



5529 Access Provisioning Center, Release 9.6.07

Northbound Interface Guide

3JL-02001-BRAA-RKZZA

Edition 02

December 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	Audience.....	13
1.2	Related documentation.....	13
1.3	Conventions used in this guide.....	14
1.3.1	Important information.....	15
1.3.2	Procedures with options or substeps.....	15
1.4	Multiple PDF file search.....	16
Getting started		
2	What's new	21
2.1	What's new in Release 9.6.07	21
2.2	What's new in Release 9.6.05	22
2.3	What's new in Release 9.6.03	24
3	5529 APC Northbound interface overview	29
3.1	5529 APC overview	29
3.1.1	Web services	29
3.1.2	Notification interface	30
3.1.3	Architecture	31
3.2	Web service technology	32
3.2.1	WSDL file.....	32
3.2.2	Verifying the 5529 APC service access point.....	34
3.3	Northbound interface security.....	34
3.3.1	Changes to HTTP login error responses	34
3.3.2	MSAI services.....	35
3.3.3	Services supporting the notification interface	35
3.4	Collaborative service provisioning	36
3.5	5529 APC in high-performance networks.....	37
3.6	5529 APC and high subscriber density configurations	37
3.7	Developer support	38
5529 APC Northbound interface		
4	Service fulfillment web services	41
4.1	Service fulfillment web services overview	42
4.2	Web service parameters.....	43
4.2.1	objectName	44
4.2.1.1	Target object ID address extension.....	44
4.2.1.2	Special cases when target object ID address may be unclear for NBI operations.....	45
4.2.2	templateName	46
4.2.3	templateVersion.....	46
4.2.4	argument	47
4.2.4.1	Reusing shared virtual arguments.....	48
4.2.5	instanceLabel	48
4.2.6	operationInitiator.....	49
4.2.7	ResultIndicator.....	49

4.3	Managing errors	50
4.3.1	HTTP exceptions	50
4.3.2	5529 APC exceptions	51
4.3.3	Profile name error message	54
4.4	Web services and operations	55
4.4.1	Web service inventory	55
4.4.2	Service provisioning	56
4.4.3	Bulk execution	57
4.4.4	Service inventory	57
4.4.5	Audit and reconciliation	58
4.4.6	Service state verification	59
4.4.7	Server availability verification	59
4.5	NBI operations supported by NE type and release	59
4.6	audit operation	60
4.7	auditNode operation	62
4.8	auditPort operation	63
4.8.1	Audit misalignment for base stacks with applied templates	65
4.9	check operation	66
4.10	clean operation	68
4.11	configure operation	70
4.12	dbClean operation	71
4.13	dbUnconfigure operation	72
4.14	discoverAllServices operation	72
4.15	execute operation	77
4.16	File-based operations	79
4.17	get operation	79
4.18	getApplicableTemplates operation	82
4.19	getChangeHistory operation	83
4.20	getConfiguredServices operation	84
4.21	getConfiguredTemplate operation	86
4.22	getConfiguredTemplateName operation	88
4.23	getConfiguredTemplates operation	88
4.24	getLogicalPorts operation	90
4.25	getNodeAuditResult operation	91
4.26	getResult operation	93
4.27	getResults operation	94
4.28	getSystemHealthInfo operation	94
4.29	getTemplate operation	94
4.30	getTemplateMetaData operation	96
4.30.1	\r character in operation response	99
4.31	getTemplateName operation	100
4.32	importTemplate operation	101
4.33	migrate operation	102
4.33.1	migrationStrategy parameter	104
4.33.2	Retain configured values	104
4.33.3	Migrating service stacks	105
4.33.4	Smart migrate	105
4.33.5	migrate with NO_REBUILD and -No-Value- as the optional attribute default value	106

4.33.6	Maintain administrative state during service migration	106
4.33.7	Migrating a template with arguments that depend on a specific argument value being set	108
4.33.8	Rollback	108
4.34	modify operation	108
4.34.1	forceModify	109
4.34.2	modify operation without forceModify and with -No-Value- as the optional attribute default value	110
4.34.3	Using forceModify after excluding attributes from audit	111
4.34.4	modify operation and shared services	111
4.34.4.1	Modifying shared arguments	111
4.34.4.2	Modifying a virtual argument mapped to an argument on a shared service	112
4.35	modifyTemplateInstance operation	112
4.36	move operation	113
4.36.1	move with -No-Value- as the optional attribute default value	114
4.37	reapplyAllServices operation	115
4.37.1	reapplyAllServices with NO_REBUILD and -No-Value- as the optional attribute default value	117
4.38	reapplyService operation	117
4.38.1	reapplyService with NO_REBUILD and -No-Value- as the optional attribute default value	119
4.38.2	reapplyService for an ONT with a single service and misaligned non-modifiable attributes	119
4.39	resume operation	119
4.40	suspend operation	121
4.41	unconfigure operation	124
5	Service inventory	125
5.1	Service inventory and export services	125
6	Collaborative service provisioning	127
6.1	Collaborative service provisioning overview	127
6.1.1	JMS client application tasks	128
6.1.2	JMS ports	129
6.1.3	JMS application parameters	129
6.1.4	Libraries	130
6.1.5	Cluster JMS	130
6.2	Configuration change notifications	131
6.2.1	Special notification messages	132
6.2.2	Filtering notification messages	133
6.2.3	Heartbeat notifications	135
6.2.3.1	Date and time in MTOSI UTC format	136
6.3	Web services	137
6.3.1	getClients operation	137
6.3.2	getNotificationTopic	138

Reference

7	Examples of web service operations, responses, and bulk request files	141
7.1	Examples of web service operations	141
7.1.1	Examples of the configure operation	141
7.1.2	Examples of template migration operations	143
7.1.3	Example of the getTemplateMetaData operation	147
7.1.4	Example of the getConfiguredServices operation	147
7.2	Examples of web service operation responses	148
7.2.1	audit operation responses	148
7.2.1.1	Successful audit operation response.....	148
7.2.1.2	Audit operation response with misalignments	149
7.2.1.3	Audit operation response with service state misalignments	150
7.2.2	auditNode operation responses.....	151
7.2.2.1	Clean port.....	151
7.2.2.2	Port not clean	152
7.2.2.3	Target object missing in NE.....	152
7.2.2.4	Template not configured.....	152
7.2.3	auditPort operation responses.....	153
7.2.3.1	auditPort operation response for an aligned service	154
7.2.3.2	auditPort operation response for a misaligned service.....	155
7.2.3.3	auditPort operation response for misaligned service states	156
7.2.4	getConfiguredServices operation response	156
7.2.5	getLogicalPorts operation response	158
7.2.6	getNodeAuditResult operation response.....	158
7.2.7	getTemplateMetaData operation response	168
7.3	Examples of imported bulk request files.....	180
7.3.1	Example imported bulk request file for template migration.....	181
7.4	Examples of audit results	181
7.4.1	Example of an audit result for a clean port.....	182
7.4.2	Example of an audit result for a dirty port.....	182
7.4.3	Example of an audit result for an aligned port.....	182
7.4.4	Example of an audit result for a misaligned port	183
7.4.5	Example of an audit result where a port is intended to be configured but remains unconfigured	183
8	Configuration change notification message and web service examples	185
8.1	Examples of change notification messages	185
8.2	Example of a getNotificationTopic operation request and response	188
9	New ONT notification	189
9.1	New ONT notifications.....	189
9.1.1	Format of empty value element.....	190
9.2	Example of a new ONT notification event	190
10	NE reparenting notifications.....	193
10.1	NE reparenting notifications	193
10.2	Examples of NE reparenting notifications.....	193

List of figures

3	5529 APC Northbound interface overview	29
Figure 1	5529 APC architecture	31
4	Service fulfillment web services	41
Figure 2	Configure request example	47
7	Examples of web service operations, responses, and	
	bulk request files	141
Figure 3	Example of an audit result for an aligned port	182
Figure 4	Example of an audit result for a misaligned port	183

List of tables

1	Preface	13
Table 1	Related documentation.....	13
Table 2	Documentation conventions	14
2	What's new	21
Table 3	What's new in Release 9.6.07	21
Table 4	What's new in Release 9.6.05	23
Table 5	What's new in Release 9.6.03	24
3	5529 APC Northbound interface overview	29
Table 6	WSDL source files	33
Table 7	Changes to HTTP login error responses	34
4	Service fulfillment web services	41
Table 8	Web service categories	43
Table 9	Object types and paths for address extension	44
Table 10	Target object ID address special case	46
Table 11	RemoteApcException message format	51
Table 12	errorCode description	51
Table 13	Web service inventory	55
Table 14	NBI operations supported by NE type and release	59
Table 15	audit operation request parameters.....	61
Table 16	audit operation response parameters.....	61
Table 17	auditNode operation request parameters	62
Table 18	auditPort operation request parameters	64
Table 19	auditPort operation response parameters	64
Table 20	Audit misalignment for base stacks with applied templates	65
Table 21	check operation request parameters	66
Table 22	statusExpressionList status expression examples	67
Table 23	check operation response items.....	67
Table 24	clean operation request parameters.....	69
Table 25	configure operation request parameters.....	70
Table 26	dbClean operation request parameters	71
Table 27	dbUnconfigure operation request parameters	72
Table 28	Notes about the discoverAllServices operation	73
Table 29	discoverAllServices operation request parameters	76
Table 30	execute operation request parameters.....	77
Table 31	File-based operations	79
Table 32	get operation request parameters	80
Table 33	Attributes retrieved by the get operation.....	81
Table 34	get operation response parameters.....	81
Table 35	getApplicableTemplates operation request parameters	82
Table 36	getApplicableTemplates operation response parameters	83
Table 37	getChangeHistory operation response parameters	83
Table 38	getConfiguredServices operation request parameters	85
Table 39	getConfiguredServices operation response parameters	85
Table 40	getConfiguredTemplate operation request parameters	86

Table 41	getConfiguredTemplate operation response parameters	87
Table 42	getConfiguredTemplateNames operation request parameters	88
Table 43	getConfiguredTemplateNames operation response parameters	88
Table 44	getConfiguredTemplates operation request parameter	89
Table 45	getConfiguredTemplates operation response parameters	89
Table 46	getLogicalPorts operation request parameter	90
Table 47	getLogicalPorts operation response parameter	91
Table 48	getNodeAuditResult operation request parameters	91
Table 49	getNodeAuditResult operation response parameters	91
Table 50	getNodeAuditResult information	92
Table 51	getResult operation request parameters	93
Table 52	getResult and getResults operations response parameters	93
Table 53	getResults operation request parameters	94
Table 54	getTemplate operation request parameters	95
Table 55	getTemplate operation response parameters	95
Table 56	getTemplateMetaData operation request parameters	97
Table 57	argumentList parameter description	97
Table 58	getTemplateMetaData operation response parameters	97
Table 59	attributeMetaData metadata description	98
Table 60	getTemplateNames operation response parameters	100
Table 61	importTemplate operation request parameters	101
Table 62	migrate operation request parameters	103
Table 63	migrationStrategy parameter description	104
Table 64	Service migration scenarios and administrative state of service templates	107
Table 65	modify operation request parameters	109
Table 66	modifyTemplateInstance operation request parameters	112
Table 67	move operation request parameters	114
Table 68	reapplyAllServices operation request parameters	116
Table 69	reapplyService operation request parameters	118
Table 70	resume operation request parameters	120
Table 71	suspend operation request parameters	122
6	Collaborative service provisioning	127
Table 72	Configuration change notification message description	131
Table 73	Message properties	134
Table 74	SOAP envelope header of JMS notification for heartbeat	135
Table 75	SOAP envelope body of JMS notification for heartbeat	135
Table 76	getClients operation response description	137
Table 77	getNotificationTopic operation request description	138
Table 78	getNotificationTopic operation response description	138
9	New ONT notification	189
Table 79	New ONT notification event arguments	189

List of procedures

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

1 Preface

The *5529 APC Northbound Interface Guide* describes the MSAI, the 5529 APC Northbound interface, including:

- general concepts
- 5529 APC template concepts
- 5529 APC functions that are available using the Northbound interface
- collaborative service provisioning support functions, including the configuration change notification interface
- capabilities to track users who perform service provisioning

1.1 Audience

This document is mainly intended for OSS integration personnel that are involved in the automation of service fulfillment processes, such as service provisioning and activation, as well as guarantee service delivery using audit functions. The personnel may work for service provider organizations, software integration, OSS vendors, or independent software vendors.

1.2 Related documentation

See the *5529 APC Template Reference* for detailed information about the 5529 APC basic templates, which are used to create service templates. There is a guide for the plug-in of each NE type.

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 Installation and Migration Guide</i>	Information about how to install the 5520 AMS server and client, and use the feature updates function for the client
5520 AMS and 5529 Access Provisioning Center	
<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

(1 of 2)

Customer documentation	Description
5529 Access Provisioning Center	
<i>5529 APC Installation, Administration, and User Guide</i>	Information about installing and customizing the 5529 APC software, and how to navigate and use the 5529 APC GUI and 5529 APC SPFE GUI, which can be installed with the 5520 AMS client when the 5529 APC is integrated with the 5520 AMS
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

(2 of 2)

1.3 Conventions used in this guide

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

Table 2 Documentation conventions

Convention	Description	Example
Key name	Press a keyboard key	Delete
Italics	Identifies 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 or Enter key.	Press ↵
—	An em dash in a table cell indicates that there is no information or that the category is not applicable.	—
*	An asterisk is a wildcard character that means “any character” in a search argument.	*
→	Indicates that a cascading submenu results from selecting a menu item	Administration → System settings

1.3.1 Important information

The following are examples of how important information is presented in this document.



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



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



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

1.3.2 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. |
|---|-----------------------------|
-

1.4 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 5529 APC Northbound interface overview](#)

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 new 5529 APC features and enhancements added to the *5529 APC 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			
LDAP and RADIUS authentication	LDAP and RADIUS authentication is supported for the 5529 APC NBI and 5529 APC Web GUI access as of Release 9.6.07.	01	—
NEs with IPv6 address management	The 5529 APC NBI is updated to support NEs that have IPv6 address management.	01	—
ResultIndicator parameter in operation responses	The ResultIndicator parameter is an optional parameter in the operation response that explicitly states that the operation was successful.	02	Sections 4.2.7 , 4.10 , 4.11 , 4.12 , 4.13 , 4.14 , 4.28 , 4.32 , 4.33 , 4.34 , 4.35 , 4.36 , 4.37 , 4.38 , 4.39 , 4.40 , 4.41 , 7.1.1 , and 7.1.2
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
auditNode operation	Removed the NE-specific tables of 5529 APC templates that are supported by the auditNode operation because all of the templates are currently supported by the auditNode operation by default	01	Section 4.7
Configuration change notification message for Move All Services operation in SPFE	The request part of the configuration change notification message (MOVE action) for the Move All Services operation in the SPFE is updated to remove the objectName parameter and replace it with the fromPort and toPort parameters.	01	Table 72
discoverAllServices operation	Revised the note about bind templates	01	Table 28
execute operation (AUDITPORT action)	In the execute operation (AUDITPORT action), the inExcess parameter is added to the operation request. When the configuration change notification message is sent, the request part of the message is updated to include more information about the AUDITPORT action.	01	Table 30

(1 of 2)

Feature/enhancement	Description	Edition	See
getApplicableTemplates operation response parameters	Changed the description of the templateId parameter and removed the state parameter	01	Table 36
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 6.1.3
Limitation when processing template stacks with network infrastructure services	Added a note that the following NBI operations will take longer than three to six seconds to process template stacks that contain network infrastructure services: <ul style="list-style-type: none"> auditPort clean configure discoverAllServices migrate reapplyAllServices reapplyService unconfigure 	01	Sections 4.8 , 4.10 , 4.11 , 4.14 , 4.33 , 4.37 , 4.38 , and 4.41
Modifying shared arguments	Added information about modifying shared arguments	01	Section 4.34.4.1
NE name format in NBI	If an NE is managed by the 5529 APC, the NE name must not contain a colon character (:); otherwise, the NBI will not work with that NE.	01	Section 4.2.1
Notification message filtering and wildcard use	Added information about filtering message properties on heartbeat and new ONT notifications and updated wildcard information	01	Section 6.2.2
Optional template attributes set to the -No-Value- special token	Updated the description for the following operations to indicate that the functionality regarding the values of optional template attributes does not apply to the 7302 ISAM/7330 ISAM FTTN R3.7/FGN3.8: <ul style="list-style-type: none"> audit auditNode auditPort migrate modify move reapplyAllServices reapplyService 	01	Sections 4.6 , 4.7 , 4.8 , 4.33.5 , 4.34.2 , 4.36.1 , 4.37.1 , and 4.38.1
PAP groups	Added a note that the 5529 APC Northbound and notification interfaces do not take into account user-assigned PAP groups and do not check the PAP rights	01	Sections 3.1.1 and 3.4
Web service technology	Added websites that you can consult for more information about the applicable specifications	01	Section 3.2

(2 of 2)

2.2 What's new in Release 9.6.05

Table [4](#) describes the new 5529 APC features and enhancements added to the *5529 APC 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			
Changes to HTTP login error responses	The HTTP response error messages that are returned when the NBI user login validation fails have changed in Release 9.6.05.	01	Section 3.3.1
execute operation	Added a note stating that if the request contains any unsupported command, the query result will return an error.	01	Section 4.15
getClient operation	The getClient NBI operation is deprecated as of R9.6.05. Nokia recommends that you use the 5520 AMS listUser NBI operation. See the <i>5520 AMS Northbound Interface Guide</i> for more information about the listUser operation.	01	Section 6.3.1
	The getClient operation response is changed to be able to list multiple user roles for a user (client).	01	Section 6.3.1
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 APC NBI server	01	Sections 4.5 and 4.28 Table 13
	Added the server availability verification category of web service	01	Tables 8 and 13 Section 4.4.7
NE releases discontinued	Support for the 7302 ISAM xVPS and 7302/7330/7356/7360/7363/7367 ISAM R5.0 and R5.1 is discontinued at R9.6.05.	01	—
User accounts and NBI security	Updated the user account information as it relates to NBI security and the Authorization Header of the HTTP/S message in the web service requests	01	Section 3.3
Documentation changes			
audit operation request parameters	Added the audit operation request parameters	01	Table 15
Configuration change notification message description	Updated the order of the SOAP header and body elements: <ul style="list-style-type: none"> Moved the topic, notificationId, and vendorNotificationType elements from the SOAP header section to the SOAP body section Changed the order of the SOAP header elements 	01	Table 72
discoverAllServices operation	Removed the information about the 5529 APC templates that are supported by the discoverAllServices operation. See the <i>5529 APC Template Reference</i> for an NE type for information about the 5529 APC templates that are not supported by the discoverAllServices operation.	01	Section 4.14 <i>5529 APC Template Reference</i>
	Notes about the discoverAllServices operation have been moved to a table and are sorted by related functionality	01	Section 4.14
getApplicableTemplates operation response parameters	Updated the getApplicableTemplates operation response parameters	01	Table 36
getClient operation	Removed the getClient operation request and response examples	01	Chapter 8
getConfiguredServices operation	Updated the getConfiguredServices operation request and response parameters	01	Tables 38 and 39

(1 of 2)

Feature/enhancement	Description	Edition	See
getConfiguredTemplate operation	Added the getConfiguredTemplate operation request and response parameters	01	Section 4.21
getConfiguredTemplateNames operation	Added the getConfiguredTemplateNames operation request and response parameters	01	Section 4.22
getTemplateName operation response parameters	Added the getTemplateName operation response parameters	01	Table 60
Internet protocol specified in NBI request header	Added information about how the Internet protocol specified in the header of NBI requests must match the Internet protocol specified for the 5520 AMS server or servers	01	Section 3.1.1
migrate operation	Moved information about migrating a template with arguments that depend on a specific argument value being set from the <i>5529 Enhanced Applications Release Notice</i> to this guide	01	Section 4.33
NE release support	Added NE releases that are supported in R9.6.05 and that were removed in R9.6.03 because they were forward compatible	01	Tables 14 , 18 , and 19
	With the support of R5.7, the name of the 7367 ISAM SX NE is changed to 7367 ISAM SX/DX.	01	Tables 12 , 14 , and 18
	With the support of R5.7, the name of the 7362 ISAM DF NE is changed to 7362 ISAM DF/SF.	01	Tables 14 and 18
New ONT notification event arguments	Updated the information about the new ONT notification event arguments	01	Table 79
SOAP envelope header of JMS notification for heartbeat	Added the communicationStyle element	01	Table 74
Special cases when target object ID address may be unclear for NBI operations	Added information about the conditions that will make the target object ID address unclear for the configure operation and other NBI operations	01	Section 4.2.1.2

(2 of 2)

2.3 What's new in Release 9.6.03

Table [5](#) describes the new 5529 APC features and enhancements added to the *5529 APC 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			
auditNode operation	Added ISAM SIP Shared Line as a supported template for the auditNode operation for Release 5.6 or later 7302/7330/7356 /7360/7362/7363/7367 ISAM NEs	01	Table 18

(1 of 4)

Feature/enhancement	Description	Edition	See
discoverAllServices operation	Removed note about discovery of services containing a VLAN template not being supported. Added a note that describes additional network infrastructure object types and their children templates that can be discovered for 7302/7330/7356/7360/7363/7367 ISAM R5.4 or later NEs when discovering services from the NE.	01	Section 4.14
	Added ISAM SIP Shared Line as a supported template for the discoverAllServices operation for Release 5.6 or later 7302/7330/7356 /7360/7362/7363/7367 ISAM NEs	01	Table 34
HTTPS interface	The HTTPS interface is the default interface. The HTTP interface is disabled by default.	01	Sections 3.2.1 , 3.2.2 , 3.3 , and 3.4
JBoss libraries for JMS client application	The name of the apc-oss-client-1_release-version.jar library is changed to apc-oss-client-release-version.jar. The jboss-client.jar library is removed. The following libraries are new: <ul style="list-style-type: none"> • <code>axs-encrption-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-log4j12-1.6.1.jar</code> • <code>wildfly-client-all.jar</code> • <code>xbean-2.6.0.jar</code> 	01	Section 6.1.4
Login mechanism	Added a note stating that the 5520 AMS implements a rate-limiting login mechanism	01	Section 3.2.1
Maintain administrative state during service migration	When migrating a service, the 5529 APC maintains the administrative state of the service template so that the administrative state of the new service is the same as the old service.	01	Section 4.33.6
Documentation changes			
\r character in getTemplateMetaData operation response	Added information about the \r character in the getTemplateMetaData operation response and how it changed in 5529 APC Release 9.6 following the upgrade to an AXIS2 library	01	Section 4.30.1
audit operation	Updated the example of a successful audit operation response	01	Section 7.2.1.1
	Updated the example of an audit operation response that reports misalignments	01	Section 7.2.1.2
	Updated the example of an audit operation response that reports a misaligned service template state and an audit operation response that reports a misaligned admin state for a service component.	01	Section 7.2.1.3
auditNode operation	Added note indicating that the AuditNode response can be misaligned if the template attribute is not present or is not supported on a board	01	Section 4.7
	Added ISAM E1 Port, ISAM L2 Switch Cross Connect, and ISAM SIP ISDN Termination as supported templates for some 7302/7330/7356 /7360/7362/7363/7367 ISAM NEs	01	Table 18
auditPort operation	Updated the example of an auditPort operation response for an aligned service	01	Section 7.2.3.1
	Updated the example of an auditPort operation response for a misaligned service	01	Section 7.2.3.2

(2 of 4)

Feature/enhancement	Description	Edition	See
AVG plug-in	Moved the AVG plug-in information to the <i>5529 APC Installation, Administration, and User Guide</i> , and removed the information about the interfaces required for developing AVG	01	<i>5529 APC Installation, Administration, and User Guide</i>
Before you proceed	Changed the title of the section to reflect that it provides developer support information	01	Section 3.7
clean operation	Added note indicating that the clean operation will fail if a 5520 AMS search tone test is in progress on an SHDSL span object	01	Section 4.10
configure operation	The 5529 APC does not support configuring the value of a mandatory attribute to be -No-Value-.	01	Section 4.11
	Updated the request and response (success and failure) examples for the configure operation	01	Section 7.1.1
discoverAllServices operation	Added note that the discoverAllServices operation may fail if there are multiple matched rules in some cases	01	Section 4.14
	Added ISAM E1 Port and ISAM SIP ISDN Termination as supported templates for some 7302/7330/7356 /7360/7362/7363/7367 ISAM NEs	01	Table 34
Example of a new ONT notification event	Updated the example of an alarm for new ONTs	01	Section 9.2
Example of a notification for an NE reparenting export operation	Updated the example of a notification for an NE reparenting export operation	01	Section 10.2
Example of a notification for an NE reparenting import operation	Updated the example of a notification for an NE reparenting import operation	01	Section 10.2
Example of auditPort operation response	Added the missing error code to the example of the auditPort operation response for a misaligned service template state and a misaligned service component state	01	Section 7.2.3.3
Example of getClients operation request and response	Updated the examples of a getClients operation request and response	01	Section 8.2
Example of getNotificationTopic operation request and response	Updated the examples of a getNotificationTopic operation request and response	01	Section 8.2
Examples of change notification messages	Updated the examples of change notification messages for the configure operation, clean operation, and heartbeat message	01	Section 8.1
Export services	Moved information about export services for the service inventory, including output file information and examples, to the <i>5529 APC Installation, Administration, and User Guide</i>	01	<i>5529 APC Installation, Administration, and User Guide</i>
get operation	Moved the following information from section 4.17 to Table 14 : The get operation is supported for 7302/7330/7356/7360/7362/7363/7367 ISAM R5.2 or later and 7342 ISAM FTU.	01	Table 14
getConfiguredServices operation	Updated the examples of a getConfiguredServices operation request and response	01	Sections 7.1.4 and 7.2.4
getConfiguredTemplates operation	Added the getConfiguredTemplates operation request and response parameters	01	Section 4.23

(3 of 4)

Feature/enhancement	Description	Edition	See
getLogicalPorts operation	Updated the example of a getLogicalPorts operation response	01	Section 7.2.5
	Added the getLogicalPorts operation request and response parameters	01	Section 4.24
getNodeAuditResult information	Added the instanceLabel parameter to the getNodeAuditResult information	01	Table 50
getNodeAuditResult operation	Updated the example of a getNodeAuditResult operation response	01	Section 7.2.6
getTemplateMetaData operation	Updated the examples of a getTemplateMetaData operation request and response	01	Sections 7.1.3 and 7.2.7
	Added the instanceLabelNamingRule parameter to the description of the getTemplateMetaData operation response	01	Table 58
HTTPS interface	Added information about the HTTPS interface and the required security certificate for the 5520 AMS server	01	Section 3.3
modify operation	The 5529 APC does not support modifying the value of a mandatory attribute to be -No-Value-.	01	Section 4.34
	Added a table that describes the modify operation request parameters	01	Table 65
MTOSI UTC format	Added information about the MTOSI UTC format used to display a date and time in an NBI operation	01	Section 6.2.3.1 Tables 74 and 75
NBI operations supported by NE type and release	Added support for the 7362 ISAM DF R5.5 or later	01	Table 14
NE release support	Removed support for NE releases that are forward compatible in 5529 APC R9.6.03	01	Tables 14 , 18 , 19 , 34 , and 35 Section 9.1
Output file	Moved information, including examples, about the output file used for service inventory to the <i>5529 APC Installation, Administration, and User Guide</i>	01	<i>5529 APC Installation, Administration, and User Guide</i>
ReapplyAllServices	Added note indicating conditions that may cause the ReapplyAllServices operation to report success for a non-existing port	01	Section 4.37
Reusing shared virtual arguments	Revised the information about reusing shared virtual arguments	01	Section 4.2.4.1
Service inventory	Moved information about export services and the output file to the <i>5529 APC Installation, Administration, and User Guide</i> so that Chapter 5 now only describes service inventory	01	Chapter 5
Supported version of Java	The supported version of Java is 1.8 as of 5529 APC R9.6.	01	Section 6.1.1
Target object ID address extension	The MDU VEIP port does not support target object ID address extension.	01	Table 9
Template migration operations	Updated the examples for template migration operations. Added an example of an execute operation response for a template migration.	01	Section 7.1.2

(4 of 4)

3 5529 APC Northbound interface overview

3.1 5529 APC overview

3.2 Web service technology

3.3 Northbound interface security

3.4 Collaborative service provisioning

3.5 5529 APC in high-performance networks

3.6 5529 APC and high subscriber density configurations

3.7 Developer support

3.1 5529 APC overview

The 5529 APC MSAI provides a synchronous interface and asynchronous interface to automate service provisioning and activation functions for Nokia access NEs. The Northbound interface is the same for the supported NEs; see the *5529 APC Template Reference* for a list of the supported NEs and releases.



Note — The client of the synchronous interface is responsible for sequencing consecutive requests to a port.

3.1.1 Web services

The 5529 APC provides web services that allow one or more client applications to perform service provisioning and service activation functions on the NEs. Web services allow web service clients to access the functionality of the web service provider application using standard Internet protocols, such as HTTP/S and XML/SOAP.



Note — In Release 9.6.05 or later, the 5529 APC Northbound interface does not take into account user-assigned PAP groups and does not check the PAP rights, so a user that is assigned a PAP group with only a subset of NEs can still perform 5529 APC NBI operations on an NE that is not listed in the PAP group.

