successful.

6.4.2 Body

Table 23 describes the elements used in the SOAP envelope body of a `getAllManagedElementNames` request message.

Table 23 **getAllManagedElementNames SOAP XML request elements**

Element	Description	Value
getAllManagedElementNames	Container element for the SOAP envelope body information in a request message	—
mdName	Container element for the managed domain name in a request	—
mdNm	Managed domain name	Alphanumeric string

Table 24 describes the elements used in the SOAP envelope body of a getAllManagedElementNames response message.

Table 24 **getAllManagedElementNames SOAP XML response elements**

Element	Description	Value
getAllObjectNamesResponse	Container element for the managed domain and NE names in a response	—
names	Container element for the managed domain names and NE names listed in a response; it contains multiple name elements.	—
name	Container element for a managed domain name and an NE name	—
mdNm	Managed domain name	Alphanumeric string
meNm	NE name or IP address (IPv4 or IPv6 address format)	Alphanumeric string

6.5 getExportedObject SOAP envelope

This section provides reference information about the standard SOAP envelope for getExportedObject messages.

6.5.1 Header

Table 25 describes the elements used in the getExportedObject SOAP envelope header, and indicates the element relevance to each type of SOAP XML message (request or response).

Table 25 **getExportedObject SOAP envelope header**

Element	Description	Request	Response	Values
activityName ⁽¹⁾	Activity (operational) name	Yes	Yes	getExportedObject
msgName ⁽¹⁾	Message name in the WSDL file	Yes	Yes	getExportedObject

(1 of 2)

Element	Description	Request	Response	Values
msgType	Message type	Yes	Yes	REQUEST RESPONSE ERROR
senderURI ⁽¹⁾	Application sending the message	Yes	Yes	Alphanumeric string
destinationURI ⁽¹⁾	Destination for the message	Yes	Yes	Alphanumeric string
activityStatus ⁽²⁾	Status of the response operation	No	Yes	SUCCESS FAILURE WARNING
communicationPattern	Sets the communication pattern	Yes	Yes	SimpleResponse
communicationStyle	Sets the 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 3.4.1 for more information about the MTOSI UTC format.

(2 of 2)

Notes

- ⁽¹⁾ A value for this element is required in the NBI operation request, but the value is not validated by the 5529 IDM. Any value can be added for this element and the operation can still be successful.
- ⁽²⁾ A FAILURE message is returned for the operation only when the data retrieval for all of the target objects in the request fails. If the data retrieval for some of the target objects succeeds while failing for others, a SUCCESS message is still returned for the operation and the objects for which the data retrieval failed are identified by an error message in the objectRetrievalStatus element of the operation response.

6.5.2 Body

Table 26 describes the elements used in the SOAP envelope body of a getExportedObject request message.

Table 26 getExportedObject SOAP XML request elements

Element	Description	Value ⁽¹⁾
targetExportedObjectList	Container element for the list of objects for which you need to retrieve data	—
targetExportedObject	Container element for information about each object for which you need to retrieve data. Data for up to 100 objects can be retrieved.	—
targetName	Container element for the object FDN	—
mdNm	Name of the managed domain; that is, the name of the 5520 AMS that manages the NE	Alphanumeric string
meNm	Name or IP address of the managed element; that is, the name or IP address of the NE supervised by the 5520 AMS. The IP address supports the IPv4 or IPv6 format.	Alphanumeric string

(1 of 2)

Element	Description	Value ⁽¹⁾
propNm ⁽²⁾	<p>The object type and object name.</p> <p>The format of the Ethernet port (IACM) object type and object name is /type=Ethernet Port/R[<i>rackNr</i>].S[<i>subrackNr</i>].LT[<i>slotNr</i>].PON[<i>portNr</i>].ONT[<i>ontNr</i>].C[<i>ontSlotNr</i>].P[<i>ontPortNr</i>]; for example, /type=Ethernet Port/R1.S1.LT1.PON1.ONT1.C1.P1.</p> <p>The format of the xDSL port object type and object name is /type=XDSL Port/R[<i>rackNr</i>].S[<i>subrackNr</i>].LT[<i>slotNr</i>].P[<i>portNr</i>]; for example, /type=XDSL Port/R1.S1.LT1.P1.</p> <p>For the list of supported values for this element, see the 5529 IDM NE support plug-in attributes guides.</p>	Alphanumeric string
source	Indicates whether the object data should be retrieved from the 5529 IDM database or the network	DB NETWORK
profile	Name of the export profile	Alphanumeric string

(2 of 2)

Notes

- (1) When an attribute contains a comma, the comma is enclosed with quotation marks. The comma separation format is preserved so the user sees the exact format as is used in export file.
- (2) *rackNr* is the rack number, *subrackNr* is the shelf number, *slotNr* is the slot number, *portNr* is the port number, *ontNr* is the ONT number, and *ontSlotNr* is the ONT slot number.

Table 27 describes the elements used in the SOAP envelope body of a `getExportedObject` response message.

Table 27 getExportedObject SOAP XML response elements

Element	Description	Value ⁽¹⁾
getExportedObjectResponse	Container element for the response information	—
fetchedObjectList	Container element for the list of the objects that are returned	—
fetchedObject	Container element for information about each object that is returned. Data for up to 100 objects can be returned.	—
name	Container element for the object FDN: managed domain name and NE name, and object type and object name	—
mdNm	Name of the managed domain; that is, the name of the 5520 AMS that manages the NE	Alphanumeric string
meNm	Name or IP address of the managed element; that is, the name or IP address of the NE supervised by the 5520 AMS. The IP address supports the IPv4 or IPv6 format.	Alphanumeric string
propNm ⁽²⁾	<p>The object type and object name.</p> <p>The format of the Ethernet port (IACM) object type and object name is /type=Ethernet Port/R[<i>rackNr</i>].S[<i>subrackNr</i>].LT[<i>slotNr</i>].PON[<i>portNr</i>].ONT[<i>ontNr</i>].C[<i>ontSlotNr</i>].P[<i>ontPortNr</i>]; for example, /type=Ethernet Port/R1.S1.LT1.PON1.ONT1.C1.P1.</p> <p>The format of the xDSL port object type and object name is /type=XDSL Port/R[<i>rackNr</i>].S[<i>subrackNr</i>].LT[<i>slotNr</i>].P[<i>portNr</i>]; for example, /type=XDSL Port/R1.S1.LT1.P1.</p> <p>For the list of supported values for this element, see the 5529 IDM NE support plug-in attributes guides.</p>	Alphanumeric string