The Internet protocol specified in the header of NBI requests must match the Internet protocol specified for the 5520 AMS server or servers in order for the NBI request to be handled by the 5520 AMS server. If your 5529 APC is part of a cluster deployment model, make sure that all the 5520 AMS servers in the cluster specify the same Internet protocol (for example, all are HTTPS, which is the default Internet protocol for 5520 AMS servers) and that this is the Internet protocol specified in the NBI request header. By having the same Internet protocol as all the 5520 AMS servers, NBI requests are able to be handled by any of the 5520 AMS servers in the cluster. See the *5529 APC Installation, Administration, and User Guide* for more information about deployment models for the 5529 APC, including the cluster deployment model. See the *5520 AMS Administrator Guide* for information about managing HTTPS or HTTP communications between the 5520 AMS and 5529 Enhanced Applications.

In the Northbound interface, templates are used to provision ports and related services. The templates support the reduction of attributes and objects that are visible for the OSS client application; only the attributes and objects that are essential for the service are visible. A significant reduction of the underlying technology complexity can expedite integration and maintenance times.

A service planner configures the templates using the 5529 APC GUI before the templates can be used for service provisioning. Client applications can use template names for the services to be managed on the ports, in addition to the template arguments that are required by the services and OSS.

The client sends a request to a web service provider, such as the 5529 APC MSAI. The web service provider processes the request and sends the response to the client. Web service information, such as requests, responses, and faults, are converted to an XML stream. An XML stream is not associated with a software or hardware platform. Web services are described in the WSDL file or in the associated XML schema file. Web services are offered by software systems that are designed to support interoperable machine-to-machine interaction in a network. The interoperability is accomplished through a set of XML-based open standards, such as WSDL, SOAP, and UDDI. See chapter 4 for information about the 5529 APC web services.

3.1.2 Notification interface

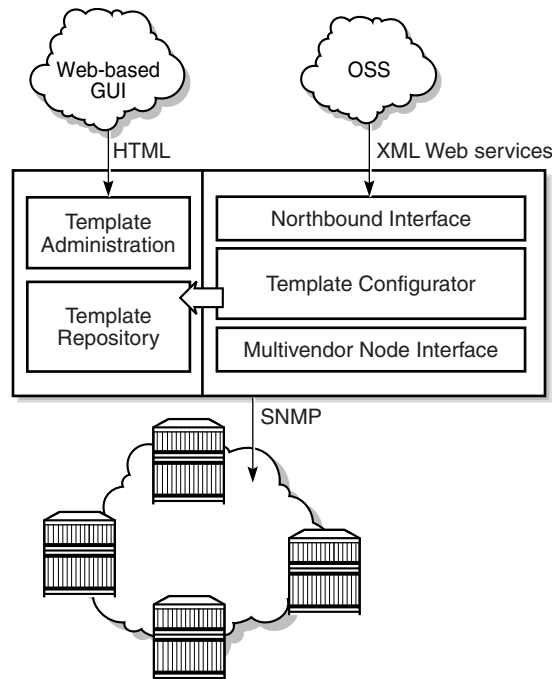
In some OSS environments, multiple OSSes are involved with the service fulfillment process or some OSSes have dependencies on what happens toward the NEs. In such cases, 5529 APC MSAI offers a notification interface that allows OSSes to collaborate. The notification interface generates configuration change notifications that relate to service provisioning and activation, regardless of whether the NE supports notifications and traps. Also, in some cases where OSS clients perform flow-through service provisioning using the 5529 APC MSAI and some 5520 AMS operators are using the 5529 APC SPFE to perform expert adjustments at the EMS layer, the notification interface is useful to update the service provisioning OSS with the changes made by the SPFE users. As a result, the service provisioning OSS does not need to poll for these changes.

However, the feature must not be misused to distribute service provisioning information among the completed OSSes. In some OSS environments, a lead OSS architect for service provisioning or inventory is appointed. Some OSSes, for example, a planning system, prefer to get their information from a consolidated inventory system instead of notifications from the 5529 APC. In general, the notifications are kept aligned with service provisioning and inventory systems. Unfortunately, clear rules on this issue do not exist; each OSS architect must decide which method is appropriate.

3.1.3 Architecture

Figure 1 shows a high-level overview of the 5529 APC architecture. The 5529 APC Template Configurator is the service provisioning engine. Using the Northbound interface data from the client application request, the Template Configurator accesses the Template Repository to retrieve information about the template and the sequence of required configurations for the NE.

Figure 1 5529 APC architecture



18517

3.2 Web service technology

The Northbound interface is based on web service technology. The Northbound interface works with Apache Axis2 version 1.7.1 software to support the following specifications:

- W3C SOAP 1.1/1.2
- W3C WSDL 1.1
- WS-I Basic Profile 1.0

See the following websites for additional information:

- <https://axis.apache.org/axis2/java/core/> (Apache Axis2)
- <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.ws-i.org/profiles/BasicProfile-1.0-2004-04-16.html> (WS-I Basic Profile 1.0)

See chapter 4 for more information about 5529 APC web services.

3.2.1 WSDL file

The Northbound interface and all of the web services are published using the associated WSDL file. After the 5529 APC is installed, the WSDL file is located in the following directory for all supported deployment models:

local_data_dir/local/repository/app-apc-release/nbi

where

local_data_dir is the local data directory you specified at installation (default is /var/opt/ams)

release is the 5529 APC software release

The WSDL file contains:

- operations that are available over the Northbound interface for each 5529 APC function
- arguments and return types for each operation
- binding information that describes how the service is implemented. For the Northbound interface, SOAP is used as the messaging protocol and document/literal is used as the data encoding style.

Use one of the following URLs to invoke the services:

- <https://host:8443/soap/services/ApcRemotePort/9.6>
- <http://host:8080/soap/services/ApcRemotePort/9.6>

where *host* is the host name or IP address of the 5520 AMS server on which the 5529 APC is installed



Note 1 — The HTTPS interface is the default interface. The HTTP interface is disabled by default. See section 3.3 for more information about the HTTPS interface. See the *5520 AMS Administrator Guide* for information about how to enable HTTP in the 5520 AMS.

Note 2 — The following URLs are supported for backward compatibility with earlier releases, but may not support the features of the current release:

- <https://host:8443/soap/services/ApcRemotePort/9.4>
- <http://host:8080/soap/services/ApcRemotePort/9.4>
- <https://host:8443/soap/services/ApcRemotePort/9.5>
- <http://host:8080/soap/services/ApcRemotePort/9.5>

where *host* is the host name or IP address of the 5520 AMS server on which the 5529 APC is installed

Note 3 — The 5520 AMS implements a rate-limiting login mechanism based on each incoming IP address. See the *5520 AMS User Guide* for more information.

The WSDL file is usually used to generate the client stub code. There are converters from the WSDL to different programming languages.

Table 6 lists the WSDL source files.

Table 6 WSDL source files

Web services operations	WSDL file
All 5529 APC web services operations, except for <code>getClients</code> and <code>getNotificationTopic</code> ; see Table 13	<code>apc-northbound.wsdl</code>
<code>getClients</code> <code>getNotificationTopic</code>	<code>apc-notification-northbound.wsdl</code>

See section 3.3, section 3.4, and chapter 9 for information about other service access points provided in the 5529 APC WSDL file.

For more information about the WSDL, see:

<http://www.w3c.org/TR/wsdl>

3.2.2 Verifying the 5529 APC service access point

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

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

This message indicates that the service access point is functional.

3.3 Northbound interface security

The Northbound interface uses the HTTP Basic Authentication mechanism (RFC 2617). Web service clients must add a username and password to the Authorization Header of the HTTP/S message in the web service requests. The username and password must match the username and password for the user account with an APC NBI role, as configured on the 5520 AMS GUI by the 5520 AMS system administrator.



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

The HTTPS interface 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.

3.3.1 Changes to HTTP login error responses

The HTTP response error messages that are returned when the NBI user login validation fails changed in Release 9.6.05.

Table 7 lists the differences in HTTP error messages between Release 9.6.03 and Release 9.6.05 or later for the NBI user login events.

Table 7 Changes to HTTP login error responses

Login event	HTTP response status code	HTTP error message (R9.6.03 or earlier)	HTTP error message (R9.6.05 or later)
5529 APC is not accessible	HTTP/1.1 401 Unauthorized	—	Unauthorized - Invalid credentials, user locked or authentication server not reachable

(1 of 2)

Login event	HTTP response status code	HTTP error message (R9.6.03 or earlier)	HTTP error message (R9.6.05 or later)
Address filter	HTTP/1.1 403 Forbidden	—	Forbidden - IP address is not allowed
Authentication failed	HTTP/1.1 401 Unauthorized	Unauthorized	Unauthorized - Invalid credentials, user locked or authentication server not reachable
License problem	HTTP/1.1 403 Forbidden	Forbidden - License problem	Forbidden - License problem
Maximum number of sessions exceeded	HTTP/1.1 403 Forbidden	—	Forbidden - Max number of sessions exceeded
Password expired	HTTP/1.1 403 Forbidden	Forbidden - Your current password has expired! Your access will be denied until the password is updated	Forbidden - Your current password is expired! Log in to the AMS GUI and choose a new password, then try again.
User locked	HTTP/1.1 401 Unauthorized	Unauthorized - Access to the requested resource has been denied - Your account is locked. Contact your administrator	Unauthorized - Invalid credentials, user locked or authentication server not reachable

(2 of 2)

3.3.2 MSAI services

When HTTPS is enabled, use one of the following URLs to invoke the MSAI services:

- <https://host:8443/soap/services/ApcRemotePort/9.4>
- <https://host:8443/soap/services/ApcRemotePort/9.5>
- <https://host:8443/soap/services/ApcRemotePort/9.6>

where *host* is the host name or IP address of the 5520 AMS server on which the 5529 APC is installed. The default TCP port is 8443.

3.3.3 Services supporting the notification interface

When HTTPS is enabled, the URL to invoke the services that support the notification interface is:

<https://host:8443/soap/services/ApcNotificationRemotePort/3.1>

where *host* is the host name or IP address of the 5520 AMS server on which the 5529 APC is installed. The default TCP port is 8443.

3.4 Collaborative service provisioning

The 5529 APC also provides a notification interface to allow OSSes to track configuration change notifications from OSS clients and SPFE users who are doing service provisioning using the 5529 APC. The notification messages describe the service provisioning activities, optionally including retrieval operations, that the clients or SPFE users are performing. See Table 13 for a list of the service provisioning operations and retrieval operations. The 5529 APC provides web services that allow a client application to retrieve a list of the 5529 APC clients from which that client application is authorized to receive notifications, and retrieve the notification topic to which a user can subscribe to receive configuration change notifications.



Note — In Release 9.6.05 or later, the 5529 APC NBI notification interface does not take into account user-assigned PAP groups and does not check the PAP rights, so a 5529 APC client that is assigned a PAP group with only a subset of NEs can still receive notifications from an NE that is not listed in the PAP group.

The notification interface is provided to separate the configuration change notification activities from the MSAI and to avoid confusion with the service provisioning operations. The notification interface is supported in the following 5529 APC deployments with 5520 AMS:

- standalone
- cluster

A separate WSDL file is provided for this notification interface. After the 5529 APC is installed, the WSDL file is located in the following directory for all supported deployment models:

`local_data_dir/local/repository/app-apc-release/nbi`

where

`local_data_dir` is the local data directory you specified at installation (default is `/var/opt/ams`)

`release` is the 5529 APC software release

See the *5529 APC Installation, Administration, and User Guide* for more information about the different 5529 APC deployment models.

The URL to invoke the services that support the notification interface is one of the following:

- `https://host:8443/soap/services/ApcNotificationRemotePort/3.1`
- `http://host:8080/soap/services/ApcNotificationRemotePort/3.1`

where *host* is the host name or IP address of the 5520 AMS server on which the 5529 APC is installed



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

Contact your 5529 APC application administrator to set up notification groups and assign clients to the notification groups to be able to receive notifications.

3.5 5529 APC in high-performance networks

A single installation of the 5529 APC is capable of communicating with the OSS using multiple NBI threads. To achieve the best performance with the 5529 APC using synchronous operations, you need to plan to use more than one NBI thread. There are no guidelines on the number of threads to use, but 10 to 40 NBI threads can meet most requirements in high-performance networks. For more information, contact your Nokia technical support representative.

3.6 5529 APC and high subscriber density configurations

Some NE types, when installed with high-capacity Ethernet LT cards, can support more subscriber services for an LT port, compared to other NE types supported by the 5529 APC where an LT port typically represents a subscriber with a limited number of services. The high-capacity Ethernet LT cards can have up to 2000 subscribers for each LT port that supports the provisioning of subscribers by lower-level objects in the object hierarchy, such as a VLAN association. This creates a high subscriber density configuration in which the number of services for each LT port can increase significantly even though there is no corresponding significant increase in the total number of subscriber services provided by the NE.

As part of its support for NE types with high-capacity Ethernet LT cards, the 5529 APC supports NE configurations that have a high density of subscribers, up to the maximum supported by the LT ports, where the NE supports up to 1000 VLAN associations for each LT port that represents a subscriber, with the following considerations:

- The following operations for the 5529 APC web services are not supported in a high subscriber density configuration:
 - auditPort (see section [4.8](#))
 - clean (see section [4.10](#))
 - dbClean (see section [4.12](#))
 - discoverAllServices (see section [4.14](#))
 - migrate—not supported at the shared port level but is supported for individual subscriber services (see section [4.33](#))
 - move (see section [4.36](#))
 - reapplyAllServices (see section [4.37](#))
- See the *5529 Enhanced Applications Release Notice* for information about operational performance issues for the 5529 APC caused by high subscriber density configurations

3.7 Developer support

Nokia provides developer support on a contract basis to help you develop and troubleshoot 5529 APC-developed applications. Contact your Nokia account or technical support representative for more information.

Independent software vendors and OSS vendors can contact the Nokia Connected+ Partner Program manager for the Access Division.

5529 APC Northbound interface

[4 Service fulfillment web services](#)

[5 Service inventory](#)

[6 Collaborative service provisioning](#)

4 Service fulfillment web services

[4.1 Service fulfillment web services overview](#)

[4.2 Web service parameters](#)

[4.3 Managing errors](#)

[4.4 Web services and operations](#)

[4.5 NBI operations supported by NE type and release](#)

[4.6 audit operation](#)

[4.7 auditNode operation](#)

[4.8 auditPort operation](#)

[4.9 check operation](#)

[4.10 clean operation](#)

[4.11 configure operation](#)

[4.12 dbClean operation](#)

[4.13 dbUnconfigure operation](#)

[4.14 discoverAllServices operation](#)

[4.15 execute operation](#)

[4.16 File-based operations](#)

[4.17 get operation](#)

[4.18 getApplicableTemplates operation](#)

[4.19 getChangeHistory operation](#)

[4.20 getConfiguredServices operation](#)

[4.21 getConfiguredTemplate operation](#)

[4.22 getConfiguredTemplateName operation](#)

[4.23 getConfiguredTemplates operation](#)

[4.24 getLogicalPorts operation](#)

[4.25 getNodeAuditResult operation](#)

[4.26 getResult operation](#)

[4.27 getResults operation](#)

[4.28 getSystemHealthInfo operation](#)

[4.29 getTemplate operation](#)

[4.30 getTemplateMetaData operation](#)

[4.31 getTemplateNames operation](#)

[4.32 importTemplate operation](#)

[4.33 migrate operation](#)

[4.34 modify operation](#)

[4.35 modifyTemplateInstance operation](#)

[4.36 move operation](#)

[4.37 reapplyAllServices operation](#)

[4.38 reapplyService operation](#)

[4.39 resume operation](#)

[4.40 suspend operation](#)

[4.41 unconfigure operation](#)

4.1 Service fulfillment web services overview

The 5529 APC displays the functions of the Northbound interface as web services. The web services are grouped in categories, as described in Table 8.

Table 8 **Web service categories**

Category	Description	See
Service provisioning	Allows the client application to manage services provided by NEs	Section 4.4.2
Bulk execution	Allows the client application to simultaneously start multiple commands. The commands are executed in the background. Web services can be used to monitor the execution status of the sequential commands.	Section 4.4.3
Service inventory	Allows the client application to get information about the network that is being managed, 5529 APC templates that are created to manage the network, and the templates associated with a port in the network	Section 4.4.4
Audit and reconciliation	Facilitates the alignment between the client application, the 5529 APC, and the network. Some web services are used at the initial introduction of the 5529 APC system in a network that is in operation, and other web services are used for routine verification of the alignment between systems.	Section 4.4.5
Service state verification	Allows the client application to retrieve the status of services that are applied to target objects	Section 4.4.6
Server availability verification	Allows the client application (and load balancers in a cluster deployment) to determine the availability of the 5529 APC NBI server	Section 4.4.7

Most web service operations use service provisioning templates. The service provisioning templates contain attributes, which can be parameters and arguments. The template parameters and arguments are defined during the service template design. See the *5529 APC Installation, Administration, and User Guide* for more information about attributes, parameters, and arguments.

Template arguments are visible and accessible to the OSS client applications.

See the *5529 APC Template Reference* for more information about templates. There is one guide for each NE plug-in type.

See section [4.2](#) for information about the web service parameters that are used by the Northbound interface. See section [4.4](#) for information about web services, operations, and associated parameters.

4.2 Web service parameters

The parameters for each web service may vary, although there are common parameters. The common parameters are:

- [objectName](#)
- [templateName](#)
- [templateVersion](#)
- [argument](#)
- [instanceLabel](#)
- [operationInitiator](#)
- [ResultIndicator](#)

4.2.1 objectName

The `objectName` parameter specifies the port or object that is addressed by the operation. See the *5529 APC Template Reference* for an NE for identification information for objects and port types. See section 4.2.1.1 for information about extending the target object ID in the `objectName` parameter when more than one object matches the target object ID.



Note 1 — If an NE is managed by the 5529 APC, the NE name must not contain a colon character (:); otherwise, the NBI will not work with that NE.

Note 2 — The 5529 APC does not support leading zeros in object IDs; for example, `NEName:1-1-3-03` is not supported, but `NEName: 1-1-3-3` is supported.

Note 3 — You should not perform NBI operations on a port in which the Customer ID attribute of the SHDSL span for that port has a -passive- value and the port is part of a multi-wire configuration. You also need to disregard the attribute values for the passive ports. In a multi-wire configuration, all of the relevant SHDSL pair objects and corresponding operational data are available under the SHDSL span of the first port, which is the only configured port; all other ports are passive.

4.2.1.1 Target object ID address extension

In some cases, there is more than one object that matches the target object specified in an NBI operation. When more than one object matches the target object, the following occurs:

- For configuration operations, the configuration attempt fails and returns a `TARGET_OBJECT_NOT_UNIQUE` error code.
- For inventory operations, the result contains all matching target objects.

To specify a target object, you can extend the target object ID to include an object type and optional NE path. You only need to include the NE path when the object type is not sufficient to identify the target object.

Address extension is supported for the objects listed in Table 9.

Table 9 Object types and paths for address extension

Object	Object type	NR path
MDU Ethernet port	EthernetPort	R-S-LT-PON-ONT-C-P
MDU POTS port	POTSPort	R-S-LT-PON-ONT-C-P
MDU VEIP port ⁽¹⁾	VEIPPort	R-S-LT-PON-ONT-C-P

Note

(1) The MDU VEIP port does not support target object ID address extension.

Address extension is supported for the following operations:

- auditPort
- clean

Use the following format to extend the target object ID:

`<objectName>APC_address/object_type/NE_path</objectName>`

where

- *APC_address* is the 5529 APC address; for example, GPON:1-1-1-1-1-1-1
- *object_type* is one of the object types listed in Table 9; the object type is case-sensitive
- *NE_path* is the optional NE path listed in Table 9

The following example shows a clean operation without address extension:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="uri://alcatel.com/apc/9.5">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:clean>
      <objectName>GPON:1-1-1-1-1-1-1</objectName>
    </ns:clean>
  </soapenv:Body>
</soapenv:Envelope>
```

The following example shows a clean operation that uses address extension, including the optional NE path:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="uri://alcatel.com/apc/9.5">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:clean>
      <objectName>GPON:1-1-1-1-1-1-1/EthernetPort/R-S-LT-PON-ONT-C-P</objectName>
    </ns:clean>
  </soapenv:Body>
</soapenv:Envelope>
```

4.2.1.2 Special cases when target object ID address may be unclear for NBI operations

Table 10 describes the conditions that will make the target object ID address unclear for the configure operation and other NBI operations. The table also provides a workaround to make sure the target object ID address is accurate for the NBI operations.

Table 10 Target object ID address special case

Case	Workaround
<p>The following conditions will make the target of the operation unclear:</p> <ul style="list-style-type: none"> A PON MDU ONT is configured with a combination port; for example, Ethernet and VEIP. The different port types in the combination port have the same number; for example, Ethernet 1 and VEIP 1. One of the following conditions exist: <ul style="list-style-type: none"> An operation without template context is run on the port; for example, an auditPort or clean operation. The configure operation is run using a multi-layer template stack at the bridge port level, which is located above the combination port. 	<p>You can avoid a target object ID address conflict by doing the following:</p> <ul style="list-style-type: none"> Avoid using combination card configurations with identical numbering for ports that are of different types. Avoid using multi-layer template stacks at the VLAN association level for this type of configuration. Use the optional target object ID address extension for operations without template context, such as the auditPort or clean operation. See section 4.2.1.1 for information about the target object ID address extension.



Note — Another case in which the target object ID address may be unclear for NBI operations is when a DSL LT and NGPON2 ONT are configured on the same NE and mapped with the same numerical address. In this case there is no workaround to make sure the target object ID address is accurate for the NBI operations, so Nokia recommends that you do not configure a DSL LT and NGPON2 ONT on the same NE.

4.2.2 templateName

The templateName parameter specifies the template for the operation. The parameter must match the template name that is set using the 5529 APC GUI. You can use the getConfiguredTemplateName operation to retrieve all of the configured template names. See section [4.22](#) for more information about the operation.

4.2.3 templateVersion

The templateVersion parameter specifies the version of the template for the operation. The templateVersion parameter field contains an integer and it is optional. If a version is not specified, the 5529 APC applies the most recent version and it also depends on the template state. For the configure operation, the latest released template is used. For all other operations, the latest template version that was previously applied is considered for the operation.

4.2.4 argument

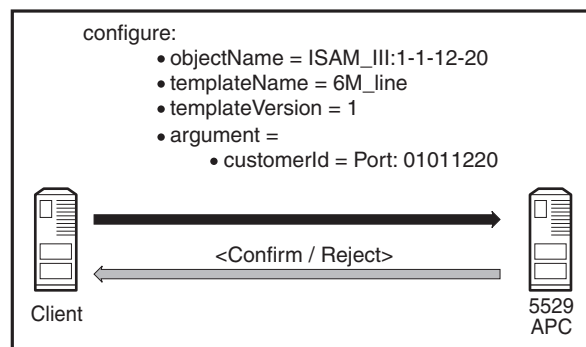
The argument parameter specifies the template arguments and associated values for the operations. The arguments must be specified in an unordered list that contains the *name/value* pairs, where *name* and *value* are strings. The list is optional and may not appear. If the list appears, the list must have at least one *name/value* pair.

At a minimum, the 5529 APC expects a template argument for which no default value is defined. If a template argument is not specified for the argument parameter in the web service, and the template argument does not have a default value, the 5529 APC will reject the operation.

For composite templates, the client application must specify the template arguments that are required for all of the basic templates that are part of the composite template. You can use the basic template arguments (the specified value is used to set all of the template arguments with the specified name in the individual templates) or use virtual arguments, if the virtual arguments were defined in the composite during creation. For the OSS client, each template (basic, composite, with or without virtual arguments) is displayed as a separate template with the template name and supported template arguments, which can be specified as a list.

Figure 2 shows an example of a configure request. See section 4.11 for information about the configure operation. See section 7.1.1 for examples of the configure operation request and response xml.

Figure 2 Configure request example



25365

You can specify whether an argument has an empty string value. The empty string value applies to string type arguments. The empty string value indicates that the string has zero length and a value is assigned to the argument. In the Northbound interface, empty strings can be provided by leaving the content empty for the value element. For example:

```

<templateName>template_name</templateName>

<argument>

  <name>argument_name</name>

```

```
<value></value>  
  
</argument>
```

If an argument is not supplied from the Northbound interface, the attribute is considered to not have a value assigned to it. In this case, the value set for the attribute during the operation depends on the default value that was specified for the attribute in the template at the template design stage.

4.2.4.1 Reusing shared virtual arguments

In the 5529 APC NBI, when you send a second configure operation request that has a shared argument that is also a virtual argument, make sure that you send the value of the virtual argument in the request. The NBI does not attempt to use the value that is already configured in the 5529 APC.

In the SPFE, for service templates in which the shared arguments prefill function is used, the argument value will be automatically populated from the 5529 APC and the issue that is previously described for the NBI will not occur.

4.2.5 instanceLabel



Note — Nokia does not recommend the use of multi-instance templates; use multi-instance templates only when necessary.

The instanceLabel applies to multi-instance templates. The instanceLabel parameter specifies a unique instance of a template within the context of a template name and a physical or logical port. The OSS client must reuse the same instanceLabel value in subsequent operations for the same template instance on the same port.

The instanceLabel parameter value is optional because you can choose whether to use multi-instance templates. When you decide to use multi-instance templates, you must use the instance labels consistently on the intended multi-instance templates. When only the template name is provided, then only the template name is used to identify the template instance. The OSS client can use multi-instance and non-multi-instance versions of a template in the same 5529 APC system and on the same port, however, the multi-instance and non-multi-instance templates must have different names.

The multi-instance template separates template identity, which is related to template management, from service identity (creation and activation), which is related to template attributes (parameters and arguments). The OSS client must combine the argument values that can be configured multiple times on a port.

A template designer can enforce a naming rule for the instance labels that can be used in combination with a template. A naming rule ensures that OSS users and SPFE users enter instance labels according to a defined character pattern. The template designer can change the rules; however, the template designer must ensure that a new naming rule is acceptable for OSS clients and SPFE users, and conforms with previously-used instance labels for the same template. Contact your template designer to define multi-instance labels and multi-instance naming rules.

Template metadata can indicate to the OSS clients whether the template is intended to be used as a multi-instance template. See section [4.30](#) for information about the `getTemplateMetaData` operation.

4.2.6 operationInitiator

The `operationInitiator` parameter specifies the originator of a Northbound interface operation. The `operationInitiator` parameter is optional. Nokia recommends that the parameter value be a combination of the system/application name and user ID at the originating system of the request, for example, `MyServProvOSSxyz/user1` or `MyCRMabc/user2`.

When a value is entered for the `operationInitiator` parameter in an operation, the previous value of the parameter is overwritten with the new value for the target object. Nokia recommends that you do not use only the client name and client username (name of the 5529 APC client) as the `operationInitiator` value, as user activities at OSS systems are typically mapped onto a 5529 APC client that interacts with the 5529 APC and the purpose of the `operationInitiator` parameter is to track users rather than systems. If originating user information is not available at the integration point with the 5529 APC, a temporary system name and client name may be used on an interim basis.

All OSS clients must use the parameter or not use it at all in order for it to be useful to track the last user who made a change to the port. The 5529 APC only keeps track of users who are identified using the `operationInitiator` parameter.

For information about how the `operationInitiator` parameter, called *Operation Initiator System Name*, is used in the SPFE GUI, see the *5529 APC Installation, Administration, and User Guide*.

4.2.7 ResultIndicator

The `ResultIndicator` parameter is an optional parameter in the operation response that explicitly states that the operation was successful.

By default, the following operations return an empty response if the operation is successful:

- clean
- configure
- dbClean
- dbUnconfigure
- discoverAllServices
- getSystemHealthInfo
- importTemplate
- migrate
- modify
- modifyTemplateInstance
- move
- reapplyAllServices
- reapplyService
- resume
- suspend
- unconfigure

To provide the ResultIndicator parameter in successful operation responses, a JMX console parameter (AddResultIndicator) is available to enable the feature, which is disabled by default. You require permission from Nokia Product Line Management to modify JMX console settings. The modifications need to be performed by Nokia personnel only. Contact your Nokia technical representative for more information.

The following is an example of a response for a successful configure operation that contains the ResultIndicator parameter:

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:configureResponse xmlns:ns="uri://alcatel.com/apc/9.6"/>
      <ResultIndicator>Success</ResultIndicator>
    </ns:configureResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

4.3 Managing errors

An error may occur during a web service operation. The errors are reported by the server to the OSS system through the Northbound interface by the use of exceptions. There are two types of exceptions:

- HTTP
- 5529 APC

4.3.1 HTTP exceptions

An HTTP exception may be returned when a request is sent to the server. This means that the request did not reach the 5529 APC components because it was intercepted in the lower stacks of the framework, for example, by the HTTP application server. The 5529 APC and NE network are not affected when the request does not reach the 5529 APC components.

Some of the common situations in which HTTP errors occur are:

- The URL that is used to indicate where the web service request should be delivered is incorrect.
- There is a security problem, for example, an incorrect or missing value in the 5529 APC username or password in the HTTP header, or the 5529 APC license has expired.
- The request does not contain the correct syntax that is defined in the 5529 APC WDSL file.

4.3.2 5529 APC exceptions

An error may occur during a web service operation. The 5529 APC performs the following functions if there is an error:

- performs a rollback, regardless of the type of template. A rollback for the clean operation is not attempted.
- returns a RemoteApcException message that describes the error to the client application

Table 11 describes the format of the RemoteApcException message.

Table 11 RemoteApcException message format

Element	Description	See
message	User-friendly description of the error, which may include information about whether the rollback was successful or not successful. The message is free-form text and only for display.	<i>5529 APC Installation, Administration, and User Guide</i> for information about how to change the error messages to different languages
errorCode	Specific error code	See Table 12
rollback	Displays when a rollback was attempted and provides the following information about the rollback completion: <ul style="list-style-type: none">• status, which indicates whether the rollback succeeded (SUCCESS) or failed (FAIL)• errorCode, which is displayed if the status is FAIL and indicates the rollback errorCode. See Table 12.• message, which is displayed if status is FAIL and indicates the message that is associated with the rollback error	—

Table 12 describes the errorCode in the RemoteApcException message.

Table 12 errorCode description

errorCode	Description	See
ACTION_NOT_DEFINED	Internal error	—

(1 of 4)

errorCode	Description	See
ATTRIBUTEVALUEGENERATION_ALGORITHM_MISSING	The required Attribute Value Generation plug-in is missing or not active.	<i>5529 APC Installation, Administration, and User Guide</i>
ATTRIBUTEVALUEGENERATION_FAILED	The plug-in that contains the attribute value generation formula encountered a problem while generating a value for an attribute	<i>5529 APC Installation, Administration, and User Guide</i>
COULD_NOT_READ_FILE	The file specified for executeURL operation request is available, but the 5529 APC cannot use the file because of the permissions	Section 4.16
DSLAM_NOT_FOUND	The NE specified in the request is not present in the 5520 AMS	—
DSLAM_NOT_IN_CONFIGURED_STATE	The NE specified in the request is not in a CONFIGURED state in the 5529 APC	<i>5529 APC Installation, Administration, and User Guide</i>
EMS_UNAVAILABLE	The connection from the 5529 APC to the EMS (5520 AMS) is not available	—
EXCPT_UNABLE_TO_COMPLY	The NE specified in the request is not in the supervision state	—
FAILED_TO_ACQUIRE_TRANSACTION	The 5529 APC cannot get ownership of a transaction session for the NE	—
FILE_NOT_FOUND	The file specified for the executeUrl operation is not available	Section 4.16
GENERIC	Error conditions that cannot be categorized by other errorCodes	—
HIGHER_LAYER_CONFIGURED	The requested template for unconfiguration is associated with a higher layer template that should be unconfigured first	—
ILLEGAL_TEMPLATE_STATE	The template name and version specified in the request cannot be used because of the associated state, for example, draft	—
INVALID_ARGUMENT	An unexpected argument was provided for the request	Section 4.2.4
INVALID_FILE_CONTENT	The file specified in the executeUrl operation is available and can be used by the 5529 APC, but the file contents are invalid, according to the XML syntax	Section 4.16
INVALID_PORT_ADDRESS	The port address that is specified in the request does not contain the correct syntax	—
INVALID_TEMPLATE_VERSION	The template version for the request does not exist	Section 4.2.3
INVALID_URL	The URL specified for the executeUrl operation is not valid	Section 4.16
LICENSE_PROBLEM	There is a problem with the 5529 APC license	<i>5529 APC Installation, Administration, and User Guide</i>
LOWER_LAYER_NOT_CONFIGURED	The template for a configure request requires a lower layer template to be configured first	—
MISSING_AMS_LINK_BETWEEN_G6_AND_GPON	The 7342 ISAM FTTU or 7360 ISAM FX NE specified in the request is not associated with a GENBAND G6 NE	—
MISSING_AMS_LINK_BETWEEN_MDU_AND_GPON	The 7367 ISAM SX/DX NE that is specified in the request is not associated with a 7342 ISAM FTTU or 7360 ISAM FX NE	—

(2 of 4)

errorCode	Description	See
MISSING_ARGUMENT	A mandatory template attribute that was defined as an argument without a default value is not present in the request	Section 4.2.4
NO_MATCHING_TEMPLATE	None of the specified candidate service templates matches network objects	Section 4.14
NON_MODIFIABLE_ARGUMENT	The argument cannot be changed using the modify operation	Section 4.34
NON_MODIFIABLE_ATTRIBUTE	All misaligned attributes are non modifiable, and the reapplyStrategy is set to MODIFY_ONLY	Sections 4.37 and 4.38
PARTIAL_SUCCESS	The operation was only partially successful. For example, if a reapplyAllServices or reapplyService operation performed with the MODIFY_ONLY strategy finds misalignments for both modifiable and non-modifiable attributes, then the 5529 APC attempts to fix the misalignments by fixing only the modifiable attributes. This action does not fix the misalignment completely, and the 5529 APC returns a PARTIAL_SUCCESS error message.	Sections 4.37 and 4.38
PORT_NOT_CLEAN	A port that is not configured according to the 5529 APC, but the port is configured on the NE	Sections 4.7 and 4.8
PORT_UNKNOWN	The port that is specified for the objectName parameter does not exist in the NE or the port is unknown by the 5529 APC	—
REQUEST_ID_IN_USE	A command is displayed twice in a bulk request	—
TARGET_OBJECT_MISSING_IN_NE	There are services configured in the 5529 APC for this port, but no service is configured on this NE port.	Section 4.7
TARGET_OBJECT_NOT_UNIQUE	There is more than one object that matches the target object specified in the operation.	Section 4.2.1.1
TEMPLATE_ALREADY_CONFIGURED	A template is being applied to the port and cannot be reapplied	—
TEMPLATE_MULTIPLICITY_VIOLATION	The specified template cannot be applied to the port more than once.	—
TEMPLATE_NOT_CONFIGURED	A template is used in an operation for a port, but the template is not configured for the port	—
TEMPLATE_NOT_SUPPORTED_ON_NODE	The template that is used in the operation cannot be applied to the port in the NE	—
TEMPLATE_NOT_SUPPORTED_ON_PORT	The template that is specified in the request is not supported by the port for the configure operation	—
TEMPLATE_REFERENCED_BY_COMPOSITE	A template cannot be unconfigured from the port because the template is part of a composite template on the port	—
TEMPLATE_REFERENCED_BY_OTHER_ADDRESS	A clean, dbClean, unconfigure, or dbUconfigure operation cannot be successfully completed when some of the templates are referenced by other ports. For example an IMA group cannot be cleaned if there are ports that reference the IMA group.	Sections 4.10 , 4.12 , 4.13 , and 4.41
TEMPLATE_UNKNOWN	The template does not exist or is unknown by the 5529 APC	—

(3 of 4)

errorCode	Description	See
TIMEOUT_ERROR	<p>The request timed out.</p> <p>If the associated error message is "Node unreachable, timeout connecting to: X.X.X.X", then SNMP has timed out after a configurable number of retries.</p> <p>If the associated error message is "Unable to get free DB connection within configured time period", then the OSS client application must hold off service provisioning for a period of time. It is advised that the number of parallel requests to the 5529 APC be configurable in the OSS client application.</p> <p>If the associated error message is "Command not executed due to max retries reached during bulk execution", then the command did not execute and there is no outstanding automatic retry pending.</p>	—
UNABLE_TO_SELECT_OR_PATH	The 5529 APC is unable to find the correct OR template stack for the operation	—
UNSUCCESSFUL_OPERATION	The operation is unsuccessful due to an error from the NE; for example, when a template cannot be unconfigured because of external dependencies	Sections 4.37 and 4.38
UNSUPPORTED_OPERATION	The requested operation is not supported for the object	—
UNSUPPORTED_STATUS_EXPRESSION	The check operation provided a valid argument and an unsupported status expression	Section 4.9

(4 of 4)

4.3.3 Profile name error message

An error message is reported by the 5529 APC NBI when a profile name that is used in a 5529 APC template is missing from the NE where the template was applied. The error message provides information about the profile name to the user.

Example of the error message:

Caused by: The value '*profile_name_attribute_value*' for attribute *profile_name_attribute_NBI_name* could not be found on '*NE_name*'.

where

- *profile_name_attribute_value* is the value (name of the profile) for the profile name attribute
- *profile_name_attribute_NBI_name* is the NBI name of the profile name attribute
- *NE_name* is the name of the NE where the template was applied

4.4 Web services and operations

This section summarizes the 5529 APC web services and operations.



Note — The operations for some 5529 APC web services may be affected in high subscriber density configurations; see section [3.6](#) for more information.

4.4.1 Web service inventory

Table [13](#) lists the operations for each 5529 APC web service category. The table also has a reference to the section that describes each operation.



Note 1 — In this document, the tables of NBI operation parameters may not list the parameters in the same order as they are listed in the actual SOAP XML requests and responses in the OSS. See the WSDL files for the correct order of the parameters.

Note 2 — Some operations in Table [13](#) are not supported when the 5529 APC is used in a high subscriber density configuration; see section [3.6](#) for information about high subscriber density configurations and for operations that are not supported in high subscriber density configurations.

Table 13 Web service inventory

Operation	For more information about the operation, see
Service provisioning	
configure	Section 4.11
unconfigure	Section 4.41
suspend	Section 4.40
resume	Section 4.39
migrate	Section 4.33
modify	Section 4.34
modifyTemplateInstance	Section 4.35
move	Section 4.36
Bulk execution	
execute	Section 4.15
executeUrl	Section 4.16
getResult	Section 4.26
getResults	Section 4.27

(1 of 2)

Operation	For more information about the operation, see
getResultsUrl	Section 4.16
getResultUrl	Section 4.16
Service inventory	
getApplicableTemplates	Section 4.18
getConfiguredServices	Section 4.20
getConfiguredTemplate	Section 4.21
getConfiguredTemplateNames	Section 4.22
getConfiguredTemplates	Section 4.23
getLogicalPorts	Section 4.24
getTemplate	Section 4.29
getTemplateMetaData	Section 4.30
getTemplateNames	Section 4.31
Audit and reconciliation	
audit	Section 4.6
auditNode	Section 4.7
auditPort	Section 4.8
clean	Section 4.10
dbClean	Section 4.12
discoverAllServices	Section 4.14
getNodeAuditResult	Section 4.25
importTemplate	Section 4.32
dbUnconfigure	Section 4.13
reapplyAllServices	Section 4.37
reapplyService	Section 4.38
Service state verification	
check	Section 4.9
get	Section 4.17
getChangeHistory	Section 4.19
Server availability verification	
getSystemHealthInfo	Section 4.28

(2 of 2)

4.4.2 Service provisioning

Service provisioning operations allow the client to execute the life cycle of the port, including service provisioning and activation.

See Table 13 for a list of the service provisioning operations.

4.4.3 Bulk execution

Bulk executions allow the client application using the 5529 APC to start multiple simultaneous commands that are executed in the background when resources on the 5529 APC are available. There are operations to request the execution status of the sequential commands. Bulk execution allows asynchronous communications between the OSS client and the 5529 APC because commands are requested by the client application without waiting for the responses.

When the 5529 APC receives a bulk execution request, or multiple bulk execution requests, from the northbound interface, the order in which the operations are executed is based on the order in which the requests are received. The order of operations to a single port or ONT in the SOAP XML message of the execute operation request is preserved during execution. The operations are received and stored in the 5529 APC and are then executed sequentially. The command ID does not affect the order of execution of the commands.

The client application and the 5529 APC exchange commands to be executed. The information about the execution can be exchanged by using web service requests and responses or files in the XML format. A combination of both methods is valid; for example, commands sent as a web service request can have the 5529 APC respond in the XML file format.

Some operations can be retried to increase the operation success rate, which means that those operations remain in the planned state for a longer period of time, until they either succeed or the final retry fails.

The execute operation (described in section 4.15) returns a requestId from the 5529 APC that allows the client application to request the status of the commands that have not been performed using the getResult and getResults operations (described in sections 4.26 and 4.27). You need to keep running the getResult and getResults operations until the state parameter in the response message displays a READY value, which indicates that the execute operation was performed.

See chapter 7 for an example of an imported bulk request file for a template migration operation.

See Table 13 for a list of the bulk execution operations.

4.4.4 Service inventory

The service inventory web services allow the client application to retrieve information about the network and the templates configured on the ports in the network from the 5529 APC.

See Table 13 for a list of the service inventory operations.

4.4.5 Audit and reconciliation

The audit operations are used to uncover discrepancies between the intended configuration (as retained by the 5529 APC on behalf of the OSS and SPFE users), and the actual configuration of the port or ONT/service in the NE. The reconciliation operations enable the alignment of the port or ONT/service. The reapply operations impose the intended configuration on the port or ONT in the NE, while the discover operation learns the actual configuration from the port or ONT by matching it to selected service templates.

Typically, service providers need subscriber contracts to be enforced on the subscriber ports in the network; service providers need the port configurations to be aligned with the BSS view. The 5529 APC allows the verification of the OSS/BSS view that is used by the 5529 APC at configuration time with the state of the network; the verification operation is called an audit.

An audit provides a method to detect incorrect configurations in the network. If an incorrect configuration is detected, there are operations that are provided to return the object to the correct or desired configuration. The client application must also be able to check the alignment between the client application and the 5529 APC; therefore, you can query the status of the 5529 APC.

The 5529 APC supports the following audit operations:

- audit
- auditPort
- auditNode
- getNodeAuditResult

The 5529 APC supports the following reconciliation operations:

- reapplyService
- reapplyAllServices
- discoverAllServices
- clean
- dbClean
- importTemplate
- dbUnconfigure



Note — When the Subscriber Location ID Mode parameter in the GPON system parameter is set to the hexadecimal format, sometimes the values of the ONT subscriber location ID in the 5529 APC and 5520 AMS will not align (for example, when the value specified in the 5529 APC is less than 20 characters), even though the audit operation does not report a discrepancy.

For information about the GPON system settings, see the 5520 AMS Operations and Maintenance guide for the NE.

See Table 13 for a list of the audit and reconciliation operations.

4.4.6 Service state verification

The service state verification operations allow the client application to retrieve the status of services applied to target objects.

See Table 13 for a list of the service state verification operations.

4.4.7 Server availability verification

The server availability verification operation allows the client application (and load balancers in a cluster deployment) to determine the availability of the 5529 APC NBI server.

Table 13 lists the server availability verification operation.

4.5 NBI operations supported by NE type and release

Table 14 summarizes the supported 5529 APC NBI operations, listed by NE type and release.



Note — Support for the getSystemHealthInfo operation is independent of NE types and releases.

Table 14 NBI operations supported by NE type and release

NE type	NE releases	Operations supported in MSAI
7302 ISAM/7330 ISAM FTTN	R3.7	All template name retrieval operations, all template retrieval operations, audit, auditNode, auditPort, bulk interface, check, clean, configure, dbClean, dbUnconfigure, getLogicalPorts, importTemplate, migrate, modify, reapplyAllServices, reapplyService, resume, suspend, unconfigure
7302 ISAM, 7330 ISAM FTTN, 7356 ISAM FTTB, 7360 ISAM FX, 7362 ISAM DF/SF, 7363 ISAM MX, 7367 ISAM SX/DX ^{(1) (2) (3) (4)}	R5.2 or later	All template name retrieval operations, all template retrieval operations, audit, auditNode, auditPort, bulk interface, check, clean, configure, dbClean, dbUnconfigure, discoverAllServices, get, getLogicalPorts, importTemplate, migrate, modify, reapplyAllServices, reapplyService, resume, suspend, unconfigure
7330 ISAM FTTN	FGN4.3 or later	All template name retrieval operations, all template retrieval operations, audit, auditNode, auditPort, bulk interface, check, clean, configure, dbClean, dbUnconfigure, discoverAllServices, getLogicalPorts, importTemplate, migrate, modify, reapplyAllServices, reapplyService, resume, suspend, unconfigure
7342 ISAM FTTU ⁽⁴⁾	R4.8 or later	All template name retrieval operations, all template retrieval operations, audit, auditNode, auditPort, bulk interface, check, clean, configure, dbClean, dbUnconfigure, discoverAllServices, get, getLogicalPorts, importTemplate, migrate, modify, reapplyAllServices, reapplyService, resume, suspend, unconfigure

(1 of 2)

NE type	NE releases	Operations supported in MSAI
7353 ISAM FTTB/CX	R2.6.0 or later	All template name retrieval operations, all template retrieval operations, audit, auditNode, auditPort, bulk interface, check, clean, configure, dbClean, dbUnconfigure, discoverAllServices, getLogicalPorts, importTemplate, migrate, modify, move, reapplyAllServices, reapplyService, resume, suspend, unconfigure
GENBAND G6	R12.3 or later	All template name retrieval operations, all template retrieval operations, audit, auditNode, auditPort, bulk interface, check, clean, configure, dbClean, dbUnconfigure, discoverAllServices, getLogicalPorts, importTemplate, migrate, modify, reapplyAllServices, reapplyService, resume, suspend, unconfigure

(2 of 2)

Notes

- (1) The EPON card is supported on 7360 ISAM FX (ISAM R5.2 or later).
- (2) The 5520 AMS supports the 7362 ISAM DF R5.5 or later and 7362 ISAM DF/SF R5.7 or later.
- (3) The 5520 AMS supports the 7367 ISAM SX R5.3 or later and 7367 ISAM SX/DX R5.7 or later.
- (4) The get operation is supported for 7302/7330/7356/7360/7362/7363/7367 ISAM R5.2 or later and 7342 ISAM FTTU.

4.6 audit operation

The audit operation is an audit and reconciliation operation that determines whether the operation parameters that are specified by the client application in the audit request are configured on the NE for the specified port. This operation can be used at any level to audit only the service against which the audit is called. A successful operation requires a template to be specified in the request operation.

For ports that are not configured using the 5529 APC, the audit returns:

- a True value if the port is configured with the template parameters using another manager application
- a False value if the port is not configured with the template parameters using another manager application

The 5529 APC allows you to audit optional attributes that have the -No-Value- special token as the default value. When -No-Value- is the attribute default value, the 5529 APC uses the NE default value as the expected attribute value. Some optional attributes do not have an NE default value defined and are omitted from the audit when the attribute value is set to -No-Value-. Audit is still supported when the attribute value is set to a value in the range. See the *5529 APC Template Reference* for an NE for information about attributes that do not have an NE default value defined.



Note — The functionality that allows the 5529 APC to use the NE default value as the expected attribute value when -No-Value- is the default value for an optional attribute is not supported on 7302 ISAM/7330 ISAM FTTN R3.7/FGN3.8.

Table 15 describes the audit operation request parameters.

Table 15 audit operation request parameters

Parameter	Description	See
objectName	The name of the port for the operation or the name of the object created on top of the port	<i>5529 APC Template Reference</i> for identification information for objects and port types
templateName	The name of the template	Section 4.2.2
templateVersion	The version of the template	Section 4.2.3
instanceLabel	Only applicable for multi-instance templates. Not applicable for regular templates. Nokia recommends the use of regular templates. To distinguish multi-instance templates from regular templates, see section 4.30.	Section 4.2.5
argument	Container element for the argument name and value parameters	Section 4.2.4
name	The name of the argument	—
value	The value of the argument	—

Table 16 describes the audit operation response parameters when the port has not been configured using the 5529 APC. See chapter 7 for examples of successful and failed audit operation response messages.

Table 16 audit operation response parameters

Parameter	Description	See
auditResponse	Container element for the audit operation response message	—
auditReport	Container element for the audit operation response parameters	—
templateName	The name of the template for the audit	—
instanceLabel	Only applicable for multi-instance templates. Not applicable for regular templates. Nokia recommends the use of regular templates. To distinguish multi-instance templates from regular templates, see section 4.30.	Section 4.2.5
templateType	The type of template for the audit	—
configured	Indicates whether the object is configured as desired (True) or not (False) on the 5529 APC. The desired configuration is identified by the operation input parameters.	—
remoteFailedAssertion	Indicates each difference between the desired configuration of the actual configuration for the object in the NE. A remoteFailedAssertion item is displayed for each difference. Each item indicates the expected and actual parameter values.	—

4.7 auditNode operation

The auditNode operation is an audit and reconciliation operation that determines whether the configuration information in the 5529 APC for all of the ports of an NE match the configuration on the NE.



Note 1 — The 7301 ASAM GELT-A card is not supported for the auditNode operation.

Note 2 — If the template attribute is not present or is not supported on a board, the AuditNode response can be misaligned.

Table 17 describes the parameters for the auditNode operation request.

Table 17 auditNode operation request parameters

Parameter	Description
objectName	Indicates which NE to audit.
inExcess	Indicates whether the operation should audit the NE for a configuration in excess of the configuration set by the templates that are being audited. If the parameter is set to true, all of the ports in the NE are audited for in-excess configuration. This parameter is optional.

The auditNode operation returns a requestId from the 5529 APC, which allows the client application to request the status of the auditNode execution using the getNodeAuditResult operation. The auditNode operation is immediately executed, indicating that the request is accepted successfully. However, the audit for the NE is performed when 5529 APC system resources are available, using an asynchronous method.

The auditNode operation result for each port is the same as the auditPort operation result. See Table 19 for descriptions of the auditPort response messages.

See section 4.25 for information about how to request the status of a previously executed auditNode operation request.

The 5529 APC has an application setting (Collection Strategy for Node Audit) to specify the data collection strategy for NEs managed by the 5520 AMS communication layer. See the *5529 APC Installation, Administration, and User Guide* for information about the data collection strategies and configuring the Collection Strategy for Node Audit setting.

The 5529 APC allows you to audit optional attributes that have the -No-Value- special token as the default value. When -No-Value- is the attribute default value, the 5529 APC uses the NE default value as the expected attribute value. Some optional attributes do not have an NE default value defined and are omitted from the audit when the attribute value is set to -No-Value-. Audit is still supported when the attribute value is set to a value in the range. See the *5529 APC Template Reference* for an NE for information about attributes that do not have an NE default value defined.



Note 1 — The functionality that allows the 5529 APC to use the NE default value as the expected attribute value when -No-Value- is the default value for an optional attribute is not supported on 7302 ISAM/7330 ISAM FTTN R3.7/FGN3.8.

Note 2 — In auditNode operation response messages when there is a misalignment between the expected and actual values of password parameters, the parameter values are displayed in MD5 format. In auditPort operation response messages, the values of misaligned password parameters are displayed in clear text.

4.8 auditPort operation

The auditPort operation is an audit and reconciliation operation that determines whether the target configuration for a port that is stored by the 5529 APC matches the actual configuration for the port on the NE. The target configuration can include multiple templates and their arguments.

The auditPort operation can only be used on root ports for auditing all of the services configured on a root port. The operation does not require a template to be specified in the request operation.

If the inExcess optional parameter is set to true, the operation audits the port for configurations that are compatible with the service templates applied on the port, but not configured by those service templates. The first discrepancy that is identified during the inExcess audit is reported in the audit report along with the regular audit report of templates that are configured on the port.



Note 1 — The auditPort operation with the inExcess parameter set to true does not work when the G7342 ONT CES Service template is configured under the G7342 ONT CES Port template.

Note 2 — The auditPort operation is not supported in a high subscriber density configuration; see section 3.6 for information about the 5529 APC and high subscriber density configurations.

Note 3 — The auditPort operation will take longer than three to six seconds to process template stacks that contain the following network infrastructure services: SAP, VLAN, and VVPLS. Their respective 5529 APC infrastructure service templates are: ISAM SAP, ISAM VLAN for IHUB-Based ISAM, and ISAM VVPLS.

See section [4.2.1.1](#) for information about specifying a target object for the auditPort operation when more than one object matches the target object ID.

Table [18](#) describes the parameters for the auditPort operation request.

Table 18 auditPort operation request parameters

Parameter	Description
objectName	Indicates which port to audit.
inExcess	Indicates whether the operation should audit the port for configuration in excess of the configuration set by the templates being audited for. If this parameter is not passed, the port is not audited for configuration in excess of the applied templates. This parameter is supported for the 7302 ISAM and the 7330 ISAM FTTN. This parameter is optional.

Ports that have not been configured using the 5529 APC or that have been cleaned can also be audited using the auditPort operation. When the auditPort operation is used on a clean or unconfigured port, an affirmative response is returned by the 5529 APC when the port is not configured on the NE. When the port that is being audited is configured in the NE, a PORT_NOT_CLEAN error code is returned. This operation allows ports that are unclean and configured, and which should be clean and unconfigured, to be discovered.

The 5529 APC allows you to audit optional attributes that have the -No-Value- special token as the default value. When -No-Value- is the attribute default value, the 5529 APC uses the NE default value as the expected attribute value. Some optional attributes do not have an NE default value defined and are omitted from the audit when the attribute value is set to -No-Value-. Audit is still supported when the attribute value is set to a value in the range. See the *5529 APC Template Reference* for an NE for information about attributes that do not have an NE default value defined.



Note — The functionality that allows the 5529 APC to use the NE default value as the expected attribute value when -No-Value- is the default value for an optional attribute is not supported on 7302 ISAM/7330 ISAM FTTN R3.7/FGN3.8.

Table [19](#) describes the auditPort operation response parameters.

Table 19 auditPort operation response parameters

Parameter	Description	See
auditPortResponse	Container element for the auditPort operation response message	—
auditResult	Container element for the auditPort operation response parameters	—
serviceTemplateId	Container element for the templateName, templateVersion, and instanceLabel parameters	—
templateName	The name of the template that is configured for the port according to the 5529 APC	—
templateVersion	Version of the template	Section 4.2.3

(1 of 2)

Parameter	Description	See
instanceLabel	Only applicable for multi-instance templates. Not applicable for regular templates. Nokia recommends the use of regular templates. To distinguish multi-instance templates from regular templates, see section 4.30.	Section 4.2.5
auditReport	Container element for the templateType, configured, and remoteFailedAssertion parameters	—
templateType	The type of template	—
configured	Indicates whether the templateName is configured for the port according to the NE	—
remoteFailedAssertion	Indicates each configuration difference between the 5529 APC and the NE. A remoteFailedAssertion item is displayed for each difference. Each item indicates the expected and actual parameter values.	—

(2 of 2)

The response contains the information for each template that is configured for the port. See chapter 7 for examples of successful and failed auditPort operation response messages.

4.8.1 Audit misalignment for base stacks with applied templates

In some cases, when you attempt to unconfigure a base stack or delete part of a base stack that has a template applied to it, an audit misalignment occurs. The 5529 APC detects the applied template, fails the unconfigure or delete operation, and rolls back to the previous configuration. The applied template is deleted on the NE, but not on the 5529 APC, causing an audit misalignment. To fix the audit misalignment, reapply the template to the base stack.

Table 20 lists the stacks and conditions that can result in an audit misalignment for base stacks with applied templates.

Table 20 **Audit misalignment for base stacks with applied templates**

Base stack	Template applied on base stack	Operation	Audit misalignment occurs for
ISAM ONT stack that has a VLAN Association template VoIP Support attribute in the ISAM ONT template is configured to Proprietary VoIP OMCI	ISAM ONT IP Host template	Unconfigure the base stack	ISAM ONT IP Host template
ISAM ONT stack that has a VLAN Association template VoIP Support attribute in the ISAM ONT template is configured to Standard VoIP OMCI	ISAM ONT FTP Based VoIP Service template	Unconfigure the base stack	ISAM ONT FTP Based VoIP Service template
ISAM ONT stack that has an ISAM ONT CES Port template and a VLAN Association template	ISAM ONT CES Service template	Delete the corresponding VLAN association on the UNI	ISAM ONT CES Service template

4.9 check operation

The check operation is a service state verification operation that allows the client application to retrieve the operational status of a specified object to determine whether basic service capabilities are available based on the template for the object. The check operation considers all templates on the specified object when retrieving the operational status. The check operation does not cause modifications to target objects or templates. The check operation behaves atomically.

The check operation also allows the client application to retrieve non-status attribute information from the NE. See Table 22 for a list of supported attributes.

The check operation parameters are `objectName` and `statusExpressionList`. The `statusExpressionList` parameter consists of a list of status expressions that are to be returned by the check operation. Status expressions evaluate the selected status attributes for the applicable object. The values for the selected status attributes are returned in the check operation response. The `statusExpressionList` parameter is optional.

The response message for the check operation returns the value of each status expression. If input was not provided for the `statusExpressionList` parameter in the check operation request, all of the status expressions that are available on the templates that have been applied to the target object are returned, including the templates that have been applied on top of the templates that are directly applied to the target object. An error code is returned if a status expression is specified in the request but the templates that support the status expression have not been applied to the target object.

If more than one instance of a template is applied to the target object and the template supports status expressions, the response message returns the status expressions from all instances of the template. The status expressions are grouped under the service template ID to which each of the template instances belongs.

The status expressions that are provided by multi-instance templates are retrieved using the check operation.

Table 21 describes the parameters for the check operation request.

Table 21 check operation request parameters

Parameter	Description
<code>objectName</code>	The name of the port for the operation or the name of the object created on top of the port. The target object must be an object that was used previously to configure templates on the port.
<code>statusExpressionList</code>	Container element for the list of status expressions that are to be returned by the check operation. See Table 22 for examples of status expressions for the <code>statusExpressionList</code> parameter. This parameter is optional.

Table 22 provides examples of status expressions for the statusExpressionList parameter.