(1 of 2)

Element	Description	Value ⁽¹⁾
targetAttributes	<p>Object attributes ordered in a comma-separated list, as defined in the export profile. The format of the targetAttributes value is a comma-separated list that reads from left to right and includes the following information: NE name, object type, object name, and attribute values (in the order in which the attribute values were defined in the export profile).</p> <p>Example of the targetAttributes element value for an Ethernet port on an ONT: <targetAttributes>ISAM100,Ethernet Port,ISAM100:R1.S1.LT2.PON8.ONT1.C1.P1,available</targetAttributes>. The following attribute was defined in the export profile: Customer ID (available is the default value for Customer ID).</p> <p>Example of the targetAttributes element value for an xDSL port: <targetAttributes>ISAM100,XDSL Port,ISAM100:R1.S1.LT4.P1,Unknown /Not Supported,available,Disabled,PTM</targetAttributes>. The following attributes were defined in the export profile: Actual VECT CPE Type, Customer ID, PM TCA Reporting, Transfer Mode (Actual). (available is the default value for Customer ID.)</p> <p>See the <i>5529 IDM Installation, Administration, and User Guide</i> for information about creating export profiles.</p>	Alphanumeric string
objectRetrievalStatus ⁽³⁾	<p>Status of the data retrieval from the object:</p> <ul style="list-style-type: none"> • Success indicates that the object data was retrieved successfully. • Invalid_Input indicates that the input data provided for the object is invalid. • Object_Not_Found indicates that the object does not exist in the system. • Profile_Not_Found indicates that the export profile does not exist in the 5529 IDM. • Unable_to_Comply indicates that the object exists but not in the profile. 	Success Invalid_Input Object_Not_Found Profile_Not_Found Unable_to_Comply

(2 of 2)

Notes

- ⁽¹⁾ When an attribute contains a comma, the comma is enclosed with quotation marks. The comma separation format is preserved so the user sees the exact format that is used in export file.
- ⁽²⁾ *rackNr* is the rack number, *subrackNr* is the shelf number, *slotNr* is the slot number, *portNr* is the port number, *ontNr* is the ONT number, *ontSlotNr* is the ONT slot number, and *ontPortNr* is the ONT port number.
- ⁽³⁾ If the object data is retrieved from the network (the source element has the NETWORK value in the operation request), and the object you are retrieving data for is on an NE that is in an unsupervised state, the operation returns an error message for that object. If the object data is retrieved from the database (the source element has the DB value in the operation request) and the object you are retrieving data for is also on an NE in an unsupervised state, the operation returns a SUCCESS message for the object.

6.6 getInventory SOAP envelope

This section provides reference information about the standard SOAP envelope for getInventory messages.

6.6.1 Header

Table 28 describes the most common elements supported by the 5529 IDM that are used in a standard SOAP envelope header of a getInventory message, and indicates the element relevance to each type of SOAP XML message (request or response).

See the TMF MTOSI documentation for information about all SOAP header elements defined in MTOSI.

Table 28 **getInventory SOAP envelope header**

Element	Description	Request	Response	Values
activityName ⁽¹⁾	Activity (operation) name	Yes	Yes	getInventory
msgName ⁽¹⁾	Message name in the WSDL file	Yes	Yes	getInventory getInventoryResponse
msgType	Message type	Yes	Yes	REQUEST RESPONSE NOTIFICATION ERROR
senderURI ⁽¹⁾	Application sending the message	Yes	Yes	Alphanumeric string
destinationURI ⁽¹⁾	Destination for the message		Yes	Alphanumeric string
activityStatus	Status of the response operation	No	Yes	SUCCESS FAILURE WARNING
correlationId	Arbitrary number assigned to the request to correlate it with the response and differentiate it from other requests sent by the OSS client	Yes	Yes	Integer
communicationPattern	Sets the communication pattern	Yes	Yes	MultipleBatchResponse
communicationStyle	Sets the communication style	Yes	Yes	RPC
requestedBatchSize	The logical size of the batch for a multi-response communication pattern	Yes	Yes	0 to 1500 Default is 1500.
batchSequenceNumber	Identifies the batch sequence number in a multiple response communication pattern	No	Yes	0 to integer (last batch number)
batchSequenceEndOfReply	Indicates the end of a batch response when the value is true	No	Yes	true false
iteratorReferenceURI	Data retrieval iterator	No	Yes	A string of numeric characters unique for each operation
time stamp	Date and time when the message was created	Yes	Yes	Date and time in MTOSI UTC format: <code>yyyyMMddHHmmss.S[Z {+ -}HHMM]</code> See section 3.4.1 for more information about the MTOSI UTC format.

Note

- ⁽¹⁾ A value for this element is required in the NBI operation request, but the value is not validated by the 5529 IDM. Any value can be added for this element and the operation can still be successful.

6.6.2 Body

Table 29 describes the elements used in a standard SOAP envelope body of a `getInventory` request message.



Note — The only element present in a `getInventoryIterator` request is `getInventoryIterator`.

Table 29 `getInventory` SOAP XML request elements

Element	Description
<code>getInventory</code>	Container element in the request body
<code>filter</code>	Container element for the objects that specify the scope of the operation. For information about filtering, see section 4.4.5.

Table 30 lists the most important object elements that can be present in a SOAP envelope body of a `getInventoryResponse` message. Each object contains a set of attributes that provide inventory data about the object, including the `vendorExtensions` or `objectExtensions` attribute.