Table 22 statusExpressionList status expression examples

Item	Description
statusExpression	A container element for each operational status or non-status attribute to be retrieved. The statusExpression contains the statusName.
statusName	<p>Specifies the operational status or non-status attribute information to be retrieved. The following are some examples of the use of statusName by the check operation to retrieve operational status (applicable templates are provided for reference ⁽¹⁾):</p> <ul style="list-style-type: none"> • states_g6CpeOperationalState (H248 CPE template) • states_g6LineOperationalState (H248 LINE template) • gponOntSwActivationStatus and gponOntOperationalStatus (G7342 ONT template) • mauParams_AutoNegotiationCapabilityBits (ISAM Gigabit Ethernet Access template) • ontSwActivationStatus and ontOperationalStatus (ISAM ONT template) • states_OperationalState (ISAM POTS template) • states_AdministrativeState (ISAM SDP template) • isamVclOperationalStatus (ISAM Terminated VCL and ISAM Terminated VCL (FGN) template) <p>The following is how the statusName is used by the check operation to retrieve values for non-status attributes directly from the NE (applicable templates are provided for reference ⁽¹⁾):</p> <ul style="list-style-type: none"> • identification_EquippedType (ISAM ONT template) • ontSerialNumber (ISAM ONT template) • identification_Version_ActiveSoftware (ISAM ONT template) • ontSoftwarePlannedVersion (ISAM ONT template) • versionsFromOntSwVersionControl_PlannedSoftware (ISAM ONT template)

Note

⁽¹⁾ See the *5529 APC Template Reference* for information about the templates that support operational status or non-status attributes.

Table 23 describes the check operation response.

Table 23 check operation response items

Item	Description
serviceTemplateId	Container element for the templateName, templateVersion, and instanceLabel
templateName	Name of the template
templateVersion	Version of the template
instanceLabel	Only applicable for multi-instance templates. However, Nokia recommends that you use regular templates. To distinguish multi-instance templates from regular templates, see section 4.30.
statusExpressionList	Container element for the list of status expressions for the check operation
statusExpression	Container element for the statusName and statusValue for each retrieved operational status or non-status attribute

(1 of 2)

Item	Description
statusName	Indicates the name of the operational status or non-status attribute that was supplied in the check operation request. See Table 22 for a list of examples of supported status names.
statusValue	<p>Indicates the value of the operational status or non-status attribute information as retrieved from the NE.</p> <p>Values for operational status include:</p> <ul style="list-style-type: none">• ENABLED - the object is operational• DISABLED - the object is not operational• ACTIVE - the planned software and active software match• INACTIVE - the planned software and active software do not match <p>Values for non-status attributes retrieved directly from the NE include:</p> <ul style="list-style-type: none">• ONT hardware type (for example, O-211M-G)• ONT serial number (for example, ALCLA0B10B89)• Active ONT software version (for example, 3FE50854EHRA06)• Planned ONT software version (for example, AUTO)• Auto ONT software planned version (for example, 3FE50854EHRA06)

(2 of 2)

4.10 clean operation

The clean operation is an audit and reconciliation operation that allows the client application to completely unconfigure the port or ONT identified by the objectName parameter. You can use the clean operation if the port or ONT configuration does not match the port or ONT configuration of the 5529 APC or the one known at the client application (unsuccessful audits). The configuration for the port or ONT is deleted from the NE and the 5529 APC. The clean operation is service affecting. After a successful clean operation, the port or ONT can be configured with new template configurations.

The clean operation also completely unconfigures all instances of multi-instance templates, if applicable.



Note 1 — A clean operation executed on an ONT MDU port deletes the services that are known to the 5529 APC from the 5529 APC and from the ONT. The clean operation does not remove the objects or services that are not known to the 5529 APC from the ONT MDU port.

Note 2 — The clean operation works correctly when run on the root port. If you run the clean operation on a non-root port, an unexpected result can occur.

Note 3 — The clean operation is not supported in a high subscriber density configuration; see [section 3.6](#) for information about the 5529 APC and high subscriber density configurations.

Note 4 — If you perform the clean operation on an SHDSL span object while a 5520 AMS search tone test is in progress on the object, the clean operation will fail and a 5529 APC audit will report the deleted service as TARGET_OBJECT_MISSING_IN_NE. You can perform the reapplyService or reapplyAllServices operation to reconcile the audit discrepancy and restore the deleted service or services.

Note 5 — The clean operation will take longer than three to six seconds to process template stacks that contain the following network infrastructure services: SAP, VLAN, and VVPLS. Their respective 5529 APC infrastructure service templates are: ISAM SAP, ISAM VLAN for IHUB-Based ISAM, and ISAM VVPLS.

See [section 4.2.1.1](#) for information about specifying a target object for the clean operation when more than one object matches the target object ID.

[Table 24](#) describes the parameters for the clean operation request. By default, there are no parameters for the response unless there is an error. For information about how to add a parameter to the response to explicitly state that the operation was a success, see [section 4.2.7](#).

Table 24 **clean operation request parameters**

Parameter	Description	See
objectName	The name of the port for the operation or the name of the object created on top of the port	<i>5529 APC Template Reference</i> for identification information for objects and port types
operationInitiator	The system and user ID of the user who is initiating the change to the port or object. This parameter is optional.	Section 4.2.6

4.11 configure operation

The configure operation is a Service provisioning operation that configures the port specified by the `objectName` parameter or an object created on top of the port. The configuration for the NE is identified by the template type, as specified by the `templateName` and `templateVersion` parameters, and the template arguments. A template instance can configure physical ports and the derived logical ports only once. A successful configure operation updates the 5529 APC with the service configuration. The 5529 APC uses the information for functions such as audit and network-wide line or template modifications.

The configure operation can be used more than once to configure specific services at the same time. The configure operation behaves atomically. If there is an error during the execution of an operation, a rollback is performed for the port by the 5529 APC. The validation of the argument value range is delegated by the 5529 APC to the NE. See section 4.3 for information about error handling.

In configure operations for service templates, the OSS client applications are not requested to send provisioned arguments with the configure request, as the configure operation has no charter to perform implicit modify operations for the provisioned arguments. To perform modify operations on provisioned arguments, you need to consider separate modify requests for the applicable arguments. Therefore, the configure requests with provisioned arguments that contain a value different from the latest value are declined; provisioned arguments do not need to be supplied with a configure request.

The 5529 APC does not support configuring the value of a mandatory attribute to be -No-Value-.



Note — The configure operation will take longer than three to six seconds to process template stacks that contain the following network infrastructure services: SAP, VLAN, and VVPLS. Their respective 5529 APC infrastructure service templates are: ISAM SAP, ISAM VLAN for IHUB-Based ISAM, and ISAM VVPLS.

See section 7.1.1 for examples of the configure operation request and response xml.

Table 25 describes the parameters for the configure operation request. By default, there are no parameters for the response unless there is an error. For information about how to add a parameter to the response to explicitly state that the operation was a success, see section 4.2.7.

Table 25 configure operation request parameters

Parameter	Description	See
<code>objectName</code>	The name of the port for the operation or the name of the object created on top of the port	<i>5529 APC Template Reference</i> for identification information for objects and port types
<code>templateName</code>	The name of the template for the operation	Section 4.2.2

(1 of 2)

Parameter	Description	See
operationInitiator	The system and user ID of the user who is initiating the change to the port or object. This parameter is optional.	Section 4.2.4
instanceLabel	Only applicable for multi-instance templates. Not applicable for regular templates. Nokia recommends the use of regular templates. To distinguish multi-instance templates from regular templates, see section 4.30 .	Section 4.2.5
templateVersion	Template version	Section 4.2.3
argument	Template argument	Section 4.2.4

(2 of 2)

4.12 dbClean operation

The dbClean operation is an audit and reconciliation operation that allows the client application to delete all of the template configuration that is associated with the port identified by the objectName parameter at the 5529 APC level. The port configuration in the NE is not affected.

The dbClean operation also deletes all instances of multi-instance templates, if applicable.



Note — The dbClean operation is not supported in a high subscriber density configuration; see section [3.6](#) for information about the 5529 APC and high subscriber density configurations.

Table [26](#) describes the parameters for the dbClean operation request. By default, there are no parameters for the response unless there is an error. For information about how to add a parameter to the response to explicitly state that the operation was a success, see section [4.2.7](#).

Table 26 dbClean operation request parameters

Parameter	Description	See
objectName	The name of the port for the operation or the name of the object created on top of the port	<i>5529 APC Template Reference</i> for identification information for objects and port types
operationInitiator	The system and user ID of the user who is initiating the change to the port or object. This parameter is optional.	Section 4.2.6

4.13 dbUnconfigure operation

The dbUnconfigure operation is an audit and reconciliation operation that removes a service from the 5529 APC only. The dbUnconfigure operation does not remove the service from the NE; therefore, the operation does not require an NE to be reachable. The 5529 APC does not execute the dbUnconfigure operation if one or more templates are present on top of the template that is being unconfigured; for example, when the template has a serving dependency to other templates.

Table 27 describes the parameters for the dbUnconfigure operation request. By default, there are no parameters for the response unless there is an error. For information about how to add a parameter to the response to explicitly state that the operation was a success, see section 4.2.7.

Table 27 dbUnconfigure operation request parameters

Parameter	Description	See
objectName	The name of the port or object for the operation. The objectName corresponds to the objectName that was used in the configure operation.	<i>5529 APC Template Reference</i> for identification information for objects and port types
templateName	The name of the template for the operation	Section 4.2.2
templateVersion	Version of the template. When you do not specify a template version, the 5529 APC unconfigures the template version that is configured on the port or object. This parameter is optional.	Section 4.2.3
instanceLabel	The current instance label of the template for the operation	Section 4.2.5
argument	Argument value list. This parameter is optional.	Section 4.2.4
operationInitiator	The system and user IDs of the user who is initiating the change on the port or object. This parameter is optional.	Section 4.2.6

4.14 discoverAllServices operation

The discoverAllServices operation is an audit and reconciliation operation that supports the reconciliation process for ports or ONTs from the NBI. The discoverAllServices operation supports changing the actual configuration on a port or ONT into an intended configuration in terms of 5529 APC service templates and argument values stored in 5529 APC.

Table 28 lists additional notes about the discoverAllServices operation.

Table 28 Notes about the discoverAllServices operation

Related functionality	Note
5529 APC NBI bulk functionality	The discoverAllServices operation is not supported by 5529 APC NBI bulk functionality.
Bind templates	The bind template (or list of bind templates) that is specified as the candidate template (templateName parameter) in the discoverAllServices operation request needs to apply to the NE type for which you are running the operation. For example, if you run the discoverAllServices operation on a 7342 ISAM FTTU NE and you include the applicable G6-G7342-VOICE-BIND template and the G6-G7367-VOICE-BIND template that is not applicable, the operation will fail.
Discovery of services that contain network infrastructure object types	<p>For 7302/7330/7356/7360/7362/7363/7367 ISAM R5.4 or later, when used with applicable templates, the discoverAllServices operation can discover services that contain the following network infrastructure object types and their children templates: VLAN, VVPLS, and SAP.</p> <p>The following are the applicable templates:</p> <ul style="list-style-type: none"> • ISAM SAP • ISAM VLAN for IHUB-Based ISAM • ISAM VVPLS <p>The discoverAllServices operation will take longer than three to six seconds to process template stacks that contain the following network infrastructure services: SAP, VLAN, and VVPLS. Their respective 5529 APC infrastructure service templates are: ISAM SAP, ISAM VLAN for IHUB-Based ISAM, and ISAM VVPLS.</p>
Discovery of services without removing the 5529 APC database	When discovering services without removing the 5529 APC database, when one attribute depends on another attribute, and both attributes are configured as -No-Value- in the 5529 APC and configured to a different value in the 5520 AMS, a misalignment may occur. Perform the discoverAllServices operation twice to fix the misalignment.
High subscriber density configurations	The discoverAllServices operation is not supported in a high subscriber density configuration; see section 3.6 for information about the 5529 APC and high subscriber density configurations.
Multiple matched rules in advanced virtual argument mapping	The discoverAllServices operation may fail if there are multiple matched rules with one discovered data (basic attributes value from NE) in advance mapping.
Reverse mapping of advanced virtual arguments	If reverse mapping of an advanced virtual argument does not find a match with the defined mapping, but matches the default value, the discoverAllServices operation is not able to perform a mapping of the discover value and the 5529 APC returns a NO_MATCHING_TEMPLATE error.

Because all port or ONT configurations in the network are expected to be expressed in terms of pre-defined service templates (services deployed on a port or ONT need to be part of a portfolio of sold and defined products), the discoverAllServices operation tries to match the actual port or ONT configuration to a set of candidate service templates. The operation tries to match as many managed objects on the port or ONT as possible, but whether it succeeds or not depends on the set of candidate service templates.

The discoverAllServices operation discovers services configured in the network on the endpoint specified in the objectName attribute, and updates the 5529 APC when the port or ONT configuration matches any of the candidate service templates that are provided to the operation, and are not yet known to the 5529 APC.

Next to the reapply operations, the `discoverAllServices` operation provides the means to resolve audit discrepancies in the configured services by updating the 5529 APC. The purpose of running the `discoverAllServices` is to align the 5529 APC with the network based on pre-defined and selected service templates. The `discoverAllServices` operation does not affect the port or ONT configuration in the NE.

The `discoverAllServices` operation executes an iterative process of trying to resolve typical audit misalignments by looking for service templates that match in a list of candidate service templates. It may cause any of the assigned service templates on the port or ONT to be unassigned from the port or ONT, and to be replaced by a better matching service template from the candidate service template list. When a correct set of candidate service templates has been supplied, the operation results in a port or ONT that is fully aligned again.

Parameter mismatches (fixed template attributes that are misaligned) cause the removal of service templates that have a parameter mismatch from the port or ONT in the 5529 APC, and their replacement by a candidate service template that also matches the attribute value in the NE. The 5529 APC tries to match service templates on a port or ONT for which some objects are missing in the NE with supplied candidate service templates.

The 5529 APC resolves argument mismatches (template arguments that are misaligned) by taking the attribute value found on the port or ONT in the NE and storing it as an intended value in 5529 APC (this applies to modifiable and non-modifiable attributes).

The 5529 APC tries to match `inExcess` objects on a port or ONT with the supplied candidate service templates.

A service template matches a configuration in the port or ONT of the NE when all basic templates of the service template (in a single OR leg of the service template) match the corresponding managed objects in the NE. Each template parameter needs to be equal to the attribute value in the NE, and the template argument is assigned the attribute value of the managed object in the NE. If the service template has virtual arguments that map multiple virtual argument values (or a range of argument values) to a single argument value, then the 5529 APC can select any of the virtual argument values that map to the single argument value. Typically, the 5529 APC selects the first value. When an argument is mapped to an AVG, the 5529 APC uses the attribute value in the NE as the argument value.

The `discoverAllServices` operation loops through the candidate service templates in the order that you have placed them in the `templateInfo` container element, and tries to find matching NE services for each template. In theory, multiple different combinations of service templates can cover a single port or ONT configuration; therefore, the order of the selected service templates in the `templateInfo` container element is considered the preferential order for discovery.

The order of the templates in the list is important. For best results, apply the following guidelines when listing service templates:

- Place the service templates that are expected to match higher in the list.
- Place the service templates that are typically configured before other service templates higher in the list.

- Place the service templates that are expected to replace misaligned service templates higher in the list. It is better not to select service templates that are aligned and have no inExcess objects, and you need to keep, unless the templates have lower-level misalignments.
- Place service templates that are commonly deployed in the network higher in the list.
- Service templates with only misaligned arguments can be included in or excluded from the candidate service template list.

5529 APC services that are aligned and have no inExcess objects are valid, and therefore are not substituted by alternative matching templates in the candidate service template list. If that is, however, the purpose of a discovery, then Nokia recommends the use of the REPLACE_ALL strategy argument with the appropriate candidate service template list.

The discoverAllServices operation supports the use of a strategy argument. The REPLACE_ALL strategy enables the full discovery of the complete port or ONT: any service templates and arguments that the 5529 APC keeps for the port or ONT are discarded. When the strategy argument is specified and set to REPLACE_ALL, the discoverAllServices operation removes all of the services on the port or ONT from the 5529 APC (however, not from the actual port or ONT configuration in the network) before attempting to discover the services that match the provided candidate service templates in the network.

The discoverAllServices operation discovers new services only on the basis of the list of templates supplied to the operation. If the candidate template list contains no templates that can match a network service, the discoverAllServices operation does not discover that network service.

When multi-instance templates are successfully matched, the discoverAllServices operation generates an instance label for each discovered service in the following format: *instance-number*, for example, instance-1, instance-2, and so on. See section 4.35 for information about changing the instance label.

If no matching service template is found, the discoverAllServices operation does not alter the service templates and arguments assigned to the port or ONT. After one or more matching service templates are found, depending on the misalignment, the matching service templates are added to the assigned service templates of the port or ONT. The misaligned service templates can be removed or replaced by other templates, and the new service template argument values can be stored. Then the OSS client can perform a getConfiguredTemplates operation on the port or ONT to find the aligned service templates of the port or ONT after the discoverAllServices operation.

Table 29 describes the parameters for the discoverAllServices operation request. By default, there are no parameters for the response unless there is an error. For information about how to add a parameter to the response to explicitly state that the operation was a success, see section 4.2.7.

Table 29 **discoverAllServices operation request parameters**

Parameter	Description	See
objectName	The address of the object for which users run the discoverAllServices operation The operation supports the following objects: ONT (non-MDU), MDU port, and physical port.	<i>5529 APC Template Reference</i> for identification information for objects and port types
templateInfo	Container element for the templateName and templateVersion parameters. The templateInfo parameter represents a list of candidate service templates for the discovery algorithm. The order of the templateInfo items in the list is important; see the operation description for the guidelines on placing templates in the list. The 5529 APC processes the templateInfo items in the sequence provided in the request. The templateInfo container can include as many templateName and templateVersion parameter pairs as required, or contain no templateName and templateVersion parameters. If the templateInfo parameter is not supplied, the 5529 APC tries to align the templates on the port.	—
templateName	The name of a candidate template for the operation You can list multiple candidate template names in the templateInfo container element, as required. The order of the templates in the list is important. See the operation description for guidelines on placing templates in the list. If an invalid template (a template that is not known to the 5529 APC) is supplied in the list of candidate templates, then the 5529 APC fails the request and returns a TEMPLATE_UNKOWN error with an error message containing the name of the invalid template.	Section 4.2.4
templateVersion	The version of a candidate template for the operation You can provide a template version for every service template listed in the templateInfo container.	Section 4.2.3
strategy	The discoverAllServices operation supports the use of a strategy argument. The REPLACE_ALL strategy enables the full discovery of the complete port or ONT: any service templates and arguments that the 5529 APC keeps for the port or ONT are discarded. This parameter is optional.	—
operationInitiator	The system and user IDs of the user who is initiating the change on the port or object. This parameter is optional.	Section 4.2.6

The 5529 APC reports the discoverAllServices operation in the configuration change notification interface because the operation can alter the 5529 APC configuration for a port or an ONT. As a result, the 5529 APC informs subscribed clients about the result of the discoverAllServices operation as part of the reconciliation process.

This execution time of the discoverAllServices operation depends on the number of candidate service templates supplied as input to the operation. Therefore, Nokia recommends that users and OSS clients keep the list of candidate service templates supplied for matching as short as possible; for example, it makes no sense to supply service templates that are not applicable to the port type, the technology, or the subscribing customer class.

An empty candidate service template list for the discoverAllServices operation means that the current service templates on the port or ONT are used as the candidate service template list in the discoverAllServices algorithm, in order of configuration on the port or ONT. With regard to new services, only additional new multi-instance service templates can be discovered (a subcategory of in excess object cases in the NE), and argument mismatches can be resolved. An empty candidate service list will most likely not help to resolve parameter mismatches or in excess templates in the 5529 APC, as no new service templates are supplied (even though a full discovery algorithm will be attempted).

If the candidate service template list is not empty, current service templates on the port or ONT are not considered during the discovery and matching process. If the current templates on the port or ONT are to be considered as well, they need to be added to the candidate service list.

A rollback occurs if no new service template was matched and parameter mismatches and in-excess alignments remain. If no new service templates are matched and there are no remaining parameter mismatches and in-excess alignments, the argument discoveries are retained (and there is no rollback executed).

See the *5529 APC Template Reference* for an NE type for information about the NE releases and 5529 APC templates that are not supported for the discoverAllServices operation. To request discoverAllServices operation support for additional templates, contact your Nokia technical support representative.

4.15 execute operation

The execute operation is a bulk execution operation that allows the client application to specify service provisioning operations that are performed in the sequence that is specified in a web service request. The instanceLabel for a template may be provided, if applicable.

When you perform bulk operations using the execute operation, Nokia recommends that you configure the Max Concurrent Job Per NE parameter to 1 in the APC Settings, so that sequential operations are limited to a single NE. See the *5529 APC Installation, Administration, and User Guide* for information about configuring the Max Concurrent Job Per NE parameter.

Table 30 describes the parameters for the execute operation request.

Table 30 execute operation request parameters

Parameter	Description	See
command	Container element for the execute operation request parameters	—
commandId	The command identifier is an integer that is provided by the client application to identify the command in a bulk request. The value must be unique for each command in the bulk request.	—

(1 of 2)

Parameter	Description	See
action	The supported operations for the bulk request are: AUDIT, AUDITPORT, CHECK, CLEAN, CONFIGURE, DBCLEAN, DBUNCONFIGURE, IMPORTTEMPLATE, MIGRATE, MODIFY, REAPPLYALLSERVICES, REAPPLYSERVICE, RESUME, SUSPEND, and UNCONFIGURE.	—
objectName	The name of the port or object for the operation specified by the Action parameter	<i>5529 APC Template Reference</i> for identification information for objects and port types
templateName	The name of the template for the operation	Section 4.2.2
instanceLabel	Only applicable for multi-instance templates. Not applicable for regular templates. Nokia recommends the use of regular templates. To distinguish multi-instance templates from regular templates, see section 4.30.	Section 4.2.5
templateVersion	Template version	Section 4.2.3
oldTemplateName	The name of the template that is currently applied to the port or object	—
oldInstanceLabel	Label that identifies the template instance that is currently applied to the port or object	—
oldTemplateVersion	The version of the template that is currently applied to the port or object	—
targetDate	Time and date for the operation in the format specified in: http://www.w3.org/2001/XMLSchema Currently, this parameter does not affect the behavior of the 5529 APC.	—
priority	Currently, this parameter does not affect the behavior of the 5529 APC.	—
state	The current state of the template (RELEASED or PHASED-OUT). Container element for the state name and value parameters.	—
argument	Template argument. Container element for the argument name and value parameters.	Section 4.2.4
strategy	Specifies the template migration strategy	Table 63
forceModify	Allows you to modify arguments that are not modifiable	Section 4.34.1
inExcess	Indicates whether or not the operation should audit the port or NE (all ports in NE) for a configuration in excess of the configuration set by the templates that are being audited.	—
operationInitiator	The system and user ID of the user who is initiating the change to the port or object. This parameter is optional.	Section 4.2.4

(2 of 2)

The execute operation is answered immediately. However, the commands in the request are performed when 5529 APC system resources are available. The execute operation returns a *requestId* from the 5529 APC that allows the client application to request the status of the commands that have not been performed. The *requestId* and *commandId* pair is unique because the specified *commandId* in an execute operation must be unique in a bulk request.



Note — If the request contains any unsupported command, the query result will return an error.

See chapter 7 for examples of the execute operation request and response messages as they apply to template migration.

4.16 File-based operations

The file-based operations are bulk execution operations that allow the client application to request the execution of commands in batch and to monitor the execution using files. The files are located in an accessible file system on the server. The file contents are in XML and correspond to the XML element definitions that are used in the request or response message. Table 31 describes the file-based operations.

Table 31 File-based operations

Operation	Parameter	Description
executeUrl	url	<p>The location of the file is specified by the URL in the input parameters. The executeUrl operation returns a requestId which specifies the bulk operation requests that can be used by the client application. The executeUrl operation immediately returns a response. The commands are performed in the sequence of appearance in the file.</p> <p>The following are examples of URLs:</p> <ul style="list-style-type: none"> file:/var/tmp/apc-northbound/result_8080808080809A80.xml file:/Nokia/configure.txt <p>When you perform bulk operations using the execute operation, Nokia recommends that you configure the Max Concurrent Job Per NE parameter to 1 in the APC Settings, so that sequential operations are limited to a single NE. See the <i>5529 APC Installation, Administration, and User Guide</i> for information about configuring the Max Concurrent Job Per NE parameter.</p>
getResultUrl	requestId commandId	<p>Requests the status of the command identified by the commandId that is associated with the bulk operation identified by the requestId. The getResultUrl returns a URL that indicates the path to the file that contains the results.</p> <p>The IP address of the 5520 AMS server from which the operation is sent is added to the front of the URL string; for example, <i>IP_address:/var/opt/ams/local/ams-release-revision/apc/apc-northbound/result_5.xml</i>. The result file URL can be accessed on the 5520 AMS server from which the operation is sent.</p>
getResultsUrl	requestId	<p>Requests the status of commands that have been performed, as identified by the requestId. The getResultsUrl returns a URL that indicates the path to the file with the results.</p> <p>The IP address of the 5520 AMS server from which the operation is sent is added to the front of the URL string; for example, <i>IP_address:/var/opt/ams/local/ams-release-revision/apc/apc-northbound/result_5.xml</i>. The result file URL can be accessed on the 5520 AMS server from which the operation is sent.</p>

4.17 get operation

The get operation is a service state verification operation that allows the client application to retrieve the values of certain data attributes of dedicated objects that are useful in the service provisioning process. When retrieved, these data attribute values can be used for service provisioning.

Before and after running a get operation, observe the following conditions:

- The target object of the get operation must exist.
- The get operation does not cause modifications to target objects or templates.

Table 32 describes the parameters for the get operation request.

Table 32 get operation request parameters

Parameter ⁽¹⁾	Description	Identification
objectName	For the PON attributes, the objectName is the NE name and PON port number.	Format: <i>NE:rack-subrack-slot-port</i> , where: <ul style="list-style-type: none"> • <i>NE</i> is the name of the NE • <i>rack</i> is the rack number • <i>subrack</i> is the subrack number • <i>slot</i> is the slot number • <i>port</i> is the PON port number
	For the MPLS and LSP attributes, the objectName is one of the following: <ul style="list-style-type: none"> • NE name • NE and PON port number 	NE format: <i>NE</i> , where <i>NE</i> is the name of the NE PON port format: <i>NE:rack-subrack-slot-port</i> , where: <ul style="list-style-type: none"> • <i>NE</i> is the name of the NE • <i>rack</i> is the rack number • <i>subrack</i> is the subrack number • <i>slot</i> is the slot number • <i>port</i> is the PON port number
argumentInfoList	Container element for the list of arguments that are to be returned by the get operation	—
argumentInfo	Container element within the argumentInfoList parameter that contains the list of argument names	—
name ⁽²⁾	Specifies the name of the argument	See Table 33 for a list of the supported argument names.
additionalInfo ⁽²⁾	Container element at the same level as the argumentInfo parameter that specifies additional information about the target object (attribute name and value) to narrow the scope of the get operation request; for example, the serial number of an ONT	—
name ⁽²⁾	Specifies the name of the attribute that is the additional information used to narrow the scope of the get operation request; for example, ontSerialNumber	<i>Attribute name</i>
value ⁽²⁾	Specifies the attribute value; for example, ALCL00000001	<i>Attribute value</i>

Notes

- ⁽¹⁾ A get operation can only be requested for attributes on a single object. You cannot run a get operation on multiple objects at the same time. Separate get operation requests need to be made for attributes on different objects. Also, you cannot run a get operation for an attribute that does not exist for an object (for example, you cannot request the next free LSP ID from a PON port).
- ⁽²⁾ Optionally, you can use the additionalInfo attributes to specify an ONT serial number and specify the nextFreeOntId attribute as the value of the argumentInfo name attribute to return the next free ONT ID on a PON. If the ONT serial number does not exist on the PON, the get operation returns the next free ONT ID. If an ONT serial number exists on the PON, the get operation does not return an ONT ID value. See Table 33 for information about the nextFreeOntId attribute.

Table 33 lists the attributes that can be retrieved using the get operation. The attribute values are retrieved from the NE.

Table 33 Attributes retrieved by the get operation

Attribute	Values	Notes
PON attributes		
eponPonGetNextFree	The next free ONT ID on the PON port. A 0 is returned if there are no more free ONT IDs.	The get operation does not reserve the ONT ID. Any delay in sending the configure operation request may result in a failure of the configure operation because the free ONT ID may be used to create another ONT on the same PON port in the meantime. Applies to EPON technology.
nextFreeOntId	The next free ONT ID on the PON port. A 0 is returned if there are no more free ONT IDs.	The get operation does not reserve the ONT ID. Any delay in sending the configure operation request may result in a failure of the configure operation because the free ONT ID may be used to create another ONT on the same PON port in the meantime. Applies to ISAM, GPON, and EPON technologies. Optionally, you can specify the nextFreeOntId attribute as the value of the argumentInfo name attribute and specify an ONT serial number for the additionalInfo attributes to return the next free ONT ID on a PON. If the ONT serial number does not exist on the PON, the get operation returns the next free ONT ID. If an ONT serial number exists on the PON, the get operation does not return an ONT ID value.
maxNumberOfOnt	The maximum number of ONTs that can be created on the PON port.	—
MPLS and LSP attributes		
mplsTunnelIndexNext	The next free MPLS path ID on an NE. A 0 is returned if there are no more free MPLS path IDs.	The get operation does not reserve the MPLS path ID. Any delay in sending the configure operation request may result in a failure of the configure operation because the free MPLS path ID may be used to create another MPLS path on the same NE in the meantime.
vRtrMplsGeneralNewLspIndex	The next free LSP ID on an NE.	The get operation does not reserve the LSP ID. Any delay in sending the configure operation request may result in a failure of the configure operation because the free LSP ID may be used to create another LSP on the same NE in the meantime.

Table 34 describes the get operation response parameters.

Table 34 get operation response parameters

Parameter	Description	See
argumentList	Container element for the list of arguments returned by the get operation	—
argument	Container element for the name of the argument that was supplied in the get operation request and the value of the argument as retrieved from the NE	Table 33
name	Provides the name of the argument	Table 33
value	Provides the value of the argument	—

4.18 getApplicableTemplates operation

The getApplicableTemplates operation is a service inventory operation that provides information about the templates that can be applied to the specified port or, depending on the value of the exclusive parameter, to a port of the same class.

The getApplicableTemplates operation uses the objectName to query the 5529 APC. You use the operation on physical ports or ONTs only.



Note — The getApplicableTemplates operation needs to reach the NE in order to load the ANSI version software. When the NE is unreachable and the ANSI version software does not load, the operation returns an error message.

The templates in the list depend on the value of the exclusive parameter:

- When the exclusive parameter is set to True, the list of templates contains the templates that are compatible with the templates that have been applied to the port. Composite templates that have shared templates that are compatible with the templates that have been applied to the port are also contained in the list.
- When the exclusive parameter is set to False, the list of templates contains the templates that are compatible with the port class. Composite templates that have shared templates that are compatible with the port class are also contained in the list.

Table 35 describes the parameters for the getApplicableTemplates operation request.

Table 35 getApplicableTemplates operation request parameters

Parameter	Description	See
objectName	The name of the port or object	Section 4.2.1 <i>5529 APC Template Reference</i> for identification information for objects and port types
exclusive	Boolean value that indicates whether the templates that returned can be configured on top of the template for the current port (True) or whether the templates that are returned match the class of the port (False)	Section 4.18

Table 36 describes the `getApplicableTemplates` response parameters. The `getApplicableTemplates` operation returns templates, which can be multi-instance templates or non-multi-instance template; however, the operation does not return the instances of the multi-instance template.



Note — Table 36 lists only the elements that are used by the `getApplicableTemplates` operation. There may be additional optional elements that are displayed in the WSDL file (for example, `instanceLabel`), but those elements are not used by the `getApplicableTemplates` operation.

Table 36 `getApplicableTemplates` operation response parameters

Parameter	Description	See
templateId	Container element for the name and version parameters if applicable.	—
name ⁽¹⁾	The name of the template	Section 4.2.2
version	The version of the template	Section 4.2.3

Note

⁽¹⁾ When the `getApplicableTemplates` operation is performed on an object that has the ISAM NGPON2 ONT address format (*object:rack-subrack-slot*), ignore the ISAM NGPON2 ONT templates that are included in the response.

4.19 `getChangeHistory` operation

The `getChangeHistory` operation is a service state verification operation that allows the client application to retrieve the OSS application and user who performed the last change on the specified object, including the date and time of the change.

The `getChangeHistory` operation request argument is `objectName`. The `objectName` argument is the name of the physical port or ONT for the operation.

Table 37 describes the `getChangeHistory` operation response parameters.

Table 37 `getChangeHistory` operation response parameters

Attribute	Description
initiator	Specifies the originator of a Northbound interface operation that is the last change performed on the specified object. The value is a combination of the system/application name and user ID at the originating system of the request, for example, <code>MyServProvOSSxyz/user1</code> or <code>MyCRMabc/user2</code> .

(1 of 2)

Attribute	Description
timeStamp	<p>Time stamp of when the operation was performed. The time is the UTC standard where the 5529 APC is installed.</p> <p>The format of the time stamp is <i>YYYYMMDDHHMMSS.sZ</i></p> <p>where</p> <ul style="list-style-type: none">• <i>YYYY</i> is the year• <i>MM</i> is the month• <i>DD</i> is the day• <i>HH</i> is the hour• <i>MM</i> is the minute• <i>SS</i> is the second• <i>s</i> is the millisecond <p>Example: <timeStamp>20161110183447.1Z</timeStamp></p>

(2 of 2)

4.20 getConfiguredServices operation

The getConfiguredServices operation is a service inventory operation that retrieves service arguments, such as the ONT ID. The getConfiguredServices operation uses the objectName to query the 5529 APC.

The following target types are supported:

- DSL:
 - slot
 - port
- PON:
 - slot
 - PON port
 - ONT
 - ONT port (MDU ONT only)

You can filter the responses using an argument name and value combination. If no filter is specified, all of the services on the target object are included in the response.

The operation returns an empty response if:

- no services match the argument name and value combination specified in the request
- the operation is performed on an unsupported target type
- services were configured outside of the 5529 APC



Note 1 — The 5529 APC supports a maximum of five concurrent getConfiguredServices operations on different targets.

Note 2 — The getConfiguredServices operation is not supported on the GENBAND G6.

The `getConfiguredServices` operation can be used to retrieve the identifier of an ONT configured with a given service—provide the PON object identifier and filter on `ontSerialNumber` argument. The response lists the ONT identifier.

Table 38 describes the parameters for the `getConfiguredServices` operation request.

Table 38 `getConfiguredServices` operation request parameters

Parameter	Description	See
objectName	The name of the target object	Section 4.2.1 <i>5529 APC Template Reference</i> for identification information for objects and port types
filter	Container element for the argumentList parameter	—
argumentList	Container element for the list of template arguments	—
nameAndStringValue	Container element for the argument name and value	—
name	The name of the argument	—
value	The value of the argument	—

Table 39 describes the parameters for the `getConfiguredServices` operation response.



Note — Table 39 lists only the elements that are used by the `getConfiguredServices` operation. There may be additional optional elements that are displayed in the WSDL file (for example, state), but those elements are not used by the `getConfiguredServices` operation.

Table 39 `getConfiguredServices` operation response parameters

Parameter	Description	See
objectName	The name of the target object	Section 4.2.1 <i>5529 APC Template Reference</i> for identification information for objects and port types
templateName	The name of the template	Section 4.2.2
version	The version of the template	Section 4.2.3
instanceLabel	Only applicable for multi-instance templates. Not applicable for regular templates. Nokia recommends the use of regular templates. To distinguish multi-instance templates from regular templates, see section 4.30.	Section 4.2.5

(1 of 2)

Parameter	Description	See
tags	The service type identifier. If the value is not specified during template design, the tags parameter does not appear in the response.	—
operationInitiator	The system and user ID of the user who performed the last change on the port	Section 4.2.6
timeStamp	The time that the operation was performed	—
suspendstate	Indicates whether the service template is in a suspended state; the states are SUSPENDED or ACTIVE	—
argument	Container element for the list of arguments	Section 4.2.4
name	The name of the argument provided in the configure request	—
value	The value of the argument	—
endOfReply	Indicates the end of the reply. The value is true.	—

(2 of 2)

See chapter [7](#) for examples of the getConfiguredServices operation request and response messages.

4.21 getConfiguredTemplate operation

The getConfiguredTemplate operation is a service inventory operation that provides information about the template that was used to configure the port using the 5529 APC and the current state, which is SUSPENDED or ACTIVE.

The getConfiguredTemplate operation request uses the objectName to query the 5529 APC. You use the operation on physical ports or ONTs only.

Table [40](#) describes the getConfiguredTemplate operation request parameters.

Table 40 getConfiguredTemplate operation request parameters

Parameter	Description	See
objectName	The name of the port for the operation or the name of the object created on top of the port	Section 4.2.1 <i>5529 APC Template Reference</i> for identification information for objects and port types
templateName	The name of the template, which may indicate a composite template	Section 4.2.2
templateVersion	The version of the template	Section 4.2.3

(1 of 2)

Parameter	Description	See
instanceLabel	Only applicable for multi-instance templates. Not applicable for regular templates. Nokia recommends the use of regular templates. To distinguish multi-instance templates from regular templates, see section 4.30.	Section 4.2.5

(2 of 2)

Table 41 describes the getConfiguredTemplate operation response parameters.

Table 41 getConfiguredTemplate operation response parameters

Parameter	Description	See
objectName	The name of the port for the operation or the name of the object created on top of the port	Section 4.2.1 <i>5529 APC Template Reference</i> for identification information for objects and port types
templateName	The name of the template	Section 4.2.2
type	The type of template	—
version	The version of the template	Section 4.2.3
instanceLabel	Only applicable for multi-instance templates. Not applicable for regular templates. Nokia recommends the use of regular templates. To distinguish multi-instance templates from regular templates, see section 4.30.	Section 4.2.5
state	The current state of the template; for example, RELEASED	—
suspendstate	Indicates whether service template is in a suspended state; the states are SUSPENDED or ACTIVE	—
argument	Container element for the template argument	Section 4.2.4
name	The name of the argument	—
value	The value of the argument	—
serviceComponents	Container element for the serviceComponent, name, and suspendstate parameters	—
serviceComponent	Container element for the name and suspendstate parameters	—
name	Specifies the name of the service component (basic template)	—
suspendstate	Indicates whether the service component is in a suspended state	—

4.22 getConfiguredTemplateName operation

The getConfiguredTemplateName operation is a service inventory operation that provides a list of templates that were used to configure a port using the 5529 APC.

The template name in the response may specify a composite template, or may refer to any combination of template instances that were configured on the object.

Table 42 describes the getConfiguredTemplateName operation request parameters.

Table 42 getConfiguredTemplateName operation request parameters

Parameter	Description	See
objectName	The name of the port for the operation or the name of the object created on top of the port	Section 4.2.1 <i>5529 APC Template Reference</i> for identification information for objects and port types

Table 43 describes the getConfiguredTemplateName operation response parameters.

Table 43 getConfiguredTemplateName operation response parameters

Parameter	Description	See
templateId	Container element for the name, version, and state parameters, and the instanceLabel parameter, if applicable	—
name	Name of the template	Section 4.2.2
version	Version of the template	Section 4.2.3
instanceLabel	Only applicable for multi-instance templates. Not applicable for regular templates. Nokia recommends the use of regular templates. To distinguish multi-instance templates from regular templates, see section 4.30.	Section 4.2.5
state	The current state of the template; for example, RELEASED	—

4.23 getConfiguredTemplates operation

The getConfiguredTemplates operation is a service inventory operation that provides information about the templates that were used to configure an object using the 5529 APC.

The template name may indicate a composite template. The list of templates may refer to any valid combination of templates instances that were configured for the object.

The getConfiguredTemplates operation also provides the template arguments and associated values. The instanceLabel for a template may be provided, if applicable. The suspend state of the service template and service components in a locked state may be provided (suspendstate parameter), if applicable. The operation response does not include the suspend state for basic templates in a stack that are defined as locking components.

Table 44 describes the parameter for the getConfiguredTemplates operation request.

Table 44 getConfiguredTemplates operation request parameter

Parameter	Description	See
objectName	The name of the port for the operation or the name of the object created on top of the port	5529 APC Template Reference for identification information for objects and port types

Table 45 describes the parameters for the getConfiguredTemplates operation response.

Table 45 getConfiguredTemplates operation response parameters

Parameter	Description	See
objectName	The name of the port for the operation or the name of the object created on top of the port	5529 APC Template Reference for identification information for objects and port types
templateName	The name of the template that is configured for the port according to the 5529 APC	Section 4.2.2
type	The type of template	—
version	Version of the template	—
instanceLabel	Only applicable for multi-instance templates; not applicable for regular templates. Nokia recommends the use of regular templates. To distinguish multi-instance templates from regular templates, see section 4.30.	Section 4.2.5
state	The current state of the template	—
suspendstate	Indicates whether service template is in a suspended state; the states are SUSPENDED or ACTIVE	—
argument	Container element for the template argument	Section 4.2.4
name	Specifies the argument name	—
value	Specifies the argument value	—

(1 of 2)

Parameter	Description	See
serviceComponents	Container element for the serviceComponent, name, and suspendstate parameters. The serviceComponents parameter represents a list of service components retrieved by the getConfiguredTemplates operation. The parameter can include as many service components as required, or contain no service components.	—
serviceComponent	Container element for the name and suspendstate parameters	—
name	Specifies the name of the service component (basic template)	—
suspendstate	Indicates whether the service component is in a suspended state	—

(2 of 2)

4.24 getLogicalPorts operation

The getLogicalPorts operation is a service inventory operation that provides a list of all of the logical ports specified by the objectName parameter. The results provided by this operation are based on the format of the objectName parameter; for example, if an LT is provided, all of the logical ports on the LT are returned.

The getLogicalPorts operation uses the objectName to query the 5529 APC. Nokia recommends that you perform the getLogicalPorts operation only on the following objects:

- LT or lower
- PON or lower

The getLogicalPorts operation performs a prefix string match rather than a location match. For example, if you provide the objectName parameter as *NE:1-1-1* (*NE:rack-subrack-slot*), the operation returns the logical ports for all of the objects that start with a number one instead of just for the *NE:1-1-1* object: *NE:1-1-1*, *NE:1-1-10*, *NE:1-1-11*, *NE:1-1-12*, *NE:1-1-13*, *NE:1-1-14*, and so on. To ensure that the operation returns all of the logical ports for the intended object, you need to provide a dash (-) at the end of the objectName string; for example, *NE:1-1-1-*.

In the case of PON-level logical ports, if you need to retrieve all of the logical ports on the ONT 1 of PON port 1, for example *NE:1-1-1-1* (*NE:rack-subrack-slot-PON_port-ONT*), you need to provide the objectName as follows with the dash included at the end of the string: *NE:1-1-1-1-*.

Table 46 describes the parameter for the getLogicalPorts operation request.

Table 46 getLogicalPorts operation request parameter

Parameter	Description	See
objectName	The name of the port for the operation or the name of the object created on top of the port	<i>5529 APC Template Reference</i> for identification information for objects and port types

Table 47 describes the parameter for the getLogicalPorts operation response. See chapter 7.2 for an example of a getLogicalPorts operation response message.

Table 47 getLogicalPorts operation response parameter

Parameter	Description	See
logicalPort	Displays the logical port of the specified port or object	5529 APC Template Reference for identification information for objects and port types

4.25 getNodeAuditResult operation

The getNodeAuditResult operation is an audit and reconciliation operation that allows the client application to request the status of a previously executed auditNode requests. You need to keep running the getNodeAuditResult operation until the state parameter in the response message displays a READY value, which indicates that the auditNode operation was executed.

Table 48 describes the parameters for the getNodeAuditResult operation request.

Table 48 getNodeAuditResult operation request parameters

Parameter	Description
requestId	The identifier of the auditNode request for which the client is requesting information

Table 49 describes the parameters for the getNodeAuditResult operation response. See section 7.2.6 for an example of a getNodeAuditResult operation response message.

Table 49 getNodeAuditResult operation response parameters

Parameter	Description	See
state	The current state of the auditNode operation: <ul style="list-style-type: none">• READY — auditNode was performed• PLANNED — auditNode was not performed	—
executionDate	Displays when the state is READY. The date and time that the auditNode was performed	—
success	Displays when the state is READY. Displays whether the command performed successfully (True) or unsuccessfully (False)	—
exceptionErrorCode ⁽¹⁾	Displays when the state is READY and the success parameter is False. The exceptionErrorCode displays the error code.	Table 12
exceptionMessage ⁽¹⁾	Displays when the state is READY and the success parameter is False. The exceptionMessage displays the message for errorCode.	Table 12

(1 of 2)

Parameter	Description	See
portResult	Displays the port identifier	—
auditResult	Displays the error code, when one is generated	Table 12
serviceTemplateId ⁽¹⁾	Container element for the templateName, templateVersion, and instanceLabel parameters for each auditResult parameter	—
templateName ⁽¹⁾	The name of the template that is configured for the port according to the 5529 APC	Section 4.2.2
templateVersion ⁽¹⁾	Template version	Section 4.2.3
instanceLabel ⁽¹⁾	Only applicable for multi-instance templates; not applicable for regular templates. Nokia recommends the use of regular templates. To distinguish multi-instance templates from regular templates, see section 4.30.	Section 4.2.5
auditReport	Container for the templateName, instanceLabel, templateType, and configured parameters for each auditResult parameter	Table 50

(2 of 2)

Note

⁽¹⁾ These parameters are provided for each port in the operation response.

Table 50 describes the information that is returned for each audited port on the NE when the returned value for the state parameter is READY and the returned value for the success parameter is True. The audit for each port in the NE is equivalent to the auditPort operation for individual ports. See section 7.2.1 for an example of the getNodeAuditResult information.

Table 50 **getNodeAuditResult information**

Parameter	Description	See
templateName	The name of the template that is configured for the port according to the 5529 APC	Section 4.2.2
instanceLabel	Only applicable for multi-instance templates; not applicable for regular templates. Nokia recommends the use of regular templates. To distinguish multi-instance templates from regular templates, see section 4.30.	Section 4.2.5
templateType	The type of template for the audit	—
configured	Indicates whether templateName is configured for the port according to the NE. The values are true and false.	—
remoteFailedAssertion	Indicates each configuration difference between the 5529 APC and the NE. A remoteFailedAssertion item is displayed for each difference and indicates the expected and actual parameter values. The parameter is a container element for the property, expected, and actual parameters.	—
property	Container element for the 5529 APC attribute name	—
expected	The expected value in the NE	—
actual	The actual value in the NE	—

4.26 getResult operation

The getResult operation is a bulk execution operation that allows the client application to request the status of a command that was part of a previously executed web service. Table 51 describes the parameters for the getResult operation request.

Table 51 getResult operation request parameters

Parameter	Description
requestId	The bulk request identifier that determines the commands for the client application
commandId	The command identifier is an integer that is provided by the client application to identify the command in a bulk request. The value must be unique for each command in the bulk request.

Table 52 describes the getResult operation response parameters.

Table 52 getResult and getResult operations response parameters

Parameter	Description	See
requestId	The bulk request identifier value	—
commandId	The command identifier value	—
state	<p>The current state of the command:</p> <ul style="list-style-type: none"> READY <ul style="list-style-type: none"> command succeeded command failed with no automatic retry command failed due to other reasons than Timeout error or Failed to acquire transaction, while automatic retry still pending due to another command command not executed and action stopped executing, with no outstanding automatic retry PLANNED <ul style="list-style-type: none"> command not yet executed and action is executing action stopped executing, but there is at least one outstanding automatic retry command qualifies for retry and this automatic retry is still outstanding 	—
executionDate	<p>Displays when the state is READY and the date and time that the command was performed.</p> <p>If the command has not executed and the action stopped executing, with no outstanding automatic retry, the executionDate is set to the time of the job completion. See Notes in exceptionErrorCode and exceptionMessage.</p>	—
success	Displays when the state is READY. Displays whether the command performed successfully (True) or unsuccessfully (False)	—
exceptionErrorCode	<p>Displays when the state is READY and the success parameter is FALSE. The exceptionErrorCode displays the error code.</p> <p>Note: If the command has not executed and the action stopped executing, with no outstanding automatic retry, the exceptionErrorCode is "TIMEOUT_ERROR".</p>	Table 12
exceptionMessage	<p>Displays when the state is READY and the success parameter is FALSE. The exceptionMessage displays the message for errorCode. The message is free-form text and only for display.</p> <p>Note: If the command has not executed and the action stopped executing, with no outstanding automatic retry, the exceptionErrorMessage is "Command not executed due to max retries reached during bulk execution."</p>	Table 12

4.27 getResults operation

The getResults operation is a bulk execution operation that allows the client application to retrieve the status of the commands that have been performed in the sequence specified by the execute web service. Table 53 describes the parameter for the getResults operation request. Table 52 describes the response parameters.

Table 53 getResults operation request parameters

Parameter	Description
requestId	The bulk request identifier that determines the commands for the client application and identifies all of the commands in the bulk request

4.28 getSystemHealthInfo operation

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

The getSystemHealthInfo operation does not specify any parameters in the request.

An operation response that is successful returns the NBI server URI; for example, uri://alcatel.com/apc/9.6. If the server is unavailable, the getSystemHealthInfo operation returns an error message. For information about how to add a parameter to the response to explicitly state that the operation was a success, see section 4.2.7.

4.29 getTemplate operation

The getTemplate operation is a service inventory operation that provides information about a specific template that is being managed by the 5529 APC. The template may or may not have been used to configure ports. The getTemplate operation provides the following information:

- template details, including name, type, version, and state
- argument
- associated values

The getTemplate operation cannot retrieve templates that are in draft state.

Table 54 describes the parameters for the getTemplate operation request.

Table 54 **getTemplate operation request parameters**

Parameter	Description	See
templateName	The name of the template for the operation	Section 4.2.2
templateVersion	Version of the template	Section 4.2.3

Table [55](#) describes the getTemplate operation response parameters.

Table 55 **getTemplate operation response parameters**

Parameter	Description	See
templateName	The name of the template for the operation	Section 4.2.2
type	The type of template	—
version	Version of the template	—
state	The current state of the template (RELEASED or PHASED-OUT)	—
instanceLabelUsePermitted	Indicates whether the template designer allows an instance label to be used with the template. The permission is not enforced by the 5529 APC. If an instance label is not allowed, the 5529 APC SPFE users who can configure service templates in the 5520 AMS cannot configure the instance label. The OSS operates in a similar way.	Contact your template designer to design templates that are intended to be used with instance label. See the <i>5529 APC Installation, Administration, and User Guide</i> for the information about how to configure an instance label in the SPFE GUI.
instanceLabelNamingRule	Indicates whether the template designer enforces a naming rule for the instance labels that can be used in combination with a template. A naming rule ensures that OSS users and SPFE users enter instance labels according to a defined character pattern. The template designer can change the rules; however, the template designer must ensure that a new naming rule is acceptable for OSS clients and SPFE users, and conforms with previously used instance labels for the same template.	Contact your template designer to define multi-instance labels and multi-instance naming rules.
argument	Container element for the template argument	Section 4.2.4
name	Specifies the argument name	
value	Specifies the argument value	

4.30 getTemplateMetaData operation

The getTemplateMetaData operation is a service inventory operation that provides metadata about a specific template that is managed by the 5529 APC. The operation provides the OSS client with the means to know what is required to configure a selected template. By using the template metadata information, the OSS client is less dependent on predefined information about the template. As a result, the getTemplateMetaData operation enables a further step in the dynamic use of templates as service components. The template may or may not have been used to configure ports. The getTemplateMetaData operation provides the following information:

- template details, including name, type, version, and state
- attribute metadata, including:
 - name
 - properties, including mode (ARGUMENT or PARAMETER) and type (NATIVE or VIRTUAL)
 - type
 - default, valid, minimum, and maximum values
 - whether an argument is mandatory or optional
 - whether an argument can be modified
 - whether an attribute is an instance argument
 - SPFE alias



Note — Only the ARGUMENT mode is supported in this 5529 APC release. Currently, the template metadata provides access to the template argument details for the OSS clients that are retrieving template metadata.

The getTemplateMetaData operation can be used to retrieve a list of serial numbers of ONTs discovered on a corresponding PON port, or a list of MAC addresses discovered on a corresponding EPON port. The ONT template must have the Serial Number or MAC Address attribute configured as an argument with the following filters provided for the operation:

- ontIdentifierSourceType, where the value is PON
- ontIdentifierSourceAddress where the value is an object name of the PON port, for example, *rack-subrack-slot-port*



Note — The list of serial numbers or MAC addresses contains the serial numbers or MAC addresses on discovered ONTs, in alphabetical order.

Table [56](#) describes parameters for the getTemplateMetaData operation request.

Table 56 **getTemplateMetaData operation request parameters**

Parameter	Description	See
templateName	The name of the template for the operation	Section 4.2.2
templateVersion	Version of the template	Section 4.2.3
filter	Container element for the argumentList parameter	—
argumentList	Container element for the list of template arguments	Section 4.2.4 and Table 57
nameAndStringValue	Container element for the name and value parameters	
name	Specifies the argument name	
value	Specifies the argument value	

Table 57 describes the arguments for the argumentList parameter.

Table 57 **argumentList parameter description**

Argument name	Description
profileSourceType	Type of source from which template metadata is retrieved; for example, NE or system. The only value that is supported in this 5529 APC release is node.
profileSourceName	Name of the source from which template metadata is retrieved. When the profileSourceType argument value is node, the value for the profileSourceName argument is the name of the NE; for example, ISAM_Brussels_47.

Table 58 describes the getTemplateMetaData operation response parameters. See chapter 7 for an example of a getTemplateMetaData operation response message.

Table 58 **getTemplateMetaData operation response parameters**

Parameter	Description	See
templateName	The name of the template for the operation	Section 4.2.2
type	The type of template	Section 4.2.3
version	Version of the template	—
state	The current state of the template (RELEASED or PHASED-OUT)	—
instanceLabelUse Permitted	Indicates whether the template designer allows an instance label to be used with the template. The permission is not enforced by the 5529 APC. If an instance label is not allowed, the 5529 APC SPFE users who can configure service templates in the 5520 AMS cannot configure the instance label. The OSS operates in a similar way.	Contact your template designer to design templates that are intended to be used with instance labels. See the <i>5529 APC Installation, Administration, and User Guide</i> for information about how to configure an instance label in the SPFE GUI.

(1 of 2)

Parameter	Description	See
instanceLabelNamingRule	Indicates whether the template designer enforces a naming rule for the instance labels that can be used in combination with a template. A naming rule ensures that OSS users and SPFE users enter instance labels according to a defined character pattern. The template designer can change the rules; however, the template designer must ensure that a new naming rule is acceptable for OSS clients and SPFE users, and conforms with previously used instance labels for the same template.	Contact your template designer to define multi-instance labels and multi-instance naming rules.
attributesList	Container element for the list of arguments for the selected template. If there are no template arguments, the list is empty. This parameter is optional.	—
attributeMetaData	The metadata for the template arguments, such as name, data type, default value, and SPFE alias	Table 59
attributeGroupMetaData	This attribute is not currently supported	—

(2 of 2)

Table 59 describes the attributeMetaData metadata.

Table 59 attributeMetaData metadata description

Name	Description
attributeName	Name of a native or virtual argument
attributeProperties	<p>Properties of the argument:</p> <ul style="list-style-type: none"> mode (ARGUMENT or PARAMETER) nature (NATIVE or VIRTUAL) <p>Only the ARGUMENT mode is supported in this 5529 APC release.</p> <p>The mode and nature values are separated by the \r escape character. The format of the attributeProperties value is <i>mode\rnature</i>, for example, ARGUMENT\rNATIVE.</p>
attributeDataType	Data type of a native argument (INTEGER, LONG, STRING, ENUM, BOOLEAN, DOUBLE, or SELECT_LIST) or a virtual argument (STRING or ENUM)
attributeDefaultValue	<p>Default value of a native or virtual argument.</p> <p>The default value of a native argument is available from the 5529 APC and is not the default value from the NE. The default value of a virtual argument is provided by the user. If a value is not provided by the user for native or virtual arguments, the value is empty.</p> <p>This metadata argument is optional.</p>
attributeValidValues	<p>Values for a native or virtual argument.</p> <p>For a native argument, the data type must be ENUM or SELECT_LIST. When profile arguments are included in the list of template arguments that are returned by the operation, and the values that are provided for the argumentList parameter in the operation request are valid, the response includes a list of available profiles from the NE that is specified in the profileSourceName argument.</p> <p>For a virtual argument, the data type must be ENUM.</p> <p>This metadata argument is optional.</p>
attributeRequired	<p>Indicates whether a native or virtual argument is MANDATORY or OPTIONAL.</p> <p>When one native argument is mandatory, then the virtual arguments are also mandatory.</p>

(1 of 2)

Name	Description
attributeModifiable	Indicates whether a native or virtual argument can be modified (YES or NO). If one native argument can be modified, then the virtual arguments can also be modified.
attributeSpfeAlias	The user-friendly name for a native or virtual argument that is displayed in the SPFE GUI. Allows the use of the same attribute name (alias) in different applications; for example, 5529 APC SPFE and OSS client. For a native argument, the user-friendly name of the argument is returned. For a virtual argument, the user-friendly name of the argument is provided by the user in the attribute properties using the 5529 APC GUI. See the <i>5529 APC Installation, Administration, and User Guide</i> for more information about configuring the SPFE alias for a virtual argument.
attributeMinValue	Minimum value for a native argument. The data type must be INTEGER or LONG.
attributeMaxValue	Maximum value for a native argument. The data type must be INTEGER or LONG.
attributeLength	Maximum length of a string value. For a native argument, the data type must be STRING. For a virtual argument, the data type must be STRING. The default string length is 255 characters.
attributeGroupName	Indicates the name of the attribute group to which the attribute belongs. If the value of group name for an argument is not specified during template design, the attributeGroupName tag will not appear in the response of getTemplateMetadata operation.
attributeIsInstanceArgument	Indicates whether the attribute is an instance argument. The value is YES or NO.
attributeExcludeInformation	Indicates that the attribute is excluded from audit, reapply, or discover operations. This property only displays when an attribute is excluded from at least one of the operations previously mentioned. This property is not displayed in the response for attributes that are included in audit, reapply, and discover operations. The values are excludeFromAudit, excludeFromReapply, and excludeFromDiscover.
attributeFormula	Indicates the formula name which would be used to auto-generate the value of the argument. If the value of formula for an argument is not specified during template design, the attributeFormula tag will not appear in the response of getTemplateMetadata operation.
attributePattern	Indicates that if the argument is a string type, it should comply to a compiled representation of a regular expression. If no value for attribute pattern is set, the attributePattern tag will not appear in the response of getTemplateMetadata operation.
attributeWebInvisibleResolvable	If the value is set to YES, the ONT Serial Number or MAC Address argument is an enum type, and the attributeValidValues is set as available values of the ENUM. The getTemplateMetadata operation response includes in its attributeValidValues, a list of serial numbers of discovered ONTs in the case of GPON ONTs or a list of MAC Addresses of discovered ONTs in the case of EPON ONTs.