For information about the subset of NE object attributes that is supported by the NBI, see the HTML files that are provided with the 5529 IDM. See the *5529 Enhanced Applications Release Notice* for information about how to access the files.

Table 30 `getInventoryResponse` SOAP XML response elements

Element	Content
<code>getInventoryResponse</code>	Inventory data in a response
<code>inventoryObjectData</code>	Inventory data in a <code>getInventoryResponse</code> element
<code>inventoryObject</code>	All inventory data attributes about an object
<code>me</code>	Inventory data about a managed element (NE) object
<code>eh</code>	Inventory data about an equipment holder object
<code>eq</code>	Inventory data about an equipment object
<code>ptp</code>	Inventory data about a physical termination point object
<code>ctp</code>	Inventory data about a bridge port, vcl, VoIP service, or HSI service
<code>as</code>	Inventory data about an association
<code>sfp</code>	Inventory data about a small factor pluggable object
<code>ftp</code>	Inventory data about a voice user port
<code>pr</code>	Inventory data about a profile (logical object)
<code>bg</code>	Inventory data about a bonding group (logical object)
<code>vlan</code>	Inventory data about a vlan (logical object)

6.7 getSystemHealthInfo SOAP envelope

This section provides reference information about the standard SOAP envelope for getSystemHealthInfo messages.

6.7.1 Header

Table 31 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 31 getSystemHealthInfo SOAP envelope header

Element	Description	Request	Response	Values
activityName ⁽¹⁾	Activity (operation) name	Yes	Yes	getSystemHealthInfo
msgName ⁽¹⁾	Message name in the WSDL file	Yes	No	getSystemHealthInfo
		No	Yes	getSystemHealthInfoResponse
msgType	Message type	Yes	Yes	REQUEST RESPONSE ERROR
senderURI ⁽¹⁾	Application sending the message	Yes	Yes	Alphanumeric string
destinationURI ⁽¹⁾	Destination for 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	No	Yes	Date and time in MTOSI UTC format: <code>yyyyMMddHHmmss.S[Z {+ -}HHMm]</code> See section 3.4.1 for more information about the MTOSI UTC format.

Note

- ⁽¹⁾ A value for this element is required in the NBI operation request, but the value is not validated by the 5529 IDM. Any value can be added for this element and the operation can still be successful.

6.7.2 Body

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

Table 32 describes the elements used in the SOAP envelope body of a `getSystemHealthInfo` response message.

Table 32 `getSystemHealthInfo` SOAP XML response element

Element	Description	Value
<code>getSystemHealthInfoResponse</code>	The XML namespace; for example, <code>xmlns="alu.v1"</code>	Alphanumeric string

6.8 `getSystemInfo` SOAP envelope

This section provides reference information about the standard SOAP envelope for `getSystemInfo` messages.

6.8.1 Header

Table 33 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 33 `getSystemInfo` SOAP envelope header

Element	Description	Request	Response	Values
<code>activityName</code> ⁽¹⁾	Activity (operation) name	Yes	Yes	<code>getSystemInfo</code>
<code>msgName</code> ⁽¹⁾	Message name in the WSDL file	Yes	Yes	<code>getSystemInfo</code>
<code>msgType</code>	The message type	Yes	Yes	REQUEST RESPONSE ERROR
<code>senderURI</code> ⁽¹⁾	Application sending the message	Yes	Yes	Alphanumeric string
<code>destinationURI</code> ⁽¹⁾	Destination for the message	Yes	Yes	Alphanumeric string
<code>activityStatus</code>	Status of the response operation	No	Yes	SUCCESS FAILURE WARNING
<code>communicationPattern</code>	Message communication pattern	Yes	Yes	<code>SimpleResponse</code>
<code>communicationStyle</code>	Message communication style	Yes	Yes	<code>RPC</code>

(1 of 2)

Element	Description	Request	Response	Values
timestamp	Date and time when the message was created	Yes	Yes	Date and time format: <i>dd/MM/yyyy HH:mm:ss</i> (not MTOSI UTC format) where <i>dd</i> is the day <i>MM</i> is the month <i>yyyy</i> is the year <i>HH</i> is the hour <i>mm</i> is the minute <i>ss</i> is the second

(2 of 2)

Note

- ⁽¹⁾ A value for this element is required in the NBI operation request, but the value is not validated by the 5529 IDM. Any value can be added for this element and the operation can still be successful.

6.8.2 Body

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

Table 34 describes the elements used in the SOAP envelope body of a `getSystemInfo` response message.



Note — Table 34 lists only the elements that are supported by the 5529 IDM 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 IDM.

Table 34 `getSystemInfo` SOAP XML response elements

Element	Description	Value
<code>getSystemInfoResponse</code>	Container element for the system information	—
<code>systemTime</code>	The current system date and time on the local host (the host that serves the request)	Date and time in MTOSI UTC format: <i>yyyyMMddHHmmss.S[Z {+ -}HHMm]</i> See section 3.4.1 for more information about the MTOSI UTC format.
<code>version</code>	Container element for application version information	—
<code>timestamp</code>	The date and time that the 5529 IDM software load was produced	Date and time in the format: <i>dd-mmm-yyyy hh:mm:ss</i> ⁽¹⁾
<code>version</code>	5529 IDM application version	Alphanumeric string For example: 9.6.07
<code>installedPluginList</code>	Container element for the list of installed 5529 IDM NE support plug-ins	—
<code>installedPlugin</code>	Container element for information about each installed 5529 IDM NE support plug-in	—

(1 of 2)

Element	Description	Value
id	The NE support plug-in ID	Alphanumeric string For example: idm-isam.5.2
description	The NE support plug-in description	Alphanumeric string For example: Nokia iSAM.5.2
domain	The description of the application using the NE support plug-in	INVENTORY
vendor	The NE support plug-in vendor	Nokia
versionInfo	The NE support plug-in version	Alphanumeric string
buildTimestamp	The date and time that the NE support log-in software load was produced	Date and time in the format: <i>dd-mmm-yyyy hh:mm:ss</i> ⁽¹⁾

(2 of 2)

Note

⁽¹⁾ *dd* is day, *mmm* is month, *yyyy* is year, *hh* is hour, *mm* is minutes, *ss* is seconds

6.9 exportNetwork SOAP envelope

This section provides reference information about the standard SOAP envelope for exportNetwork messages.