(2 of 2)

4.30.1 \r character in operation response

Prior to 5529 APC Release 9.6, in the getTemplateMetaData operation response, the \r character (carriage return) is sent as \r. The OSS client receives the \r character in the response.

In 5529 APC Release 9.6 or later, following the upgrade to an AXIS2 library, the `\r` character is encoded as `#xd;`. The `#xd;` character string follows the XML standard, so the OSS client is expected to support the string and decode `#xd;` to `\r` when it processes a SOAP XML response.

If the OSS client does not follow the XML standard, the `getTemplateMetaData` operation response will display the `\r` character as `#xd;` in 5529 APC Release 9.6 or later. In the operation response, if you have multiple values for the value element in a `nameAndStringValue` element, the values will not be separated on different lines to indicate multiple values. The values will display as a single string. For example:

OSS client supports XML standard in Release 9.6 or later:

```
<nameAndStringValue>
  <name>attributeValidValues</name>
  <value>ALCLF8A88326
ALCLF9B58720</value>
</nameAndStringValue>
```

OSS client does not support XML standard in Release 9.6 or later:

```
<nameAndStringValue>
  <name>attributeValidValues</name>
  <value>ALCLF8A88326ALCLF9B58720</value>
</nameAndStringValue>
```

4.31 getTemplateNames operation

The `getTemplateNames` operation is a service inventory operation that provides the list of names for all templates that are known by the 5529 APC for service provisioning. Template names are typically used when you configure ports or service connections using the 5529 APC Northbound interface. The `getTemplateNames` operation request does not have any parameters.

Table 60 describes the `getTemplateNames` operation response parameters.



Note — Table 60 lists only the elements that are used by the `getTemplateNames` operation. There may be additional optional elements that are displayed in the WSDL file (for example, `instanceLabel`), but those elements are not used by the `getTemplateNames` operation.

Table 60 `getTemplateNames` operation response parameters

Parameter	Description	See
templateId	Container element for the name, version, and state parameters if applicable.	—
name	The name of the template	Section 4.2.2
version	The version of the template	Section 4.2.3

(1 of 2)

Parameter	Description	See
state	The current state of the template; for example, RELEASED	—

(2 of 2)

4.32 importTemplate operation

The importTemplate operation is an audit and reconciliation operation that allows the OSS client application to populate the 5529 APC with the target configuration for a port, as intended by the OSS. This web service allows the OSS/BSS client application to inform the 5529 APC about the port configuration as it should or may exist in the NE, without any attempt by the 5529 APC to configure it in the NE. If needed, this operation can be applied multiple times with different templates to get a complete port configuration populated in the 5529 APC.

The importTemplate operation can be used when a port is configured by a management application other than the 5529 APC and you need to align the 5529 APC with the configuration. This web service is intended to be used at the initial introduction of the 5529 APC system in a network that is in operation.

Table 61 describes the parameters for the importTemplate operation request. By default, there are no parameters for the response unless there is an error. For information about how to add a parameter to the response to explicitly state that the operation was a success, see section 4.2.7.

Table 61 importTemplate operation request parameters

Parameter	Description	See
objectName	The name of the port for the operation or the name of the object created on top of the port	<i>5529 APC Template Reference</i> for identification information for objects and port types
templateName	The name of the template for the operation	Section 4.2.2
templateVersion	Template version	Section 4.2.3
instanceLabel	Only applicable for multi-instance templates. Not applicable for regular templates. Nokia recommends the use of regular templates. To distinguish multi-instance templates from regular templates, see section 4.30.	Section 4.2.5
state	Container element for the suspend state of the service template and its service components. The suspend state is also supported for bulk importTemplate operations. This parameter is optional.	—
name	Specifies the state name	—
value	Specifies the state value	—

(1 of 2)

Parameter	Description	See
argument	Container element for the template argument	Section 4.2.4
name	Specifies the argument name	
value	Specifies the argument value	
operationInitiator	The system and user ID of the user who is initiating the change to the port or object. This parameter is optional.	Section 4.2.6

(2 of 2)

4.33 migrate operation

The migrate operation is a service provisioning operation that you can use to perform the following tasks:

- Move from one template version to another version—the template names and instance labels are the same, while the template version and the arguments can be different. Use the migrate operation to change from one service implementation to another service implementation.
- Change the template—the template names and the arguments can be different. Use the migrate operation to change from one service for a port to another service.



Note 1 — The migrate operation is not supported at the shared port level in a high subscriber density configuration; see section [3.6](#) for information about the 5529 APC and high subscriber density configurations.

Note 2 — The migrate operation will take longer than three to six seconds to process template stacks that contain the following network infrastructure services: SAP, VLAN, and VVPLS. Their respective 5529 APC infrastructure service templates are: ISAM SAP, ISAM VLAN for IHUB-Based ISAM, and ISAM VVPLS.

Table [62](#) describes the parameters for the migrate operation request. The object for migration is identified by the values of the objectName, oldTemplate, oldTemplateVersion, oldInstanceLabel, and oldArgument parameters. By default, there are no parameters for the response unless there is an error. For information about how to add a parameter to the response to explicitly state that the operation was a success, see section [4.2.7](#).

Table 62 migrate operation request parameters

Parameter	Description	See
objectName	The name of the port for the operation	<i>5529 APC Template Reference</i> for identification information for objects and port types
oldTemplateName	The name of the template that is currently applied to the port	Section 4.2.2
oldTemplateVersion	The version of the template that is currently applied to the port	Section 4.2.3
oldTemplateInstanceLabel	Label that identifies the template instance that is currently applied to the port	Section 4.2.5
oldArgument	Container element for the template arguments of the template that is currently applied to the port	Section 4.2.4
name	Specifies the argument name	
value	Specifies the argument value	
newTemplateName	The name of the template to be applied on the port instead of oldTemplateName	Section 4.2.2
newTemplateVersion	The version for the newTemplateName template	Section 4.2.3
newTemplateInstanceLabel	The label that identifies the template instance for the newTemplateName template	Section 4.2.5
newArgument	Container element for the template arguments for the newTemplateName template	Section 4.2.4
name	Specifies the argument name	
value	Specifies the argument value	
migrationStrategy	The bulk configuration execute operation supports the use of the migrationStrategy parameter for the migrate operation. This parameter is optional.	Section 4.33.1
operationInitiator	The system and user ID of the user who is initiating the change to the port or object. This parameter is optional.	Section 4.2.6

The following migrate operation request parameters are described in more detail:

- migrationStrategy (see section [4.33.1](#))

Additional information about the following is provided in the rest of this section:

- retain configured values (see section [4.33.2](#))
- migrating service stacks (see section [4.33.3](#))
- smart migrate (see section [4.33.4](#))
- migrate with NO_REBUILD and -No-Value- as the optional attribute default value (see section [4.33.5](#))
- maintain the administrative state of the service template during migration (see section [4.33.6](#))

- migrating a template with arguments that depend on a specific argument value being set (see section 4.33.7)
- rollback (see section 4.33.8)

4.33.1 migrationStrategy parameter

The migrate operation uses an optional migrationStrategy parameter that can be set by the operator. The selected migrationStrategy option affects how the migrate operation is executed. Table 63 describes the migrationStrategy parameter options.

Table 63 migrationStrategy parameter description

migrationStrategy option	Migration behavior	Rollback behavior
NO_REBUILD	The operation attempts to migrate without unconfiguring and reconfiguring. If the unconfiguring of the old service template is required to achieve the migration, the system rejects the operation request.	The operation unconfigures any templates added to the initial template stack, and restores any modified attributes to their initial values.
REBUILD_IF_NEEDED (default)	The operation attempts to perform the migration without unconfiguring the old service template. If the attempt fails, then the operation continues by unconfiguring and configuring to achieve the migration.	If the operation was attempting to follow a NO_REBUILD strategy when an error occurred, the operation then attempts to perform a rollback defined for the NO_REBUILD migration strategy. If the NO_REBUILD strategy is not possible and the operation arrives at an error, the operation unconfigures the new template stack and reconfigures the initial template stack.
FORCE_REBUILD	The operation performs the migration by unconfiguring the old service template and configuring the new service template.	The operation unconfigures the new template, and reconfigures the initial template.

When the migrate operation uses the strategy REBUILD_IF_NEEDED or FORCE_REBUILD, and there is a non-modifiable argument that is supported by AVG in the template, send the value for the non-modifiable argument in the NBI request.

4.33.2 Retain configured values

You can configure a template migration request in which the migration retains the configured argument values. To retain a configured argument value, specify the APC_RETAIN_CONFIGURATION value in the newArgument parameter. For example:

```
<newArgument>
  <name>ontSerialNumber</name>
  <value>APC_RETAIN_CONFIGURATION</value>
</newArgument>
```


If an argument was not present in the original template, APC_RETAIN_CONFIGURATION cannot be used. An INVALID_ARGUMENT error message is returned in the NBI response and added to the apc.log.

If a virtual argument with the same name, but different mappings, is used in an old argument and a new argument with the APC_RETAIN_CONFIGURATION value, the 5529 APC uses the previously configured value, even if doing so would result in different values of mapped arguments. See the *5529 APC Installation, Administration, and User Guide* for more information about virtual arguments and mapped values.

The actual configured value for all new arguments with the APC_RETAIN_CONFIGURATION value is logged in the apc-audit.log; see the *5529 APC Installation, Administration, and User Guide* for more information about logs.

4.33.3 Migrating service stacks

The 5529 APC supports the migration from a service stack (A) to another service stack (B) without unconfiguring the configuration in the NE port if the stacks meet some defined conditions. These conditions depend on the differences between the two service template stacks. For information about the differences allowed between the two service template stacks, see the *5529 APC Installation, Administration, and User Guide*.

The 5529 APC migrate operation also supports the migration of a service template stack in a multi-layer stack of services. The service stack to be migrated can be at the top, in the middle, or at the bottom of a configured multi-layer stack. The migration takes place if there are no dependency constraints that prevent the operation from unconfiguring and reconfiguring the services.

The 5529 APC attempts to perform the migration of a service template stack in a multi-layer stack of services only if the operator set the migrationStrategy parameter to FORCE_REBUILD or REBUILD_IF_NEEDED, and an unconfigure is needed.

4.33.4 Smart migrate

Smart migration is migration of a service without removing the original service. Smart migration is done with the same template and can be done with either of the following:

- with modification of modifiable arguments (modify only)
- with added new parts and modification of modifiable arguments (modify and add)

Smart migration requires that the migrationStrategy parameter (described in section [4.33.1](#)) is NO_REBUILD or REBUILD_IF_NEEDED.

Smart migration is not supported for every combination of original and final template and is based on compatible structure of the two templates. Smart migration done with modification of modifiable arguments (modify only) usually has a compatible structure between the original and final templates. Another case where smart migration is supported is the addition of new templates on top of the old one (basic template or a stack).

Nokia recommends that you use smart migrate on unlocked services only. Using smart migrate on a locked service may result in an audit misalignment. For more information, contact your Nokia technical support representative.

4.33.5 migrate with NO_REBUILD and -No-Value- as the optional attribute default value



Note — The information in this section does not apply to 7302 ISAM/7330 ISAM FTTN R3.7/FGN3.8.

When a migrate operation with NO_REBUILD or a rollback for a migrate operation with NO_REBUILD occurs, and -No-Value- is the optional attribute default value, the 5529 APC uses the NE default value as the expected attribute value. Some optional attributes do not have an NE default value defined, and those attributes are not configured. See the *5529 APC Template Reference* for an NE for information about optional attributes that do not have an NE default value defined.

4.33.6 Maintain administrative state during service migration

When migrating a service, the 5529 APC maintains the administrative state of the service template so that the administrative state of the new service is the same as the old service.

Retention of the administrative state during service migration is supported as much as possible by the 5529 APC for service components. However, retention of the administrative state during service migration cannot be supported if the structure of the template is changed significantly as a result of migration, for example, a change in defined locking or service components. In this case Nokia recommends that services are unlocked before migration.



Note 1 — The administrative state may not be migrated correctly during a service migration if the template structure has been changed significantly, for example, a change of explicitly defined locking component, removal of service components, and so on. Retention of the administrative state of service components is not guaranteed to work correctly if service component definitions are changed in migration, specifically if service components are removed or not available in the final template. Nokia recommends that you always unlock the service and service components before migration if locking and/or service component definition will be changed in the migration.

Note 2 — Retention of the administrative state of bind service is not guaranteed to work correctly if locking component definition is changed in migration. Nokia recommends that you always unlock bind template before migration if locking and/or component definition will be changed in the migration.

Note 3 — Although referred to as being maintained or retained during migration, the template is actually migrated in the unlocked state then locked after migration according to the state of the template before migration.

Table 64 provides additional information about the administrative state for various service migration scenarios.

Table 64 **Service migration scenarios and administrative state of service templates**

Service migration scenario	Administrative state of service template
Smart migration (modify only)	It is advised to unlock the templates before migration to maintain a consistent administrative state during migration from the default locking component ⁽¹⁾ to an explicit locking component definition for customers moving from the deprecated administrative state management using named basic templates. Otherwise migration from the default locking component to an explicitly defined locking component even with the same template can result in an inconsistent state of template.
Smart migration (modify and add)	Retention of the administrative state of service is not guaranteed to work correctly if locking component definition (that includes a change of defined or default locking component ⁽¹⁾) is changed in migration. Recommendation is to always unlock template before migration if there is a change in the locking component definition.
Migration with reconfiguration	Retention of the administrative state of service is not guaranteed to work correctly if locking component definition (that includes a change of defined or default locking component ⁽¹⁾) is changed in migration. The administrative state is migrated in the unlocked state then locked after migration according to the state of the template before migration.

Note

⁽¹⁾ The default locking component is the top right-most lockable template in a stack.

4.33.7 Migrating a template with arguments that depend on a specific argument value being set

There are template arguments that are applicable only when another argument value is set to a specific value. When the argument value is changed from one value to another, the dependent arguments of the first value become not applicable and the dependent arguments of the second value become applicable. When you migrate these types of templates, you need to be aware of how the arguments are managed in the 5529 APC and the NE. Nokia recommends that you set the values of the dependent arguments that will become not applicable to -No-Value- when you attempt to change the value of the main argument.

Alternatively, you can use one of the following operations to manage these types of arguments:

- modify operation with the forceModify parameter set to True
- migrate operation with the migrationStrategy parameter set to FORCE_REBUILD

4.33.8 Rollback

The 5529 APC attempts to perform a rollback if the migrate operation fails to complete in any supported migration scenario.

4.34 modify operation

The modify operation is a Service provisioning operation that changes the service for the port that is specified by the objectName parameter or an object created on top of the physical port. The object to be modified is identified by the objectName parameter, templateName parameter, and template arguments. Only template attributes that are defined as arguments when the template is created can be modified due to the functionality of the attribute or the NE implementation. Some attributes that are configured as arguments cannot be modified. See the *5529 APC Template Reference* for information about the parameters that can be modified. The modify operation can be prevented by the NE depending on the objects on top of the port or if the port does not support the requested modification.



Caution — The modify operation may cause a service interruption.

The 5529 APC does not support modifying the value of a mandatory attribute to be -No-Value-.

There are differences between the migrate and modify operations. For the modify operation, you can only change the template arguments. For the migrate operation, you can change the template of a port to another version of the template or to another template.

Table 65 describes the parameters for the modify operation request. By default, there are no parameters for the response unless there is an error. For information about how to add a parameter to the response to explicitly state that the operation was a success, see section 4.2.7.

The modify operation behaves atomically. If there is an error, a rollback is performed for the port by the 5529 APC. Argument value range checking is delegated by the 5529 APC to the NE.

When the modify operation has the forceModify parameter set to True on a template stack and the reconfiguration fails, but a successful rollback occurs, a NON_MODIFIABLE_ARGUMENT error code may be generated. The reconfiguration failure and successful rollback are the correct operation results, so the error code can be disregarded.

Table 65 **modify operation request parameters**

Parameter	Description	See
objectName	The name of the port or object for the operation	<i>5529 APC Template Reference</i> for identification information for objects and port types
templateName	The name of the template for the operation	Section 4.2.2
templateVersion	The version of the template for the operation. This parameter is optional.	Section 4.2.3
instanceLabel	Only applicable for multi-instance templates. Not applicable for regular templates. Nokia recommends the use of regular templates. To distinguish multi-instance templates from regular templates, see section 4.30.	Section 4.2.5
argument	Container element for the argument	Section 4.2.4
name	Specifies the argument name	
value	Specifies the argument value	
forceModify	Specifies whether to force a non-modifiable attribute to be modified	Section 4.34.1
operationInitiator	The system and user IDs of the user who is initiating the change on the port or object. This parameter is optional.	Section 4.2.6

4.34.1 forceModify

To modify arguments that are not modifiable, the modify operation has an optional forceModify parameter. Without the forceModify optional parameter, the impacted template and the related client templates must be deleted and the template with the appropriate argument values and the associated client templates must be recreated.

The forceModify parameter is used to modify attributes according to the following rules:

- Attributes for which NEs support direct modification can be modified using the modify operation without using the forceModify parameter; do not provide the forceModify parameter in the request, or set the parameter to False.
Due to the NE behavior, there is a minimal set of attributes that can cause service interruption when the NE is modified.
- Attributes for which NEs do not support direct modification can be modified using the modify operation and setting the forceModify parameter to True in the request.
When a modify request with the forceModify parameter is set to True and a non-directly-modifiable attribute is requested to be modified, the 5529 APC performs the required unconfigure and configure operations, including a rollback, if required. This behavior can involve resource removal and creation. Therefore, there is usually service interruption during the execution of the modify operation for the services that are related to these resources.

The modify operation with the unconfigure and configure operations can be unsuccessful when:

- there are insufficient resources on the NE to re-establish client layer dependent objects or templates
- there is a compatibility problem with the new values with client layer dependent objects or templates

See the *5529 APC Template Reference* for information about whether the modification of an attribute requires the forceModify parameter to be set to True.

When the 5529 APC attempts to perform a modification, with the forceModify parameter, of arguments in a service template stack that is at the middle or lower end of a multi-layer service stack, the 5529 APC first unconfigures the upper layers of services, and then unconfigures and reconfigures the target service template stack with the new argument values before reconfiguring the upper layers of services.

After a successful modification, the change is taken into account by the 5529 APC for subsequent operations, such as the audit and getConfiguredTemplates operations.

4.34.2 modify operation without forceModify and with -No-Value- as the optional attribute default value



Note — The information in this section does not apply to 7302 ISAM/7330 ISAM FTTN R3.7/FGN3.8.

When a modify operation without the forceModify parameter or a rollback for a modify operation without the forceModify parameter occurs, and -No-Value- is the optional attribute default value, the 5529 APC uses the NE default value as the expected attribute value. Some optional attributes do not have an NE default value defined, and those attributes are not configured. See the *5529 APC Template Reference* for an NE for information about optional attributes that do not have an NE default value defined.

4.34.3 Using forceModify after excluding attributes from audit

When NE object attributes are excluded from an audit operation, the NE object attributes are audited for configurations that are compatible with the service templates applied to the object but not configured by the service templates. The attribute values might be modified outside of the 5529 APC, which causes the values in the NE to be different from the values stored by the 5529 APC. As a result, when the modify operation uses the forceModify function, the 5529 APC changes the attribute values in the NE to the values that are stored by the 5529 APC.

Only a service template designer with a role that contains the APC - Template Super User function can exclude attributes from audit. See the *5529 APC Installation, Administration, and User Guide* for information about user functions and roles and excluding attributes from audit.

4.34.4 modify operation and shared services

This section provides information about using the modify operation with shared services.

4.34.4.1 Modifying shared arguments

The forceModify parameter does not support modifying shared arguments. You can modify shared arguments according to the following rules:

- If the shared argument is modifiable, use the modify operation to modify the argument on any of the services that contain the argument.
- If the shared argument is non-modifiable, unconfigure the services that share the argument and then configure the services again with the new argument value.

4.34.4.2 Modifying a virtual argument mapped to an argument on a shared service

Modifying the value of a virtual argument mapped to an argument on a shared service on the same port results in a misalignment for that virtual argument in the other shared services, as recorded in the audit report.

For example:

- the service template stack contains the ONT, card, and port templates, and a virtual argument that is mapped to an argument in the ONT template
- in the ONT, two services (service 1 and service 2) are configured as shared

When the user modifies the virtual argument value in service 1, the virtual argument value in service 2 is not modified automatically. When an audit operation is performed, a misalignment is reported for the virtual argument value in service 2. The user needs to modify the virtual argument value in service 2 to avoid an audit misalignment.

4.35 modifyTemplateInstance operation

The modifyTemplateInstance operation is a service provisioning operation that modifies the instance label for a template instance after the template instance has been applied. This operation does not apply to clients that use regular templates. The operation only applies to clients that use multi-instance templates. If the new instance label is provided, the format must conform to the format that is regulated by the instance label naming rule for the template. This operation behaves atomically.

Contact your template designer to administrate the instance label naming rules.

Table 66 describes the parameters for the modifyTemplateInstance operation request. By default, there are no parameters for the response unless there is an error. For information about how to add a parameter to the response to explicitly state that the operation was a success, see section 4.2.7.

Table 66 modifyTemplateInstance operation request parameters

Parameter	Description	See
objectName	The name of the port or object for the operation	5529 APC Template Reference for identification information for objects and port types
templateName	The name of the template for the operation	Section 4.2.2
templateVersion	Version of the template. This parameter is optional.	Section 4.2.3
instanceLabel	The current instance label of the template for the operation	Section 4.2.5

(1 of 2)

Parameter	Description	See
argument	Container element for the instance label argument	Section 4.2.5
name	The argument name is always instanceLabel.	
value	The argument value is the new instance label.	
operationInitiator	The system and user IDs of the user who is initiating the change on the port or object. This parameter is optional.	Section 4.2.6

(2 of 2)

4.36 move operation

The move operation is a service provisioning operation that moves all templates for a port to another port. The move operation can be used when a subscriber moves to another location, which also requires a different port. The operation performs the new configuration on the destination port after the source port is unconfigured.

The move operation also moves multi-instance templates, if applicable.

You can move templates between NEs that have the same release number, such as R4.8. You cannot move templates between NEs that have different release numbers, such as R4.8 and R4.9. To move templates between NEs that have different release numbers, unconfigure the template on the source NE and configure it on the destination NE.

The move operation is not applicable to ports that are involved in IMA or are part of a bonding group.

For NGPON2 ONTs, the channel pair associated with a channel group and sub-channel group is unique, which means that one channel pair can be associated with only one channel group and sub-channel group. The move operation supports moving services within the same channel group and sub-channel group. The services from one NGPON2 ONT can be moved to another NGPON2 ONT successfully because the source port and target port have the same channel group and sub-channel group, and the same associated channel pair. The move operation does not support moving services from one channel group and sub-channel group to a different channel group and sub-channel group. The move operation does not provide the option to specify the channel pair.

If there are no configurations on the source port (fromPort) or there are configurations for the target port (toPort), an exception is received.

This operation behaves atomically. If there is an error, a rollback is performed for the port by the 5529 APC.



Note 1 — For the move operation to be successful for Ethernet LT, SHDSL, and xDSL ports, you first need to configure the Port Locking After unconfigure/clean system setting to True; see the *5529 APC Installation, Administration, and User Guide* for information about configuring the 5529 APC system settings.

Note 2 — The move operation is not supported in a high subscriber density configuration; see section 3.6 for information about the 5529 APC and high subscriber density configurations.

Table 67 describes the parameters for the move operation request. By default, there are no parameters for the response unless there is an error. For information about how to add a parameter to the response to explicitly state that the operation was a success, see section 4.2.7.

Table 67 move operation request parameters

Parameter	Description	See
fromPort	ObjectName of the source port	<i>5529 APC Template Reference</i> for identification information for objects and port types
toPort	ObjectName of the target port	
operationInitiator	The system and user ID of the user who is initiating the change to the port or object. This parameter is optional.	Section 4.2.3

4.36.1 move with -No-Value- as the optional attribute default value



Note — The information in this section does not apply to 7302 ISAM/7330 ISAM FTTN R3.7/FGN3.8.

When a move operation or a rollback of a move operation occurs, and -No-Value- is the optional attribute default value, the 5529 APC uses the NE default value as the expected attribute value. Some optional attributes do not have an NE default value defined, and those attributes are not configured. See the *5529 APC Template Reference* for an NE for information about optional attributes that do not have an NE default value defined.

4.37 reapplyAllServices operation

The reapplyAllServices operation is an audit and reconciliation operation that supports the reconciliation process for ports or ONTs from the NBI. The reapplyAllServices operation enables the 5529 APC to reconfigure all services as intended on the selected port or ONT; the 5529 APC keeps the intended configuration as service templates and arguments. The reapplyAllServices operation performs reconciliation tasks on all of the configured services on the object identified by the objectName. The 5529 APC supports the operation for the following objects: ONT (non-MDU), MDU port, and physical port. The operation does not change the assigned service templates and its argument values, but performs changes to the port or ONT configuration to align with the intended configuration stored in the 5529 APC.

The operation uses a reapplyStrategy parameter that can be set by the operator. The selected reapplyStrategy parameter affects the level of service impact. The operation supports the following options for the parameter:

- **MODIFY_ONLY** (default)
The operation reconciles the misalignment of modifiable attributes. The level of service interruption depends on the NE ability to handle attribute modifications. The operation attempts to align as many attributes as possible, and the outcome can be a complete or a partial success. Partial success means that some misalignments have been resolved, but there are still some misalignments that have not been resolved. The operation goal is to achieve a level of alignment that is as close as possible to the intended configuration for the object. This strategy does not allow full service alignment when misalignments are caused by missing objects and inExcess objects.
- **REBUILD_WHERE_NEEDED**
The operation first attempts to use a realignment method that affects the service the least (just by modifying attributes). If the simple modification is not sufficient (for example, when the misalignment involves non-modifiable attributes, or unexpected or missing managed objects), the operation forces the realignment by unconfiguring and reconfiguring the service.
- **FORCE_REBUILD**
The operation forces the reconfiguration of all intended services on the port or ONT as retained by the 5529 APC. As a result of the forced full reconfiguration, service interruption is certain. The purpose of this option is to reconfigure the service with the intended configuration retained by the 5529 APC, regardless of the audit result that was registered before running the operation (even though no misalignment was present).

As the intent of the `reapplyAllServices` operation is to restore the intended configuration as retained in the 5529 APC, this operation does not attempt to roll back the misaligned configuration on the port or ONT if an error occurs during processing.



Note 1 — The `reapplyAllServices` operation is not supported in a high subscriber density configuration; see section 3.6 for information about the 5529 APC and high subscriber density configurations.

Note 2 — The `reapplyAllServices` operation may indicate success for a non-existing port if the address pattern of the non-existing port is supported by the NE model and by the clean operation.

Note 3 — The `reapplyAllServices` operation will take longer than three to six seconds to process template stacks that contain the following network infrastructure services: SAP, VLAN, and VVPLS. Their respective 5529 APC infrastructure service templates are: ISAM SAP, ISAM VLAN for IHUB-Based ISAM, and ISAM VVPLS.

Table 68 describes the parameters for the `reapplyAllServices` operation request. By default, there are no parameters for the response unless there is an error. For information about how to add a parameter to the response to explicitly state that the operation was a success, see section 4.2.7.

Table 68 `reapplyAllServices` operation request parameters

Parameter	Description	See
objectName	The address of the object for which users run the operation The operation supports the following objects: ONT (non-MDU), MDU port, and physical port.	<i>5529 APC Template Reference</i> for identification information for objects and port types
reapplyStrategy	The operation supports the following <code>reapplyStrategy</code> values: <ul style="list-style-type: none"> MODIFY_ONLY (default) REBUILD_WHERE_NEEDED FORCE_REBUILD This parameter is optional.	—
operationInitiator	The system and user IDs of the user who is initiating the change on the port or object. This parameter is optional.	Section 4.2.6

The 5529 APC reports the `reapplyAllServices` operation in the configuration change notification interface because the operation can alter the 5529 APC configuration for a port or an ONT. As a result, the 5529 APC informs subscribed clients about the result of the `reapplyAllServices` operation as part of the reconciliation process.

4.37.1 reapplyAllServices with NO_REBUILD and -No-Value- as the optional attribute default value



Note — The information in this section does not apply to 7302 ISAM/7330 ISAM FTTN R3.7/FGN3.8.

When a reapplyAllServices operation or a rollback of a reapplyAllServices operation occurs, and -No-Value- is the optional attribute default value, the 5529 APC uses the NE default value as the expected attribute value. Some optional attributes do not have an NE default value defined, and those attributes are not configured. See the *5529 APC Template Reference* for an NE for information about optional attributes that do not have an NE default value defined.

4.38 reapplyService operation

The reapplyService operation is an audit and reconciliation operation that supports the reconciliation process for each service from the NBI. The reapplyService operation enables the 5529 APC to reconfigure the selected service as intended on the selected port or ONT; the 5529 APC keeps the intended configuration as a service template.