6.9.1 Header

Table 35 describes the elements used in the exportNetwork SOAP envelope header, and indicates the element relevance to each type of SOAP XML message (request or response).

Table 35 exportNetwork SOAP envelope header

Element	Description	Request	Response	Values
activityName ⁽¹⁾	Activity (operation) name	Yes	Yes	exportNetwork
msgName ⁽¹⁾	Message name in the WSDL file	Yes	Yes	exportNetwork
msgType	The message type	Yes		REQUEST RESPONSE ERROR
senderURI ⁽¹⁾	Application sending the message	Yes		Alphanumeric string
destinationURI ⁽¹⁾	Destination for the message	Yes	Yes	Yes
activityStatus	Status of the response operation	No	Yes	Yes
communicationPattern	Message communication pattern	Yes	Yes	SimpleResponse
communicationStyle	Message communication style	Yes	Yes	RPC

(1 of 2)

Element	Description	Request	Response	Values
timestamp	Date and time when the message was created	No	Yes	Date and time in MTOSI UTC format: <code>yyyyMMddHHmmss.S[Z {+ -}HHMm]</code> See section 3.4.1 for more information about the MTOSI UTC format.

(2 of 2)

Note

- (1) A value for this element is required in the NBI operation request, but the value is not validated by the 5529 IDM. Any value can be added for this element and the operation can still be successful.

6.9.2 Body

Table 36 describes the elements used in a standard SOAP envelope body of an exportNetwork request message.

Table 36 exportNetwork SOAP XML request elements

Element	Required	Description	Value
exportNetwork	—	Container element for the request information	—
profile	Yes	The name of the export profile to be used	String For example: my.export.profile
fileTransferType	Yes	The file transfer type to be used	FileTransferType_T Values are: <ul style="list-style-type: none"> NONE FTP SFTP
targetList	No	Container element for the list of target NEs. This element is optional and can contain no NEs, or one or more NEs up to a specified maximum. The maximum number of NEs allowed is set in the NBI settings.	—
name	No	Container element for a managed domain name and an NE name	—
mdNm	No	Name of the managed domain; that is, the name of the 5520 AMS that manages the NE	Alphanumeric string
meNm	No	Name or IP address of the managed element; that is, the name or IP address of the NE supervised by the 5520 AMS. The IP address supports the IPv4 or IPv6 format.	Alphanumeric string
remoteHost	Not required if file transfer type is NONE Required for FTP and SFTP file transfer	The hostname or IP address of the remote host to send the file to	String For example: server2

(1 of 2)

Element	Required	Description	Value
directory	Not required if file transfer type is NONE Required for FTP and SFTP file transfer	The directory location for file transfer	String For example: /var/tmp
username	Not required if file transfer type is NONE Required for FTP and SFTP file transfer	The FTP or SFTP username for the remote host	String
password	Not required if file transfer type is NONE Required for FTP file transfer Required for SFTP file transfer that uses password as the authentication method Do not send password tag for SFTP file transfer when SFTP key-based authentication is being used	The FTP or SFTP password for the remote host	String
compression	Not required if file transfer type is NONE Required for FTP and SFTP file transfer	Indicates whether the output file should be compressed before transfer	CompressionType_T Values are: <ul style="list-style-type: none"> • NO_COMPRESSION • GZIP

(2 of 2)

Table 37 describes the elements used in the SOAP envelope body of an exportNetwork response message. The SOAP envelope body of an exportNetwork response message contains the operation name and the requestId element.

Table 37 exportNetwork SOAP XML response elements

Element	Description	Value
exportNetworkResponse	Container element for the response information	—
requestId	The request ID of the operation. The ID uniquely identifies the request within a 5520 AMS cluster; the value is unique even after server restarts.	0 to 9223372036854775807

6.10 exportNetwork output file example

This sections provides an example of an exportNetwork output file. See section 4.4.1 for more information about the output file.

In the following example of two rows from an output file, including the header information row the precedes them, the NE name is DETECTED_AT_123.249.33.190 (first column on the left). The object type is Slot (second column from the left), which was an object selected for export. The object names are DETECTED_AT_123.249.33.190:R1.S1.NTA and DETECTED_AT_123.249.33.190:R1.S1.LT8 (third column from the left). The remaining columns are the values for the Slot object attributes that were selected for export: Board Type (NANT-E), Vendor Name (ALCL), Serial Number (AA1118ZA060), Available Status (Available and Not Installed), and Primary Service State (IS-NR and OOS-AU). If there is no attribute value to display, the field is blank.

```
#NE Name, Object Type, Object Name, eqptBoardInventoryTypeName,
eqptBoardInventoryAlcatelCompanyId, eqptBoardInventorySerialNumber,
eqptBoardAvailStatus, primaryServiceState
```

```
DETECTED_AT_123.249.33.190, Slot, DETECTED_AT_123.249.33.190:R1.S1.NTA,
NANT-E, ALCL, AA1118ZA060, Available, IS-NR
```

```
DETECTED_AT_123.249.33.190, Slot, DETECTED_AT_123.249.33.190:R1.S1.LT8, , ,
, Not Installed, OOS-AU
```

In the example output file that follows, the following object and attributes were selected for export with ALL selected as the NE family:

- NE
 - Active Software Version (activeSoftwareVersionAndRelease)
 - IP Address (ipAddress)
 - Name (name)
 - Supervision State (supervision)

In the example output file that follows, the objects and attributes that are listed in Table 38 were selected for export from the ISAM/FTTB/FTTN NE family.

Table 38 Objects and attributes selected for export for ISAM/FTTB/FTTN

Object	Attributes
EFM Bridge Port	Operational State (ifOperStatus)
IP Address (IHUB)	IP Address (vRialnetAddress)
IP Interface (IHUB)	VLAN ID (vRtrIfEncapValue)
L2CP Partition	Partition
L2CP Session	BRAS IP Address (L2CP Current Session-BRAS Parameters) BRAS IP Address (L2CP Session Configuration-Configuration) DSLAM Partition ID (L2CP Current Session-DSLAM Parameters) DSLAM Partition ID (L2CP Session Configuration-Configuration)
NE System	Contact (sysContact) Description (eqptHolderDescription)

(1 of 2)

Object	Attributes
SHDSL Span	Administrative State (General-States) Administrative State (General-States-Span) Customer ID Operational State (General-States)
SIP Voice Port	Administrative State (sipTerminationAdminStatus) Directory Number (sipTerminationDn) Line ID (sipTerminationLineId) Operational State (sipTerminationOperStatus)
Slot	Available State (eqptBoardAvailStatus) Board Type (eqptBoardInventoryTypeName) Primary Service State (primaryServiceState) Serial Number (eqptBoardInventorySerialNumber) Vendor Name (General-Remote Inventory Info) (eqptBoardInventoryAlcatelCompanyId)
Subrack	Available State (eqptHolderAvailStatus)
VCL Bridge Port	Operational State
XDSL Port	Administrative State (ifAdminStatus) Customer ID (asamIfExtCustomerId) Operational State (ifOperStatus) XDSL Line Service Profile (serviceProfile)

(2 of 2)

If an object is selected for export and it does not exist on any of the selected NEs, there is no row provided for that selected object (for example, L2CP Session object).

See the 5529 IDM NE support plug-in attributes guides to match the 5529 IDM NBI attribute names and values to the 5529 IDM GUI attribute names and values.

```
#NE Name, Object Type, Object Name, name, ipAddress,
activeSoftwareVersionAndRelease, supervision,
```

```
DETECTED_AT_123.249.33.190, NE, DETECTED_AT_123.249.33.190,
DETECTED_AT_123.249.33.190, 123.249.33.190, OSWPAA52.429 (Unmapped),
Supervised,
```

```
#NE Name, Object Type, Object Name, sysContact, eqptHolderDescription, , ,
DETECTED_AT_123.249.33.190, NE System, DETECTED_AT_123.249.33.190, Nokia, ,
, ,
```

```
#NE Name, Object Type, Object Name, eqptHolderAvailStatus, , , ,
```

```
DETECTED_AT_123.249.33.190, Subrack, DETECTED_AT_123.249.33.190:R1.S1,
Available, , , ,
```

```
DETECTED_AT_123.249.33.190, Subrack, DETECTED_AT_123.249.33.190:R1.S2, Not
Installed, , , ,
```

```
DETECTED_AT_123.249.33.190, Subrack, DETECTED_AT_123.249.33.190:R1.S3, Not
Installed, , , ,
```

```

DETECTED_AT_123.249.33.190, Subrack, DETECTED_AT_123.249.33.190:R1.S4, Not
Installed, , , ,

(...)

DETECTED_AT_123.249.33.190, Subrack, DETECTED_AT_123.249.33.190:R7.S1, Not
Installed, , , ,

DETECTED_AT_123.249.33.190, Subrack, DETECTED_AT_123.249.33.190:R7.S2, Not
Installed, , , ,

DETECTED_AT_123.249.33.190, Subrack, DETECTED_AT_123.249.33.190:R7.S3, Not
Installed, , , ,

DETECTED_AT_123.249.33.190, Subrack, DETECTED_AT_123.249.33.190:R7.S4, Not
Installed, , , ,

#NE Name, Object Type, Object Name, eqptBoardInventoryTypeName,
eqptBoardInventoryAlcatelCompanyId, eqptBoardInventorySerialNumber,
eqptBoardAvailStatus, primaryServiceState

DETECTED_AT_123.249.33.190, Slot,
DETECTED_AT_123.249.33.190:R1.S1.ACU/NTIO, , , , Not Installed, OOS-AU

DETECTED_AT_123.249.33.190, Slot, DETECTED_AT_123.249.33.190:R1.S1.NTA,
NANT-E, ALCL, AA1118ZA060, Available, IS-NR

DETECTED_AT_123.249.33.190, Slot, DETECTED_AT_123.249.33.190:R1.S1.LT8, , ,
, Not Installed, OOS-AU

DETECTED_AT_123.249.33.190, Slot, DETECTED_AT_123.249.33.190:R1.S1.NTB, , ,
, Not Installed, OOS-AU

DETECTED_AT_123.249.33.190, Slot, DETECTED_AT_123.249.33.190:R1.S1.LT1,
NVPS-A, ALCL, CP065006473, Available, IS-NR

DETECTED_AT_123.249.33.190, Slot, DETECTED_AT_123.249.33.190:R1.S1.LT2, , ,
, Not Installed, OOS-AU

(...)

DETECTED_AT_123.249.33.190, Slot, DETECTED_AT_123.249.33.190:R1.S1.LT7, , ,
, Not Installed, OOS-AU

#NE Name, Object Type, Object Name, ifAdminStatus, ifOperStatus,
asamIfExtCustomerId, serviceProfile,

DETECTED_AT_123.249.33.190, XDSL Port,
DETECTED_AT_123.249.33.190:R1.S1.LT5.P1, Locked, Down, available, None,

DETECTED_AT_123.249.33.190, XDSL Port,
DETECTED_AT_123.249.33.190:R1.S1.LT5.P10, Locked, Down, available, None,

DETECTED_AT_123.249.33.190, XDSL Port,
DETECTED_AT_123.249.33.190:R1.S1.LT5.P11, Locked, Down, available, None,

(...)