The reapplyService operation execution can affect dependent high-layer services, but does not affect the underlying service or the independent services.

The reapplyService operation performs reconciliation tasks on a configured service template on the object identified by the objectName. The 5529 APC supports the operation for the following objects: ONT (non-MDU), MDU port, physical port, and logical port. The operation does not change the assigned service templates and its argument values, but performs changes to the port or ONT configuration for this service to align with the intended configuration kept in 5529 APC.

The operation uses a reapplyStrategy parameter that can be set by the operator. The selected reapplyStrategy parameter affects the level of service impact. The operation supports the following options for this parameter:

- **MODIFY_ONLY** (default)
The operation reconciles the misalignment of modifiable attributes. The level of service interruption depends on the NE ability to handle attribute modifications. The operation attempts to align as many attributes as possible, and the outcome can be a complete or a partial success. Partial success means that some misalignments have been resolved, but there are still some misalignments that have not been resolved. The operation goal is to achieve a level of alignment that is as close as possible to the intended configuration for the object. This strategy does not allow full service alignment when misalignments are caused by missing objects.

- **REBUILD_WHERE_NEEDED**
The operation first attempts to use a realignment method that affects the service the least (just by modifying attributes). If the simple modification is not sufficient (for example, when the misalignment involves non-modifiable attributes, or unexpected or missing managed objects), the operation forces the realignment by unconfiguring and reconfiguring the service.
- **FORCE_REBUILD**
The operation forces the reconciliation, and service interruption is certain. The purpose of this strategy option is to reconfigure the service with the intended configuration retained by the 5529 APC, regardless of the audit result that was registered before running the operation (even though no misalignment was present).

As the intent of the `reapplyService` operation is to restore the intended configuration as retained in the 5529 APC, this operation does not attempt to roll back the misaligned configuration on the port or ONT if an error occurs during processing.



Note — The `reapplyService` operation will take longer than three to six seconds to process template stacks that contain the following network infrastructure services: SAP, VLAN, and VVPLS. Their respective 5529 APC infrastructure service templates are: ISAM SAP, ISAM VLAN for IHUB-Based ISAM, and ISAM VVPLS.

Table 69 describes the parameters for the `reapplyService` operation request. By default, there are no parameters for the response unless there is an error. For information about how to add a parameter to the response to explicitly state that the operation was a success, see section 4.2.7.

Table 69 `reapplyService` operation request parameters

Parameter	Description	See
objectName	The address of the object for which users run the operation The operation supports the following objects: ONT (non-MDU), MDU port, physical port, and logical port.	<i>5529 APC Template Reference</i> for identification information for objects and port types
templateName	The name of a candidate template for the operation	Section 4.2.2
templateVersion	The version of a candidate template for the operation. This parameter is optional.	Section 4.2.3
instanceLabel	The system and user IDs of the user who is initiating the change on the port or object. This parameter is optional.	Section 4.2.5
reapplyStrategy	The operation supports the following <code>reapplyStrategy</code> values: <ul style="list-style-type: none"> • MODIFY_ONLY (default) • REBUILD_WHERE_NEEDED • FORCE_REBUILD This parameter is optional.	—

(1 of 2)

Parameter	Description	See
operationInitiator	The system and user IDs of the user who is initiating the change on the port or object. This parameter is optional.	Section 4.2.6

(2 of 2)

The 5529 APC reports the reapplyService operation in the configuration change notification interface because the operation can alter the 5529 APC configuration for a service. As a result, the 5529 APC informs subscribed clients about the result of the reapplyService operation as part of the reconciliation process.

4.38.1 reapplyService with NO_REBUILD and -No-Value- as the optional attribute default value



Note — The information in this section does not apply to 7302 ISAM/7330 ISAM FTTN R3.7/FGN3.8.

When a reapplyService operation or a rollback of a reapplyService operation occurs, and -No-Value- is the optional attribute default value, the 5529 APC uses the NE default value as the expected attribute value. Some optional attributes do not have an NE default value defined, and those attributes are not configured. See the *5529 APC Template Reference* for an NE for information about optional attributes that do not have an NE default value defined.

4.38.2 reapplyService for an ONT with a single service and misaligned non-modifiable attributes

When the Generic ONT override mode parameter is enabled, and there is a single service on an ONT and misaligned non-modifiable attributes at the ONT level, the ONT is not deleted and the misaligned non-modifiable attributes remain. Use the reapplyAllServices operation to correct the misaligned attributes.

The Generic ONT override mode parameter is in the APC tab of the System settings window on the 5529 APC GUI; see the *5529 APC Installation, Administration, and User Guide* for more information.

4.39 resume operation

The resume operation is a service provisioning operation that unlocks or resumes the service for the port that is specified by the objectName parameter or an object created on top of the port. The object, service, or service component to be resumed (unlocked) is identified by the objectName and templateName parameters.

You can also unlock or resume one or more service components independently from the service template to which they belong by using the optional `serviceComponents` parameter in the resume operation. This functionality allows for consistency and flexibility when managing the administrative state of services; for example, when locking multiple services for a subscriber or locking part of a multi-branch service. Changing the administrative state of a service component does not affect the administrative state of the service template. The administrative state of the service component can be retrieved (`getConfiguredTemplates`), audited, and managed using the 5529 APC interfaces.

Some templates do not support the resume operation because the operation does not apply or the basic template does not support the operation. See the *5529 APC Template Reference* for the NE for information about whether a template supports suspend and resume operations.

Table 70 describes the parameters for the resume operation request. By default, there are no parameters for the response unless there is an error. For information about how to add a parameter to the response to explicitly state that the operation was a success, see section 4.2.7. If there is an error, a rollback is performed for the port by the 5529 APC.

Table 70 resume operation request parameters

Parameter	Description	See
objectName	The name of the port for the operation or the name of the object created on top of the port	<i>5529 APC Template Reference</i> for identification information for objects and port types
templateName	The name of the template for the operation	Section 4.2.3
templateVersion	Template version	Section 4.2.3
instanceLabel	Only applicable for multi-instance templates. Not applicable for regular templates. Nokia recommends the use of regular templates. To distinguish multi-instance templates from regular templates, see section 4.30.	Section 4.2.5
argument	Container element of the template argument	Section 4.2.4
name	Specifies the argument name	
value	Specifies the argument value	
serviceComponents	Container element for the serviceComponent and name parameters. The <code>serviceComponents</code> parameter represents a list of service components for the resume operation. The parameter can include as many serviceComponent and name parameter pairs as required, or contain no serviceComponent and name parameters. Changing the administrative state of a service component does not affect the administrative state of the service template. This parameter is optional.	—
serviceComponent	Container element for the name parameter	—
name	Specifies the name of the service component (basic template) that you need to unlock or resume	—

(1 of 2)

Parameter	Description	See
operationInitiator	The system and user ID of the user who is initiating the change to the port or object. This parameter is optional.	Section 4.2.6

(2 of 2)

The following considerations are relevant to designing templates:

- The suspend or resume operations can be successful for a composite template when at least one of the basic templates in the composite template supports the suspend or resume operation. An error code is returned if none of the basic templates support suspend or resume operation.
- If a composite template is resumed, only the top basic template that supports the resume operation of the composite template is affected by the resume operation. If a composite template shares basic templates with another service (implemented by other basic or composite templates applied to the port), a resume operation does not affect these other services, if the resume operation does not need to reach a template that is shared with these services.

See section [4.40](#) for information about:

- auditing the suspend and actual state of the service
- recording the suspend state when discovering services



Note — During the period that the named basic template locking strategy remains deprecated, the discovery of service templates with named basic templates that can be suspended in the NE and are different from the default locking component of the service template (top right-most lockable template in a stack) may result in an audit misalignment of the service template state. To correct the misalignment, suspend or resume the service template.

4.40 suspend operation

The suspend operation is a service provisioning operation that locks or suspends the port specified by the objectName parameter or an object created on top of the physical port. The object, service, or service component to be suspended (locked) is identified by the objectName and templateName parameters.

You can also lock or suspend one or more service components independently from the service template to which they belong by using the optional serviceComponents parameter in the suspend operation. This functionality allows for consistency and flexibility when managing the administrative state of services; for example, when locking multiple services for a subscriber or locking part of a multi-branch service. Changing the administrative state of a service component does not affect the administrative state of the service template. The administrative state of the service component can be retrieved (getConfiguredTemplates), audited, and managed using the 5529 APC interfaces.

Some templates do not support the suspend operation because the operation does not apply, or the basic template does not support the operation. See the *5529 APC Template Reference* for the NE for information about whether a template supports suspend and resume operations.

Table 71 describes the parameters for the suspend operation request. By default, there are no parameters for the response unless there is an error. For information about how to add a parameter to the response to explicitly state that the operation was a success, see section 4.2.7. This operation behaves atomically. If there is an error, a rollback is performed for the port by the 5529 APC.

Table 71 suspend operation request parameters

Parameter	Description	See
objectName	The name of the port for the operation or the name of the object created on top of the port	<i>5529 APC Template Reference</i> for identification information for objects and port types
templateName	The name of the template for the operation	Section 4.2.2
templateVersion	Template version	Section 4.2.3
instanceLabel	Only applicable for multi-instance templates. Not applicable for regular templates. Nokia recommends the use of regular templates. To distinguish multi-instance templates from regular templates, see section 4.30.	Section 4.2.5
argument	Container element for the template argument	Section 4.2.4
name	Specifies the argument name	
value	Specifies the argument value	
serviceComponents	Container element for the serviceComponent and name parameters. The serviceComponents parameter represents a list of service components for the suspend operation. The parameter can include as many serviceComponent and name parameter pairs as required, or contain no serviceComponent and name parameters. Changing the administrative state of a service component does not affect the administrative state of the service template. This parameter is optional.	—
serviceComponent	Container element for the name parameter	—
name	Specifies the name of the service component (basic template) that you need to lock or suspend	—
operationInitiator	The system and user ID of the user who is initiating the change to the port or object. This parameter is optional.	Section 4.2.6

The following is information that is relevant for designing a template:

- The suspend or resume operations can be successful for a composite template when at least one of the basic templates in the composite template supports the suspend or resume operation. An error code is returned if none of the basic templates support suspend or resume operation.
- If a composite template is suspended, only the top basic template that supports the suspend operation of the composite template is affected by the suspend operation.

If a composite template shares basic templates with another service (implemented by other basic or composite templates applied to the port), a suspend operation does not affect these other services, if the suspend operation does not need to reach a template that is shared with these services.

At template design, you can also specify which basic templates in a stack can be locked when the suspend operation is performed. See the *5529 APC Installation, Administration, and User Guide* for more information.



Note 1 — If a template was selected to be a locking component at template design, it is locked automatically when the service template is locked.

Note 2 — You cannot modify whether a template is a locking component or not in a service template that is applied in the network.

In addition to auditing the suspend state of service components (as basic templates), you can audit the actual state of the service in the NE and determine any service state misalignments between the NE and the 5529 APC. See sections [4.6](#) and [4.8](#) for information about the audit and auditPort operations. See sections [7.2.1.3](#) and [7.2.3.3](#) for examples of audit misalignment reports.

When you discover services, the suspend state of the service template and its basic components is recorded in the 5529 APC. During discovery of service templates, the suspend state is recorded as the actual suspend state of the locking component. The suspend state is also recorded for all configured service components if they are suspended (locked).

See section [4.14](#) for more information about the discoverAllServices operation.



Note — During the period that the named basic template locking strategy remains deprecated, the discovery of service templates with named basic templates that can be suspended in the NE and are different from the default locking component of the service template (top right-most lockable template in a stack) may result in an audit misalignment of the service template state. To correct the misalignment, suspend or resume the service template.

4.41 unconfigure operation

The unconfigure operation is a service provisioning operation that unconfigures the port specified by the `objectName` parameter or an object created on top of the physical port. The configuration to be unconfigured is identified by the `objectName` parameter, `templateName` parameter, and `template` arguments.

An unconfigure operation is the reverse of the configure operation with the same parameters. A successful unconfigure operation updates the 5529 APC. The unconfigure operation can be used more than once to unconfigure specific services at a time. This operation behaves atomically. If there is an error, a rollback is performed for the port by the 5529 APC.

The unconfigure operation request parameters are the same as the configure operation parameters, as described in Table 25. By default, there are no parameters for the response unless there is an error. For information about how to add a parameter to the response to explicitly state that the operation was a success, see section 4.2.7.



Note — The unconfigure operation will take longer than three to six seconds to process template stacks that contain the following network infrastructure services: SAP, VLAN, and VVPLS. Their respective 5529 APC infrastructure service templates are: ISAM SAP, ISAM VLAN for IHUB-Based ISAM, and ISAM VVPLS.

5 Service inventory

5.1 Service inventory and export services

5.1 Service inventory and export services

The 5529 APC keeps all of the intended service instances that have been created from the OSS through the 5529 APC Northbound interface or by SPFE operators. The service inventory consists of the target configurations for the service instances that the 5529 APC uses to enable the auditing of actual configurations against the intended configuration of the service instances.

The 5529 APC provides a mechanism that exports the inventory of service instances retained in the 5529 APC to a CSV output file that can be used to:

- populate the OSS after a period of operating the 5529 APC using only the SPFE (before putting the 5529 APC NBI into service)
- make the OSS aware of the template versions deployed on all ports/ONTs
- allow the OSS to perform a mass synchronization operation with the 5529 APC with respect to the services deployed using the 5529 APC

In the scope of a port or ONT, a combination of the service template name and instance label identifies the service instance. In the scope of a network, a combination of the service template name, instance label, and port or ONT identifier (a combination of NE name and port or ONT object name) identify the service instance

- allow service providers to use a CSV file containing service inventory information to perform statistical analyses; for example, to determine the total number of services or the number of services of a certain type

The properties of the service instances captured in the CSV file can be relevant to service providers; for example, the service template name, the service template name combined with the instance label, and the service type value.

The output CSV file also contains metadata for the templates that are used to configure the service instances.



Note — See the *5529 APC Installation, Administration, and User Guide* for more information about export services and the output file.

6 Collaborative service provisioning

6.1 Collaborative service provisioning overview

6.2 Configuration change notifications

6.3 Web services

6.1 Collaborative service provisioning overview

In some OSS environments, multiple OSSes are involved with the service fulfillment process or some OSSes have dependencies on what happens toward the NEs. In both cases, the 5529 APC MSAI offers a notification interface that allows OSSes to act collaboratively. The notification interface generates configuration change notifications about service provisioning and activation, independent from whether the NE supports notifications and traps. The OSS clients receive instant notification messages about configuration changes entered by other OSS clients and SPFE clients in the network.

Also, in some cases when OSS clients perform flow-through service provisioning using the 5529 APC MSAI and some 5520 AMS operators use the 5529 APC SPFE to perform expert adjustments at the EMS layer, the notification interface updates the service provisioning OSS with the changes from the SPFE users. As a result, the service provisioning OSS does not need to poll for those changes. However, the function must not be misused to distribute service provisioning information among the complete OSS.

In some OSS environments, a lead OSS architect for service provisioning or inventory is appointed. Some OSSes, for example, a planning system, would prefer to get their information from a consolidated inventory system than through notifications from the 5529 APC, which, in general, is kept aligned with service provisioning and inventory systems. Unfortunately, clear rules on this issue do not exist; each OSS architect must decide the appropriate method.

Some examples of OSSes that actively intervene in service provisioning and activation apart from the traditional service provisioning OSS that changes orders into requests to various domains are:

- test OSSes (to temporarily change configurations on a line)
- policy managers, such as the 5750 SSC, that target dynamic in-session configuration changes
- dynamic line management modules, such as the 5530 NA DLM

The 5529 APC also supports an architecture in which all service provisioning requests are handled through a central service provisioning and activation engine.

The 5529 APC allows OSSes to work together in a collaborative way, if needed.

An OSS client can discover whether it can receive configuration change notifications by retrieving the clients it can listen to through the notification interface. A 5529 APC application administrator can provision an OSS client to listen to a defined group of clients, known as a notification group. OSS clients can listen to the same notification group or to different groups. Subsequently, the OSS client can retrieve the notification topic to which it is allowed to subscribe. After the OSS client is subscribed to the topic, it receives configuration change notifications from the clients in the notification group. When an OSS client no longer needs to receive notifications, it can unsubscribe from the topic.

The OSS client must choose between a durable or non-durable connection. A durable connection allows resynchronization if connectivity is lost. When a non-durable connection is selected, the OSS client is less influenced by missing a few configuration change notifications.

The 5529 APC application administrator can specify that all requests from a notification group are reflected to the subscribing client, or only configuration requests that target changes in the NE.

The notification interface is supported in the following 5529 APC deployments with the 5520 AMS:

- standalone
- cluster

6.1.1 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.

The 5529 APC supports JMS notifications for OSS client applications running Java 1.8.

If your installation includes a NAT server, see the *5520 AMS Installation and Migration Guide* for information about how to configure the 5520 AMS server to support NAT. To configure the 5529 APC JMS client behind the NAT server, replace the application server IP address in the `java.naming.provider.url` parameter with the NAT IP address; for example, `NAT_IP_address:4447`.

6.1.2 JMS ports

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

6.1.3 JMS application parameters

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

The following parameters apply to standalone deployments:

- `java.naming.factory.initial= "org.jboss.naming.remote.client.InitialContext Factory"`
- `java.naming.provider.url="remote://host:4447"`

where *host* is the 5529 APC host IP address

The following parameters apply to cluster deployments:

- `java.naming.factory.initial= "org.jboss.naming.remote.client.InitialContext Factory"`
- `java.naming.provider.url=`use a comma-separated list of entries in the format `"remote://IP address1:4447"`; for example, `"remote://IP address2:4447, remote://IP address3:4447, remote://IP address4:4447"`

where *IP addressx* 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

6.1.4 Libraries

The 5529 APC is delivered with a set of support files (5529 APC OSS client and JBoss libraries) intended to help 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 5520 AMS cluster support requires that the JMS client code reference the following JBoss libraries in its Java classpath:

- *axs-encryption-app-release-version.jar*
- *axs-mobject-api-release-version.jar*
- *jboss-logging-3.3.0.Final.jar*
- *jdom-1.1.2-ALU-2.jar*
- *jms-1.1.jar*
- *log4j-1.2.14.jar*
- *picketbox-4.9.6.Final.jar*
- *picketbox-infinispan-4.9.6.Final.jar*
- *slf4j-log4j12-1.6.1.jar*
- *wildfly-client-all.jar*
- *xbean-2.6.0.jar*

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

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



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

For information about the support files delivered with the 5529 APC, see the *5529 Enhanced Applications Release Notice*. For information about downloading and installing the support files, see the *5520 AMS Installation and Migration Guide*.

6.1.5 Cluster JMS

The cluster JMS feature for 5529 APC notifications is available on all servers in the cluster. The JMS client application does not need to be connected to the master application server to receive notifications.

6.2 Configuration change notifications

Configuration change notifications are sent on notification topics, which are JMS topics. An OSS client retrieves its applicable JMS topic by querying the notification interface using the `getNotificationTopic` operation. The OSS client application must subscribe (or can unsubscribe, if required) using the JMS subscription API.

A configuration change notification (JMS message) is sent after completion of its related service provisioning request. Subscribed OSS clients receive the full service provisioning request as well as its response in the configuration change notification. As a result, the subscribed OSS clients can judge what has occurred on a line or ONT.

The 5529 APC application administrator can determine whether requests that target configuration changes are sent or also target information retrieval operations.

Table 72 describes the information that is contained in a configuration change notification message. The message format is based on MTOSI Version 1.1.

Table 72 Configuration change notification message description

Element	Description
SOAP header	
communicationPattern	The communication pattern of the message. The value is always Notification.
msgType	The type of message. The value is always NOTIFICATION.
destinationURI	The final destination of the message, that is to say, the operating system that processes the message content. The format of the value is <code>apc/ng/notification_topic_name</code> , where <code>notification_topic_name</code> is the name of the JMS topic where the notifications are published.
activityName	The business transaction activity of the message. The value is always notify.
msgName	The name of the message. The value is always notify.
communicationStyle	The communication style of the message. The value is always MSG.
senderURI	The application that sends the message. The value is the IP address of the 5520 AMS server.
SOAP body	
topic	The topic name to which the message is sent. The format of the value is <code>apc/ng/notification_topic_name</code> , where <code>notification_topic_name</code> is the name of the JMS topic where the notifications are published.
notificationId	A numerical ID for each notification message. The ID is incremental and restarts at 1 after reaching the maximum ID value of 9 223 372 036 854 775 807. The ID is reset to 1 after a system restart.
vendorNotificationType	The type of vendor notification. The value is always <code>ApcServiceProvisioningNotification</code> .
operationDate	Date when the configuration change operation was made. The format of the date is <code>yyyy/mm/dd</code> .
operationTime	Time when the configuration change operation was completed. The format is <code>hh:mm</code> .
operationMode	Mode of the configuration change operation. The value is always Automated.
operationKind	The kind of configuration change operation. The value is always Service Provisioning.
clientName	Name of the client from which the configuration change operation originated

(1 of 2)

Element	Description
operationName	Name of the configuration change or retrieval operation, for example, configure, unconfigure, or modify. See Table 13 for a list of the configuration change and retrieval operations.
apcRequest	<p>The 5529 APC SOAP XML request, which includes items such as the object name, template name, template version, arguments, instance label, and operation initiator. See section 4.2 for more information about these items.</p> <p>For the Move All Services operation in the SPFE, the configuration change notification (MOVE action) provides the fromPort and toPort parameters instead of the objectName parameter. The fromPort and toPort parameters indicate the object name of the source and target ports.</p>
apcResponse	The 5529 APC SOAP XML response, which includes the result of the operation (SUCCESS or FAILED) and the error code for a failed operation

(2 of 2)

6.2.1 Special notification messages

In general, the notification messages that are generated by the 5529 APC when an operation is performed contain all of the information that is contained in the SOAP XML request that is sent by the client and the response that is generated. However, there are the following special cases:

- **getNodeAuditResult** — The notification message does not display the auditNode result of the NE because the report can be very large in size. The notification message only indicates that the getNodeAuditResult operation was performed. A client that is configured to receive this notification message can retrieve the auditNode result using the requestId information that is present in the <apcRequest> part of the notification message.
- **getResults** and **getNodeAuditResults** — These two operations can be distinguished only by the requestId parameter that is provided in the request. If an incorrect requestId is provided in the request, the operation that was performed cannot be identified exactly. Regardless of the operation performed, getResults or getNodeAuditResults, a notification message that corresponds to the getResults operation is generated.
- **execute** — The priority and targetDate parameters that are present in the SOAP XML request are not used by the 5529 APC, so these two parameters do not appear in the notification message that is generated for the execute operation.
- **executeUrl** — A notification message for the executeUrl operation is not supported. Internally, the execute and executeUrl operations perform the same operation in the 5529 APC. So, if an executeUrl operation is performed, a notification message for the execute operation is generated.

6.2.2 Filtering notification messages

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 notification messages that match the filter criteria. If you did not specify a filter, the OSS client applications receive all of the event notifications sent by the 5529 APC.

You use properties in the JMS notification messages to filter on notification messages. The clients that are subscribed to receive notifications can set up the filters at the time of subscribing to a notification group. The JMS provider then filters the notification messages before sending them to the client.

Table 73 lists notification message properties that allow filtering on configuration change notifications. If a client needs to receive configuration change notifications from a specific NE, the client can supply a filter when subscribing to a notification group. In the following example, the configuration change notifications are filtered on the `APCAffectedObject` message property where the string is the object name. The object name in this example is the name of the specific NE (GPON200) and the percent symbol (%) is a wildcard character that means “any character” in the string. Using the wildcard character will make sure that all of the objects and ports in the GPON200 NE are filtered on.

```
APCAffectedObject like 'GPON200%'
```

Additional filtering is available by filtering on heartbeat and new ONT notifications on some notification message properties in addition to, or instead of, filtering on configuration change notifications. For example, if you want to filter on heartbeat and new ONT notifications where the object names for the new ONT notifications start with “NOC” and include all the respective objects and ports, you can use the following filter with the % wildcard:

```
filter=APCOperationName='Heartbeat' OR (APCOperationName='newONT' AND  
APCAffectedObject like 'NOC%')
```

For more information about the SQL-92 syntax, see section 3.8 (Message Selection) in Java Messaging Service - Version 1.1 Specification at the following URL:

<http://download.oracle.com/otndocs/jcp/7195-jms-1.1-fr-spec-oth-JSpec/>

Table 73 describes the JMS message properties on which the notification messages can be filtered. The table indicates which message properties allow you to filter on configuration change notifications, heartbeat notifications, and new ONT notifications. See section 6.2.3 for information about heartbeat notifications and see chapter 9 for information about newONT notifications.

Table 73 **Message properties**

Message property name	Type	Description	Configuration change	Heartbeat	New ONT
APCOperationDate	String	Date of the event, in the following format: <i>yyyy/mm/dd</i> where <i>yyyy</i> is the year, <i>mm</i> is the month, and <i>dd</i> is the day	✓		✓
APCOperationTime	String	Time of the event, in the following format: <i>hh:mm</i> , where <i>hh</i> is the hour, and <i>mm</i> is the minute	✓		✓
APCOperationMode	String	Mode of the configuration change operation. The value is always Automated.	✓		✓
APCOperationKind	String	The kind of configuration change operation. The value is always Service Provisioning.	✓		✓
APCClientName	String	Name of the client from which the event originated	✓		✓
APCOperationName	String	Name of the event, for example for a configuration change or retrieval operation, configure, unconfigure, or modify. See Table 13 for a list of the configuration change and retrieval operations. If you want to filter by APCOperationName on heartbeat notifications or new ONT notifications, enter "Heartbeat" or "newONT" as the value for the property	✓	✓	✓
Activity details					
APCAffectedObject	String	The name of the port or object for the operation. See section 4.2.1.	✓		✓
APCTemplateName	String	The name of the template for the operation. May or may not be applicable depending on the operation. See section 4.2.2.	✓		
APCTemplateVersion	Integer	Version of the template. The template version is an optional parameter in an operation. See section 4.2.3.	✓		
APCInstanceLabel	String	The current instance label of the template for the operation. The instance label is present only when multi-instance templates are being used. See section 4.2.5.	✓		
APCOperationInitiator	String	The system and user IDs of the user who is initiating the change on the port or object. This parameter is optional. See section 4.2.6.	✓		
Response details					
APCOperationResult	String	Result of an operation. The value is either SUCCESS or FAILED.	✓		
APCResultType	String	Result type of an operation. The value is either newOnt or Provisioning.	✓		
APCOperationErrorCode	String	An error code occurs in the case of a failed operation. See Table 12 for a list of error codes.	✓		

6.2.3 Heartbeat notifications

The 5529 APC JMS subscribers require a means of knowing that the 5529 APC is still running and the JMS connection is healthy, even when no change notifications have been sent for a long period of time. When heartbeat notification is enabled, 5529 APC sends messages with its topics to inform the client applications that the event notification service is up and running.

See section [8.1](#) for an example of a 5529 APC JMS notification for heartbeat messages.

Heartbeat notifications are sent on fixed intervals that are configurable. See the *5529 APC Installation, Administration, and User Guide* for information about the configurable 5529 APC system settings.



Note — The 5529 APC R9.6 heartbeat notification schema is different from the 5529 APC R9.5 schema. The change is backward-compatible, so R9.5 can process heartbeat notifications generated by R9.6. However, R9.6 cannot process heartbeat notifications generated by R9.5. Nokia recommends that you update OSS clients with the R9.6 heartbeat notification schema only after the server has been upgraded to the release that supports the new schema.

Tables [74](#) and [75](#) describe the SOAP envelope header and body of a 5529 APC JMS notification for heartbeat messages.

Table 74 SOAP envelope header of JMS notification for heartbeat

Element	Description	Type or value
activityName	Notification name	notify
msgName	Message name in the WSDL file	notify
msgType	Message type	NOTIFICATION ERROR
senderURI	Application sending the message	Alphanumeric string
destinationURI	Destination for the message	Alphanumeric string
communicationPattern	Communication pattern	Notification
communicationStyle	Communication style	MSG
timestamp	Date and time when the message was created	Date and time in MTOSI format; see section 6.2.3.1 for more information

Table 75 SOAP envelope body of JMS notification for heartbeat

Element	Description	Type or value
notify	Container element for the notification	—
topic	JMS subscription topic	topic/Inventory

(1 of 2)

Element	Description	Type or value
message	Container element for the notification message	—
Heartbeat	Container element for the heartbeat notification	—
notificationId	Notification ID	Alphanumeric string
objectType	Object type	OT_OS
objectName	Container element that includes the object name attribute: osNm (alphanumeric strings)	—
osNm	Name of client application that creates the notification	Alcatel-Lucent APC
osTime	Date and time when the notification was created	Date and time in MTOSI format; see section 6.2.3.1 for more information

(2 of 2)

6.2.3.1 Date and time in MTOSI UTC format

This section describes the MTOSI UTC format used for a date and time that may be displayed in a 5529 APC 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

6.3 Web services

The notification interface provides two collaborative service provisioning web services:

- getClients operation
- getNotificationTopic operation

These web services do not have any associated error codes.

6.3.1 getClients operation



Note — The getClients operation is deprecated as of 5529 APC Release 9.6.05. Nokia recommends that you use the 5520 AMS listUser NBI operation. See the *5520 AMS Northbound Interface Guide* for more information about the listUser operation.