```

```
DETECTED_AT_123.249.33.190, XDSL Port,  
DETECTED_AT_123.249.33.190:R1.S1.LT5.P2, Locked, Down, available, None,  
  
DETECTED_AT_123.249.33.190, XDSL Port,  
DETECTED_AT_123.249.33.190:R1.S1.LT5.P20, Locked, Down, available, None,  
  
DETECTED_AT_123.249.33.190, XDSL Port,  
DETECTED_AT_123.249.33.190:R1.S1.LT5.P21, Locked, Down, available, None,  
  
(...)  
  
DETECTED_AT_123.249.33.190, XDSL Port,  
DETECTED_AT_123.249.33.190:R1.S1.LT5.P3, Locked, Down, available, None,  
  
DETECTED_AT_123.249.33.190, XDSL Port,  
DETECTED_AT_123.249.33.190:R1.S1.LT5.P30, Locked, Down, available, None,  
  
DETECTED_AT_123.249.33.190, XDSL Port,  
DETECTED_AT_123.249.33.190:R1.S1.LT5.P31, Locked, Down, available, None,  
  
(...)  
  
DETECTED_AT_123.249.33.190, XDSL Port,  
DETECTED_AT_123.249.33.190:R1.S1.LT5.P4, Locked, Down, available, None,  
  
DETECTED_AT_123.249.33.190, XDSL Port,  
DETECTED_AT_123.249.33.190:R1.S1.LT5.P40, Locked, Down, available, None,  
  
DETECTED_AT_123.249.33.190, XDSL Port,  
DETECTED_AT_123.249.33.190:R1.S1.LT5.P41, Locked, Down, available, None,  
  
(...)  
  
DETECTED_AT_123.249.33.190, XDSL Port,  
DETECTED_AT_123.249.33.190:R1.S1.LT5.P5, Locked, Down, available, None,  
  
DETECTED_AT_123.249.33.190, XDSL Port,  
DETECTED_AT_123.249.33.190:R1.S1.LT5.P6, Locked, Down, available, None,  
  
DETECTED_AT_123.249.33.190, XDSL Port,  
DETECTED_AT_123.249.33.190:R1.S1.LT5.P7, Locked, Down, available, None,  
  
DETECTED_AT_123.249.33.190, XDSL Port,  
DETECTED_AT_123.249.33.190:R1.S1.LT5.P8, Locked, Down, available, None,  
  
DETECTED_AT_123.249.33.190, XDSL Port,  
DETECTED_AT_123.249.33.190:R1.S1.LT5.P9, Locked, Down, available, None,  
  
DETECTED_AT_123.249.33.190, XDSL Port,  
DETECTED_AT_123.249.33.190:R1.S1.LT6.P1, Locked, Down, available, None,  
  
DETECTED_AT_123.249.33.190, XDSL Port,  
DETECTED_AT_123.249.33.190:R1.S1.LT6.P10, Locked, Down, available, None,  
  
DETECTED_AT_123.249.33.190, XDSL Port,  
DETECTED_AT_123.249.33.190:R1.S1.LT6.P11, Locked, Down, available, None,  
  
(...)
```



```

DETECTED_AT_123.249.33.190, XDSL Port,
DETECTED_AT_123.249.33.190:R1.S1.LT7.P9, Locked, Down, available, None,

#NE Name, Object Type, Object Name, sipTerminationAdminStatus,
sipTerminationOperStatus, sipTerminationDn, sipTerminationLineId,

DETECTED_AT_123.249.33.190, SIP Voice Port,
DETECTED_AT_123.249.33.190:R1.S1.LT2.P1, Locked, Down, 43, ,

#NE Name, Object Type, Object Name, ifOperStatus, , , ,

DETECTED_AT_123.249.33.190, EFM Bridge Port,
DETECTED_AT_123.249.33.190:R1.S1.LT7.P1, , , , ,

DETECTED_AT_123.249.33.190, EFM Bridge Port,
DETECTED_AT_123.249.33.190:R1.S1.LT7.P8, , , , ,

#NE Name, Object Type, Object Name, vRtrIfEncapValue, , , ,

DETECTED_AT_123.249.33.190, IP Interface (IHUB),
DETECTED_AT_123.249.33.190:IHUB:RT1.ITF1, None, , , ,

DETECTED_AT_123.249.33.190, IP Interface (IHUB),
DETECTED_AT_123.249.33.190:IHUB:RT1.ITF2, 10, , , ,

#NE Name, Object Type, Object Name, vRiaInetAddress, , , ,

DETECTED_AT_123.249.33.190, IP Address (IHUB),
DETECTED_AT_123.249.33.190:IHUB:RT1.ITF2.ID1, 123.249.33.190, , , ,

```

6.11 query SOAP envelope

This section provides reference information about the standard SOAP envelope for query messages.

6.11.1 Header

Table 39 describes the elements used in the query SOAP envelope header, and indicates the element relevance to each type of SOAP XML message (request or response).

Table 39 query SOAP envelope header

Element	Description	Request	Response	Values
activityName ⁽¹⁾	Activity (operation) name	Yes	Yes	query
msgName ⁽¹⁾	Message name in the WSDL file	Yes	Yes	query

(1 of 2)

Element	Description	Request	Response	Values
msgType	The message type	Yes	Yes	REQUEST RESPONSE ERROR
senderURI ⁽¹⁾	Application sending the message	Yes	Yes	Alphanumeric string
destinationURI ⁽¹⁾	Destination for the message	Yes	Yes	Alphanumeric string
correlationId	Arbitrary number assigned to the request to correlate it with the response and differentiate it from other requests sent by the OSS client	Yes	Yes	Integer
activityStatus	Status of the response operation	No	Yes	SUCCESS FAILURE WARNING
communicationPattern	Message communication pattern	Yes	Yes	MultipleBatchResponse
communicationStyle	Message communication style	Yes	Yes	RPC
requestedBatchSize	The logical size of the batch for a multi-response communication pattern	Yes	Yes	0 to 1500 Default is 1500.
batchSequenceNumber	Identifies the batch sequence number in a multiple response communication pattern	No	Yes	0 to integer (last batch number)
batchSequenceEndOfReply	Indicates the end of a batch response when the value is true	No	Yes	true false
iteratorReferenceURI	Data retrieval iterator	No	Yes	A string of numeric characters unique for each operation
timestamp	Date and time when the message was created	Yes	Yes	Date and time in MTOSI UTC format: yyyyMMddHHmmss.S[Z]{+ -}HHMM See section 3.4.1 for more information about the MTOSI UTC format.

(2 of 2)

Note

- ⁽¹⁾ A value for this element is required in the NBI operation request, but the value is not validated by the 5529 IDM. Any value can be added for this element and the operation can still be successful.

6.11.2 Body

Table 40 describes the elements used in a standard SOAP envelope body of a query request message.

Table 40 query SOAP XML request elements

Element	Required	Description	Value
query	—	Container element for the request information	—

(1 of 3)

Element	Required	Description	Value
baseObjectList	No	Container element for the list of objects. This element can contain no objects, or one or more objects up to a specified maximum. When no objects are specified, the value is the whole network. The maximum number of objects allowed is set in the NBI settings.	—
baseObject	No	Container element for a managed domain name, NE name, and object type name	—
mdNm	No	Name of the managed domain; that is, the name of the 5520 AMS that manages the NE	Alphanumeric string
meNm	No	Name or IP address of the managed element; that is, the name or IP address of the NE supervised by the 5520 AMS. The IP address supports the IPv4 or IPv6 format.	Alphanumeric string
propNm	No	The object type and object name. For the list of supported values for this element, see the 5529 IDM NE support plug-in attributes guides.	String For example: /type=Slot/R1.S1.LT4
level	No	The query retrieval level: <ul style="list-style-type: none"> BASE_OBJECT retrieves the objects provided in baseObjectList. WHOLE_SUBTREE retrieves the objects in the baseObjectList and all the objects under the baseObjectList. The value cannot be BASE_OBJECT when the baseObjectList value is the whole network.	BASE_OBJECT (default) WHOLE_SUBTREE
source	No	Indicates whether the object data should be retrieved from the 5520 AMS database or the network NETWORK is only applicable when the level value is BASE_OBJECT.	AMS (default) NETWORK
filterlist	No	Container element for the list of filters	—
filter	No	Container element for the filter attributes	—
type	No	Attribute type When a value is not provided, the baseObject type is used	String For example: ONT
attributefilterList	No	Container element for the list of attribute filters	—
attributefilterInfo	No	Container element for the attribute filter information	—
name	No	Attribute name For the list of supported values for this element, see the 5529 IDM NE support plug-in attributes guides.	String For example: bponOntSerialNumber
value	No	Attribute value For the list of supported values for this element, see the 5529 IDM NE support plug-in attributes guides.	String For example: 100
logicalOperator	No	Logical operator for filtering the attribute values	AND OR

(2 of 3)

Element	Required	Description	Value
relationalOperator	No	Relational operator for filtering the attribute values	EQUALS GREATER GREATER_EQUAL LESS LESS_EQUAL NOT_EQUAL

(3 of 3)

Table 41 describes the elements used in the SOAP envelope body of a query response message.

Table 41 query SOAP XML response elements

Element	Description	Value
queryResponse	Container element for the query response information	—
queryObjectData	Container element for the query response object data	—
queryObject	Container element for the query response object	—
name	Container element for a managed domain name, NE name, and object type name	—
mdNm	Name of the managed domain; that is, the name of the 5520 AMS that manages the NE	Alphanumeric string
meNm	Name or IP address of the managed element; that is, the name or IP address of the NE supervised by the 5520 AMS. The IP address supports the IPv4 or IPv6 format.	Alphanumeric string
propNm	The object type and object name. For the list of supported values for this element, see the 5529 IDM NE support plug-in attributes guides.	String For example: /type=Slot/R1.S1.LT4
vendorExtensions	Container object for vendor extensions attributes	—
package	Container object for a set of NameAndStringValue elements	—
NameAndStringValue	Container object for the attribute name and value	—
name	Attribute name	String For example: eqptSlotPowerStatus
value	Attribute value	String For example: powerUp

6.12 JMS notification SOAP envelope

This section provides reference information about the SOAP envelope of a JMS notification message.

6.12.1 Header

Table 42 describes the elements used in the SOAP envelope header of a JMS notification message.

See the TMF MTOSI documentation for the definition of each SOAP envelope element.

Table 42 JMS notification SOAP envelope header

Element	Description	Value
tmf854:activityName	Notification name	notify
tmf854:msgName	Message name in the WSDL file	notify
tmf854:msgType	The message type	NOTIFICATION ERROR
tmf854:senderURI	Application sending the message	Alphanumeric string
tmf854:destinationURI	Destination for the message	Alphanumeric string
tmf854:communicationPattern	Sets the communication pattern	Notification
tmf854:communicationStyle	Sets the communication style	MSG
tmf854:timestamp	Date and time when the message was created	Date and time in MTOSI UTC format: <code>yyyyMMddHHmmss.S[Z {+ -}HHMm]</code> See section 3.4.1 for more information about the MTOSI UTC format.

6.12.2 Body

Table 43 describes the elements used in the SOAP envelope body of a JMS notification message.

Table 43 JMS notification SOAP envelope body

Element	Description	Type or value
tmf854:notify	Container element for the notification	—
tmf854:topic	JMS subscription topic	topic/Inventory
tmf854:message	Container element for the notification message	—
tmf854:ObjectCreation	Container element for the object creation event notification	—
tmf854:ObjectDeletion	Container element for the object deletion event notification	—
tmf854:eventInfo	Container element for the event information	—

(1 of 2)

Element	Description	Type or value
tmf854:notificationId	Notification ID	Integer
tmf854:objectName	Container element that includes the object name attributes: mdNm and meNm (alphanumeric strings)	—
tmf854:objectType	Object type	OT_EQUIPMENT OT_EQUIPMENT_HOLDER
tmf854:osTime	Date and time when the notification was created	Date and time in MTOSI UTC format: <code>yyyyMMddHHmmss.S[Z {+ -}HHMm]</code> See section 3.4.1 for more information about the MTOSI UTC format.
tmf854:vendorExtensions	Object vendorExtensions attributes The tmf854:vendorExtensions element applies only to object creation notifications	—

(2 of 2)

Table 44 describes the elements used in the SOAP envelope body of a JMS notification message for the exportNetwork operation.

Table 44 JMS notification SOAP envelope body for exportNetwork

Element	Description	Type or value
tmf854:notify	Container element for the notification	—
tmf854:topic	JMS subscription topic	topic/Inventory
tmf854:message	Container element for the notification message	—
tmf854:VendorNotification	Container element for the network export event notification	VendorNotification_T
tmf854:notificationId	Notification ID	String
tmf854:VendorNotificationType	The type of notification: success or failure	NetworkExportComplete NetworkExportFailed
tmf854:vendorExtensions	Object type	VendorNotificationExt_T
tmf854:vendorExtensions	Object vendorExtensions attributes	—
tmf854:package	Vendor Extensions attributes	PackageOfNamedTypeAndValues_T
package	Container object for a set of NameAndStringValue elements	—
NameAndStringValue	Container object for the name and value	—
tmf854:name	Object name	Alphanumeric string
tmf854:value	Object value	Alphanumeric string

Table 45 describes the elements used in the SOAP envelope body of a JMS notification for a heartbeat.

Table 45 JMS notification SOAP envelope body for heartbeat

Element	Description	Type or value
tmf854:notify	Container element for the notification	—
tmf854:topic	JMS subscription topic	topic/Inventory
tmf854:message	Container element for the notification message	—
tmf854:Heartbeat	Container element for the heartbeat event notification	—
tmf854:notificationId	Notification ID	String
tmf854:objectType	Object type	OT_OS
tmf854:objectName	Container element for the object name attribute	—
tmf854:osNm	Object name	<osNm>Alcatel-Lucent/IDM</osNm>
tmf854:osTime	Date and time when the notification was created	Date and time in MTOSI UTC format: <i>yyyyMMddHHmmss.S[Z {+ -}HHMM]</i> See section 3.4.1 for more information about the MTOSI UTC format.

6.13 SOAP envelopes for shortcut operations

This section provides reference information about the SOAP envelope for the getManagedElement, getEquipment, and getTP shortcut operations.

6.13.1 Header

Table 46 describes the elements used in a standard SOAP envelope header for a shortcut operation message.

See the TMF MTOSI documentation for the definition of each SOAP header element.

Table 46 SOAP envelope header for shortcut operations

Element	Description	Values	Operations
activityName ⁽¹⁾	Operation name	getManagedElement	getManagedElement
		getEquipment	getEquipment
		getTP	getTP

(1 of 2)

Element	Description	Values	Operations
msgName ⁽¹⁾	Message name in the WSDL file	getManagedElement tmf854.v1:getManagedElementResponse	getManagedElement
		getEquipment tmf854.v1:getEquipmentResponse	getEquipment
		getTP tmf854.v1:getTPResponse	getTP
messageType	The message type	REQUEST RESPONSE ERROR	All
senderURI ⁽¹⁾	Application sending the message	Alphanumeric string	All
destinationURI ⁽¹⁾	Destination for the message	Alphanumeric string	All
activityStatus	Status of the response operation	SUCCESS FAILURE WARNING	All
correlationId	Arbitrary number assigned to the request to correlate it with the response and differentiate it from other requests sent by the OSS client	Integer	All
communicationPattern	Sets the communication pattern	SimpleResponse	All
communicationStyle	Sets the communication style	RPC	All
timestamp	Date and time when the message was created	Date and time in MTOSI UTC format: yyyyMMddHHmmss.S[Z {+ -}HHMM] See section 3.4.1 for more information about the MTOSI UTC format.	All

(2 of 2)

Note

- ⁽¹⁾ A value for this element is required in the NBI operation request, but the value is not validated by the 5529 IDM. Any value can be added for this element and the operation can still be successful.

6.13.2 Body

Table 47 describes the elements used in a standard SOAP envelope body of a request message for a shortcut operation.

Table 47 SOAP XML request elements for shortcut operations

Element	Description	Operation
getManagedElement	This container element includes the managed element name: the mdNm and meNm attributes	getManagedElement

(1 of 2)

Element	Description	Operation
getEquipment	Container element for equipmentOrHolderName	getEquipment
equipmentOrHolderName	This element contains the equipment or holder object name: the mdNm, meNm, ehNm, and eqNm attributes	
getTP	Container element for tpName	getTP
tpName	This element contains the tp object name: the mdNm, meNm, and ptpNm/ftpNm attributes	

(2 of 2)

Table 48 describes the object elements used in the SOAP envelope body of a response message for a shortcut operation. Each object contains a set of attributes that provides inventory data about the object, including the vendorExtensions attributes.

For information about the subset of NE object attributes that is supported by the NBI, see the HTML files that are provided with the 5529 IDM. See the *5529 Enhanced Applications Release Notice* for information about how to access these files.

Table 48 SOAP XML response elements for shortcut operations

Element	Description	Operation
getManagedElementResponse	Container element for the managed element inventory data	getManagedElement
me	Managed element object that contains inventory attributes for the NE, including vendorExtensions attributes	
getEquipmentResponse	Container element for the equipment holder or equipment inventory data	getEquipment
equip	Container element for the eh or eq object	
eh	Equipment holder object containing inventory attributes, including vendorExtensions attributes	
eq	Equipment object containing inventory attributes, including vendorExtensions attributes	
getTPResponse	Container element for the physical termination point or floating termination point inventory data	getTP
ptp	Physical termination point object containing inventory attributes, including vendorExtensions attributes	
as	Association object containing inventory attributes	
ctp	Connection Termination Point object containing inventory data including vendorExtensions attributes	
sfp	Small Factor Pluggable object containing inventory data	
ftp	Floating termination point object containing inventory attributes, including vendorExtensions attributes	

Customer document and product support



Customer documentation

[Customer Documentation Welcome Page](#)



Technical Support

[Product Support Portal](#)



Documentation feedback

[Customer Documentation Feedback](#)