The getClients operation is used to retrieve a list of all of the 5529 APC users (clients) that are accessible to the user that is sending the getClients operation request. The list of clients contains the users that are members of the notification group assigned to the user that is sending the getClients request. The operation response specifies the user role (client type), for example, APC Administrator or APC NBI.

Table 76 describes the getClients operation response.

Table 76 **getClients operation response description**

Parameter	Description
clientList	Container element for the list of users from which the user making the request is allowed to receive configuration change notifications
client	Container element for the user parameters
name	Name of the user. The username is configured in the 5520 AMS GUI.
description	Description of the user. The user description is configured in the 5520 AMS GUI.
attributes	Container element for the list of user attributes
attribute	Container element for a user attribute

(1 of 2)

Parameter	Description
name	Name of the attribute: Type.
value	Value of the Type attribute, which is the user role; for example, APC Administrator or APC NBI. A user can have one or more roles. Multiple role values are separated by the \r escape character. The format of the parameter value is <i>role\rrole</i> , for example, APC Administrator\rAPC NBI. See section 4.30.1 for more information about the \r character in operation responses.

(2 of 2)

6.3.2 getNotificationTopic

The getNotificationTopic operation is used to retrieve the JMS topic from which the client can receive configuration change notifications if the client is subscribed to the topic. The client is required to be authenticated before subscribing to the JMS topic. See the *5529 APC Installation, Administration, and User Guide* for information about assigning clients to a notification group.

See chapter 8 for an example of a getNotificationTopic operation request and response.

Table 77 describes the getNotificationTopic operation request.

Table 77 getNotificationTopic operation request description

Parameter	Description
eventType	The type of event as specified by MTOSI. The value is always VendorNotification.
notificationType	The type of vendor notification. The value is always ApcServiceProvisioningNotification.

Table 78 describes the getNotificationTopic operation response.

Table 78 getNotificationTopic operation response description

Parameter	Description
topicName	Name of the JMS topic to which the configuration change notifications are sent for the client making the request. The format is <i>topic/apc/notification_topic_name</i> , where <i>notification_topic_name</i> is the name of the JMS topic that corresponds to the notification group that is assigned to the user performing the getNotificationTopic operation.

Reference

- [7 Examples of web service operations, responses, and bulk request files](#)
- [8 Configuration change notification message and web service examples](#)
- [9 New ONT notification](#)
- [10 NE reparenting notifications](#)

7 Examples of web service operations, responses, and bulk request files

7.1 Examples of web service operations

7.2 Examples of web service operation responses

7.3 Examples of imported bulk request files

7.4 Examples of audit results

7.1 Examples of web service operations

This section shows examples of web service operations.

- [Examples of the configure operation](#)
- [Examples of template migration operations](#)
- [Example of the getTemplateMetaData operation](#)
- [Example of the getConfiguredServices operation](#)

7.1.1 Examples of the configure operation

The following is an example of the configure request operation.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns="uri://alcatel.com/apc/9.6">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:configure>
      <objectName>135.249.41.119:1-1-10-1-3</objectName>
      <templateName>10M10M</templateName>
      <templateVersion>1</templateVersion>
      <instanceLabel>1-1-1002-100</instanceLabel>
      <argument>
        <name>hsiVlanId</name>
        <value>1001</value>
      </argument>
      <argument>
        <name>ontType</name>
        <value>GMDU</value>
      </argument>
      <argument>
        <name>cardType</name>
        <value>Ethernet</value>
      </argument>
    </ns:configure>
  </soapenv:Body>
</soapenv:Envelope>
```

```

        </argument>
        <argument>
            <name>ontCardId</name>
            <value>2</value>
        </argument>
        <argument>
            <name>portType</name>
            <value>ontEthernetPort</value>
        </argument>
        <argument>
            <name>ethPortId</name>
            <value>1</value>
        </argument>
        <argument>
            <name>cvlanId</name>
            <value>1002</value>
        </argument>
        <operationInitiator>Admin</operationInitiator>
    </ns:configure>
</soapenv:Body>
</soapenv:Envelope>

```

The following is an example of a success response for a configure operation. By default, there are no parameters for the response of a successful operation. For information about how to add a parameter to the response to explicitly state that the operation was a success, see section [4.2.7](#).

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:configureResponse xmlns:ns="uri://alcatel.com/apc/9.6"/>
  </soapenv:Body>
</soapenv:Envelope>

```

The following is an example of a failure response for a configure operation.

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Server</faultcode>
      <faultstring>RemoteApcExceptionMessage</faultstring>
      <detail>
        <ns:RemoteApcException xmlns:ns="uri://alcatel.com/apc/9.6">
          <message>NE not found: GPON</message>
          <errorCode>DSLAM_NOT_FOUND</errorCode>
        </ns:RemoteApcException>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>

```

7.1.2 Examples of template migration operations

The following is an example of the execute operation for a template migration.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns="uri://alcatel.com/apc/9.6">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:execute>
      <command>
        <commandId>1</commandId>
        <action>MIGRATE</action>
        <objectName>135.249.41.119:1-1-10-1-3</objectName>
        <templateName>10M10M</templateName>
        <instanceLabel>No.1</instanceLabel>
        <templateVersion>1</templateVersion>
        <oldTemplateName>20M20M</oldTemplateName>
        <oldInstanceLabel>No.2</oldInstanceLabel>
        <oldTemplateVersion>1</oldTemplateVersion>
        <argument>
          <name>ethPortId</name>
          <value>1</value>
        </argument>
        <argument>
          <name>ontCardId</name>
          <value>1</value>
        </argument>
        <argument>
          <name>portType</name>
          <value>ontEthernetPort</value>
        </argument>
        <argument>
          <name>hsiServiceLabel</name>
          <value/>
        </argument>
        <argument>
          <name>cvlanId</name>
          <value>0</value>
        </argument>
        <argument>
          <name>hsiServiceId</name>
          <value>1</value>
        </argument>
        <argument>
          <name>cardType</name>
          <value>Ethernet</value>
        </argument>
        <argument>
          <name>ontType</name>
          <value>GMDU</value>
        </argument>
        <argument>
          <name>hsiStaticCVlanID</name>
          <value/>
        </argument>
        <argument>
          <name>hsiVlanId</name>
          <value>100</value>
        </argument>
      </command>
    </ns:execute>
  </soapenv:Body>
</soapenv:Envelope>
```

```

        <argument>
            <name>hsiCustomerId</name>
            <value/>
        </argument>
    </command>
<command>
    <commandId>2</commandId>
    <action>MIGRATE</action>
    <objectName>135.249.41.119:1-1-10-1-3</objectName>
    <templateName>20M20M</templateName>
        <instanceLabel>No.2</instanceLabel>
        <templateVersion>1</templateVersion>
        <oldTemplateName>10M10M</oldTemplateName>
        <oldInstanceLabel>No.1</oldInstanceLabel>
        <oldTemplateVersion>1</oldTemplateVersion>
    <argument>
        <name>ethPortId</name>
        <value>1</value>
    </argument>
    <argument>
        <name>ontCardId</name>
        <value>2</value>
    </argument>
    <argument>
        <name>portType</name>
        <value>ontEthernetPort</value>
    </argument>
    <argument>
        <name>hsiServiceLabel</name>
        <value/>
    </argument>
    <argument>
        <name>cvlanId</name>
        <value>0</value>
    </argument>
    <argument>
        <name>hsiServiceId</name>
        <value>1</value>
    </argument>
    <argument>
        <name>cardType</name>
        <value>Ethernet</value>
    </argument>
    <argument>
        <name>ontType</name>
        <value>GMDU</value>
    </argument>
    <argument>
        <name>hsiStaticCVlanID</name>
        <value/>
    </argument>
    <argument>
        <name>hsiVlanId</name>
        <value>100</value>
    </argument>
    <argument>
        <name>hsiCustomerId</name>
        <value/>
    </argument>

```



```

        </command>
      </ns:execute>
    </soapenv:Body>
  </soapenv:Envelope>

```

The following is an example of an execute operation response for a template migration.

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:executeResponse xmlns:ns="uri://alcatel.com/apc/9.6">
      <requestId>13</requestId>
    </ns:executeResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

The following is an example of a template migration request.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="uri://alcatel.com/apc/9.6">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:migrate>
      <objectName>135.249.41.119:1-1-10-1-3</objectName>
      <oldTemplateName>10M10M</oldTemplateName>
      <oldTemplateVersion>1</oldTemplateVersion>
      <oldTemplateInstanceLabel>no.1</oldTemplateInstanceLabel>
      <oldArgument>
        <name>ontSubscriberLocationId</name>
        <value>WILDCARD</value>
      </oldArgument>
      <newTemplateName>20M20M</newTemplateName>
      <newTemplateVersion>1</newTemplateVersion>
      <newTemplateInstanceLabel>no.2</newTemplateInstanceLabel>
      <newArgument>
        <name>ethPortId</name>
        <value>1</value>
      </newArgument>
      <newArgument>
        <name>ontCardId</name>
        <value>2</value>
      </newArgument>
      <newArgument>
        <name>portType</name>
        <value>ontEthernetPort</value>
      </newArgument>
      <newArgument>
        <name>hsiServiceLabel</name>
        <value/>
      </newArgument>
      <newArgument>
        <name>cvlanId</name>
        <value>0</value>
      </newArgument>
      <newArgument>
        <name>hsiServiceId</name>
        <value>1</value>
      </newArgument>
    </ns:migrate>
  </soapenv:Body>
</soapenv:Envelope>

```

```

        </newArgument>
        <newArgument>
            <name>cardType</name>
            <value>Ethernet</value>
        </newArgument>
        <newArgument>
            <name>ontType</name>
            <value>GMDU</value>
        </newArgument>
        <newArgument>
            <name>hsiStaticCVlanID</name>
            <value/>
        </newArgument>
        <newArgument>
            <name>hsiVlanId</name>
            <value>100</value>
        </newArgument>
        <newArgument>
            <name>hsiCustomerId</name>
            <value/>
        </newArgument>
        <operationInitiator>Admin</operationInitiator>
    </ns:migrate>
</soapenv:Body>
</soapenv:Envelope>

```

The following is an example of a template migration response. By default, there are no parameters for the response of a successful operation. For information about how to add a parameter to the response to explicitly state that the operation was a success, see section 4.2.7.

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Header/>
    <soapenv:Body>
        <ns:migrateResponse xmlns:ns="uri://alcatel.com/apc/9.6"/>
    </soapenv:Body>
</soapenv:Envelope>

```

The following is an example of a template migration request in which the migration retains configured argument values.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="uri://alcatel.com/apc/9.5">
    <soapenv:Header/>
    <soapenv:Body>
        <ns:migrate>
            <objectName>ISAM45:1-1-7-2-2</objectName>
            <oldTemplateName>Captive</oldTemplateName>
            <oldTemplateVersion>1</oldTemplateVersion>
            // <oldTemplateInstanceLabel>Config</oldTemplateInstanceLabel>
            <oldArgument>
                <name>ontSerialNumber</name>
                <value>ALCLF8A8EAD6</value>
            </oldArgument>
            <oldArgument>
                <name>ontBatteryBackup</name>
                <value>TRUE</value>
            </oldArgument>
        </ns:migrate>
    </soapenv:Body>
</soapenv:Envelope>

```

```

</oldArgument>
<newTemplateName>Data_Service</newTemplateName>
<newTemplateVersion>1</newTemplateVersion>
//<newTemplateInstanceLabel>NewILConfig</newTemplateInstanceLabel>
<newArgument>
  <name>ontSerialNumber</name>
  <value>APC_RETAIN_CONFIGURATION</value>
</newArgument>
<newArgument>
  <name>ontBatteryBackup</name>
  <value>APC_RETAIN_CONFIGURATION</value>
</newArgument>
<newArgument>
  <name>vlanAssociationCustomerName</name>
  <value>1714555555</value>
</newArgument>
</ns:migrate>
</soapenv:Body>
</soapenv:Envelope>

```

7.1.3 Example of the getTemplateMetaData operation

The following is an example of a getTemplateMetaData operation request.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="uri://alcatel.com/apc/9.6">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:getTemplateMetaData>
      <templateName>24SIP_FTP</templateName>
      <templateVersion>1</templateVersion>
      <!--Optional:-->
    </ns:getTemplateMetaData>
  </soapenv:Body>
</soapenv:Envelope>

```

7.1.4 Example of the getConfiguredServices operation

The following is an example of a getConfiguredServices operation request.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="uri://alcatel.com/apc/9.6">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:getConfiguredServices>
      <objectName>135.249.41.119:1-1-10-1-3</objectName>
      <filter>
        <argumentList>
          <nameAndStringValue>
            <argument>
              <name>hsiVlanId</name>
              <value>100</value>
            </argument>

```

```
        </nameAndStringValue>
      </argumentList>
    </filter>
  </ns:getConfiguredServices>
</soapenv:Body>
</soapenv:Envelope>
```

7.2 Examples of web service operation responses

This section shows examples of the following web service operations.

- [audit operation responses](#)
- [auditNode operation responses](#)
- [auditPort operation responses](#)
- [getConfiguredServices operation response](#)
- [getLogicalPorts operation response](#)
- [getNodeAuditResult operation response](#)
- [getTemplateMetadata operation response](#)

7.2.1 audit operation responses

This section shows examples of the following audit operation responses.

- [Successful audit operation response](#)
- [Audit operation response with misalignments](#)
- [Audit operation response with service state misalignments](#)

The examples are for an HSI service stack that contains an ONT template, an ONT card template, an ONT port template, and an HSI template.

7.2.1.1 Successful audit operation response

The following is an example of an audit operation successful response.

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:auditResponse xmlns:ns="uri://alcatel.com/apc/9.6">
      <auditReport>
        <templateName>VZ_Ont</templateName>
        <templateType>G7342 ONT</templateType>
        <configured>true</configured>
      </auditReport>
      <auditReport>
        <templateName>VZ_Ethernet_Card</templateName>
```

```

        <templateType>G7342 ONT Card</templateType>
        <configured>true</configured>
    </auditReport>
    <auditReport>
        <templateName>VZ_Ethernet_Port</templateName>
        <templateType>G7342 ONT Ethernet Port</templateType>
        <configured>true</configured>
    </auditReport>
    <auditReport>
        <templateName>VZ_BridgePort</templateName>
        <templateType>G7342 Bridge Port</templateType>
        <configured>true</configured>
    </auditReport>
    <auditReport>
        <templateName>VZ_HSI_20M20M</templateName>
        <templateType>G7342 HSI Service</templateType>
        <configured>true</configured>
    </auditReport>
</ns:auditResponse>
</soapenv:Body>
</soapenv:Envelope>

```

7.2.1.2 Audit operation response with misalignments

The following is an example of an audit operation response that reports misalignments.

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:auditResponse xmlns:ns="uri://alcatel.com/apc/9.6">
      <auditReport>
        <templateName>VZ_Ont</templateName>
        <templateType>G7342 ONT</templateType>
        <configured>true</configured>
        <remoteFailedAssertion>
          <property>ontSubscriberLocationId</property>
          <expected>WILDCARD</expected>
          <actual>WILDCARD1</actual>
        </remoteFailedAssertion>
      </auditReport>
      <auditReport>
        <templateName>VZ_Ethernet_Card</templateName>
        <templateType>G7342 ONT Card</templateType>
        <configured>true</configured>
      </auditReport>
      <auditReport>
        <templateName>VZ_Ethernet_Port</templateName>
        <templateType>G7342 ONT Ethernet Port</templateType>
        <configured>true</configured>
      </auditReport>
      <auditReport>
        <templateName>VZ_BridgePort</templateName>
        <templateType>G7342 Bridge Port</templateType>
        <configured>true</configured>
      </auditReport>
    </ns:auditResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

```

        </auditReport>
      <auditReport>
        <templateName>VZ_HSI_20M20M</templateName>
        <templateType>G7342 HSI Service</templateType>
        <configured>true</configured>
      <remoteFailedAssertion>
        <property>hsiCustomerId</property>
        <expected/>
        <actual>EDITED</actual>
      </remoteFailedAssertion>
    </auditReport>
  </ns:auditResponse>
</soapenv:Body>
</soapenv:Envelope>

```

7.2.1.3 Audit operation response with service state misalignments

The following is an example of an audit operation response that reports a misaligned service template state.

```

<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:auditResponse xmlns:ns="uri://alcatel.com/apc/9.6">
      <auditReport>
        <templateName>RF_Video</templateName>
        <templateType>Stack</templateType>
        <configured>true</configured>
        <remoteFailedAssertion>
          <property>Suspended</property>
          <expected>false</expected>
          <actual>true</actual>
        </remoteFailedAssertion>
      </auditReport>
      <auditReport>
        <templateName>VZ_Ont</templateName>
        <templateType>G7342 ONT</templateType>
        <configured>true</configured>
        <remoteFailedAssertion>
          <property>Suspended</property>
          <expected>false</expected>
          <actual>true</actual>
        </remoteFailedAssertion>
      </auditReport>
      <auditReport>
        <templateName>VZ_RF_Video_Card</templateName>
        <templateType>G7342 ONT Card</templateType>
        <configured>true</configured>
      </auditReport>
      <auditReport>
        <templateName>VZ_Video_Port</templateName>
        <templateType>G7342 ONT Video Port</templateType>
        <configured>true</configured>
      </auditReport>
    </ns:auditResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

```

        </auditReport>
      </ns:auditResponse>
    </soapenv:Body>
  </soapenv:Envelope>

```

The following is an example of an audit operation response that reports a misaligned admin state for a service component.

```

<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:auditResponse xmlns:ns="uri://alcatel.com/apc/9.6">
      <auditReport>
        <templateName>VZ_Ont</templateName>
        <templateType>G7342 ONT</templateType>
        <configured>true</configured>
        <remoteFailedAssertion>
          <property>Suspended</property>
          <expected>false</expected>
          <actual>true</actual>
        </remoteFailedAssertion>
      </auditReport>
      <auditReport>
        <templateName>VZ_RF_Video_Card</templateName>
        <templateType>G7342 ONT Card</templateType>
        <configured>true</configured>
      </auditReport>
      <auditReport>
        <templateName>VZ_Video_Port</templateName>
        <templateType>G7342 ONT Video Port</templateType>
        <configured>true</configured>
      </auditReport>
    </ns:auditResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

7.2.2 auditNode operation responses

This section shows examples of the following auditNode operation responses:

- [Clean port](#)
- [Port not clean](#)
- [Target object missing in NE](#)
- [Template not configured](#)

7.2.2.1 Clean port

The following is part of an auditNode operation response when a port is clean, and the physical port is not configured.

```
<portResult address="1-1-12-3">
  <auditResult/>
</portResult>
```

7.2.2.2 Port not clean

The following is part of an auditNode operation response that illustrates a PORT_NOT_CLEAN error code, reported when the physical port is configured by other means than the 5529 APC.

```
<portResult address="1-1-4-1">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
```

7.2.2.3 Target object missing in NE

The following is part of an auditNode operation response that illustrates a TARGET_OBJECT_MISSING_IN_NE error code, reported when the root port is missing on the NE.

```
<portResult address="1-1-5-6-3">
  <auditResult errorCode="TARGET_OBJECT_MISSING_IN_NE" xmlns="">
    <serviceTemplateId>
      <templateName>Isam ont stack</templateName>
      <templateVersion>1</templateVersion>
      <instanceLabel>1</instanceLabel>
    </serviceTemplateId>
    <auditReport>
      <templateName/>
      <templateType>ISAM ONT</templateType>
      <configured>>false</configured>
    </auditReport>
    <auditReport>
      <templateName/>
      <templateType>ISAM ONT Card</templateType>
      <configured>>false</configured>
    </auditReport>
  </auditResult>
</portResult>
```

7.2.2.4 Template not configured

The following is part of an auditNode operation response that illustrates a TEMPLATE_NOT_CONFIGURED error code, reported when the configured status of a service containing a template is false.

```
<portResult address="1-1-2-9">
  <auditResult errorCode="TEMPLATE_NOT_CONFIGURED">
    <serviceTemplateId>
```



```

        <templateName>xdsl stack</templateName>
        <templateVersion>1</templateVersion>
    </serviceTemplateId>
    <auditReport>
        <templateName>xdsl</templateName>
        <templateType>XDSL (ETSI)</templateType>
        <configured>true</configured>
        <remoteFailedAssertion>
            <property>customerId</property>
            <expected/>
            <actual>available</actual>
        </remoteFailedAssertion>
        <remoteFailedAssertion>
            <property>serviceProfileName</property>
            <expected>1</expected>
            <actual/>
        </remoteFailedAssertion>
        <remoteFailedAssertion>
            <property>Suspended</property>
            <expected>false</expected>
            <actual>true</actual>
        </remoteFailedAssertion>
        <remoteFailedAssertion>
            <property>dpboProfileName</property>
            <expected>1</expected>
            <actual/>
        </remoteFailedAssertion>
        <remoteFailedAssertion>
            <property>spectrumProfileName</property>
            <expected>1</expected>
            <actual/>
        </remoteFailedAssertion>
    </auditReport>
    <auditReport>
        <templateName/>
        <templateType>XDSL line overrule (ETSI)</templateType>
        <configured>false</configured>
    </auditReport>
</auditResult>
</portResult>

```

7.2.3 auditPort operation responses

This section shows examples of the following auditPort operation responses.

- [auditPort operation response for an aligned service](#)
- [auditPort operation response for a misaligned service](#)
- [auditPort operation response for misaligned service states](#)

7.2.3.1 auditPort operation response for an aligned service

The following is an example of an auditPort operation response for an aligned service.

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:auditPortResponse xmlns:ns="uri://alcatel.com/apc/9.6">
      <auditResult>
        <serviceTemplateId>
          <templateName>20M20M</templateName>
          <templateVersion>1</templateVersion>
          <instanceLabel>no.2</instanceLabel>
        </serviceTemplateId>
        <auditReport>
          <templateName>VZ_Ont</templateName>
          <templateType>G7342 ONT</templateType>
          <configured>true</configured>
        </auditReport>
        <auditReport>
          <templateName>VZ_Ethernet_Card</templateName>
          <templateType>G7342 ONT Card</templateType>
          <configured>true</configured>
        </auditReport>
        <auditReport>
          <templateName>VZ_Ethernet_Port</templateName>
          <templateType>G7342 ONT Ethernet Port</templateType>
          <configured>true</configured>
        </auditReport>
        <auditReport>
          <templateName>VZ_BridgePort</templateName>
          <templateType>G7342 Bridge Port</templateType>
          <configured>true</configured>
        </auditReport>
        <auditReport>
          <templateName>VZ_HSI_20M20M</templateName>
          <templateType>G7342 HSI Service</templateType>
          <configured>true</configured>
        </auditReport>
        <auditReport>
          <templateName/>
          <templateType>apc.g7342.ont.card</templateType>
          <configured>true</configured>
        </auditReport>
      </auditResult>
    </ns:auditPortResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

7.2.3.2 auditPort operation response for a misaligned service

The following is an example of an auditPort operation response for a misaligned service.

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:auditPortResponse xmlns:ns="uri://alcatel.com/apc/9.6">
      <auditResult errorCode="TEMPLATE_NOT_CONFIGURED">
        <serviceTemplateId>
          <templateName>20M20M</templateName>
          <templateVersion>1</templateVersion>
          <instanceLabel>no.2</instanceLabel>
        </serviceTemplateId>
        <auditReport>
          <templateName>VZ_Ont</templateName>
          <templateType>G7342 ONT</templateType>
          <configured>true</configured>
          <remoteFailedAssertion>
            <property>ontSubscriberLocationId</property>
            <expected>WILDCARD</expected>
            <actual>WILDCARD1</actual>
          </remoteFailedAssertion>
          <remoteFailedAssertion>
            <property>identification_SlidAccessibility</property>
            <expected>Disabled</expected>
            <actual>Enabled_Read_Only</actual>
          </remoteFailedAssertion>
        </auditReport>
        <auditReport>
          <templateName>VZ_Ethernet_Card</templateName>
          <templateType>G7342 ONT Card</templateType>
          <configured>true</configured>
        </auditReport>
        <auditReport>
          <templateName>VZ_Ethernet_Port</templateName>
          <templateType>G7342 ONT Ethernet Port</templateType>
          <configured>true</configured>
        </auditReport>
        <auditReport>
          <templateName>VZ_BridgePort</templateName>
          <templateType>G7342 Bridge Port</templateType>
          <configured>true</configured>
        </auditReport>
        <auditReport>
          <templateName>VZ_HSI_20M20M</templateName>
          <templateType>G7342 HSI Service</templateType>
          <configured>true</configured>
        </auditReport>
      </auditResult>
    </ns:auditPortResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

7.2.3.3 auditPort operation response for misaligned service states

The following is an example of an auditPort operation response for a misaligned service template state and a misaligned service component state.

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:auditPortResponse xmlns:ns="uri://alcatel.com/apc/9.6">
      <auditResult errorCode="TEMPLATE_NOT_CONFIGURED">
        <serviceTemplateId>
          <templateName>RF_Video</templateName>
          <templateVersion>1</templateVersion>
          <instanceLabel>card4-port1</instanceLabel>
        </serviceTemplateId>
        <auditReport>
          <templateName>RF_Video</templateName>
          <templateType>Stack</templateType>
          <configured>true</configured>
          <remoteFailedAssertion>
            <property>Suspended</property>
            <expected>false</expected>
            <actual>true</actual>
          </remoteFailedAssertion>
        </auditReport>
        <auditReport>
          <templateName>VZ_Ont</templateName>
          <templateType>G7342 ONT</templateType>
          <configured>true</configured>
        </auditReport>
        <auditReport>
          <templateName>VZ_RF_Video_Card</templateName>
          <templateType>G7342 ONT Card</templateType>
          <configured>true</configured>
        </auditReport>
        <auditReport>
          <templateName>VZ_Video_Port</templateName>
          <templateType>G7342 ONT Video Port</templateType>
          <configured>true</configured>
          <remoteFailedAssertion>
            <property>Suspended</property>
            <expected>false</expected>
            <actual>true</actual>
          </remoteFailedAssertion>
        </auditReport>
      </auditResult>
    </ns:auditPortResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

7.2.4 getConfiguredServices operation response

The following is an example of a getConfiguredServices operation response.

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:getConfiguredServicesResponse
xmlns:ns="uri://alcatel.com/apc/9.6">
      <apcServiceDataResponse>
        <apcServiceEntry>
          <objectName>135.249.41.119:1-1-10-1-3</objectName>
          <templateName>20M20M</templateName>
          <version>1</version>
          <instanceLabel>no.2</instanceLabel>
          <tags>Data</tags>
          <operationInitiator>SPFE\admin</operationInitiator>
          <timeStamp>20170220041413.2Z</timeStamp>
          <suspendstate>SUSPENDED</suspendstate>
          <argument>
            <name>ethPortId</name>
            <value>1</value>
          </argument>
          <argument>
            <name>ontCardId</name>
            <value>2</value>
          </argument>
          <argument>
            <name>portType</name>
            <value>ontEthernetPort</value>
          </argument>
          <argument>
            <name>hsiServiceLabel</name>
            <value/>
          </argument>
          <argument>
            <name>cvlanId</name>
            <value>0</value>
          </argument>
          <argument>
            <name>hsiServiceId</name>
            <value>1</value>
          </argument>
          <argument>
            <name>cardType</name>
            <value>Ethernet</value>
          </argument>
          <argument>
            <name>ontType</name>
            <value>GMDU</value>
          </argument>
          <argument>
            <name>hsiStaticCVlanID</name>
            <value/>
          </argument>
          <argument>
            <name>hsiVlanId</name>
            <value>100</value>
          </argument>
          <argument>
            <name>hsiCustomerId</name>
            <value/>
          </argument>
        </apcServiceEntry>
      </apcServiceDataResponse>
    </ns:getConfiguredServicesResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

```

        </argument>
      </apcServiceEntry>
    <endOfReply>true</endOfReply>
  </apcServiceDataResponse>
</ns:getConfiguredServicesResponse>
</soapenv:Body>
</soapenv:Envelope>

```

7.2.5 getLogicalPorts operation response

The following is an example of a getLogicalPorts operation response.

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:getLogicalPortsResponse xmlns:ns="uri://alcatel.com/apc/9.6">
      <logicalPort>135.249.41.119:1-1-10-1-3</logicalPort>
      <logicalPort>135.249.41.119:1-1-10-1-3-1</logicalPort>
      <logicalPort>135.249.41.119:1-1-10-1-3-1-1</logicalPort>
      <logicalPort>135.249.41.119:1-1-10-1-3-1-1-1</logicalPort>
      <logicalPort>135.249.41.119:1-1-10-1-3-2</logicalPort>
      <logicalPort>135.249.41.119:1-1-10-1-3-2-1</logicalPort>
      <logicalPort>135.249.41.119:1-1-10-1-3-2-1-1</logicalPort>
    </ns:getLogicalPortsResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

7.2.6 getNodeAuditResult operation response

The following is an example of a getNodeAuditResult operation response.

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:getNodeAuditResultResponse xmlns:ns="uri://alcatel.com/apc/9.6">
      <commandResult>
        <state>READY</state>
        <executionDate>2017-02-20T14:52:30.000+07:00</executionDate>
        <success>true</success>
        <portResult address="1-1-5-2-4">
          <auditResult errorCode="PORT_NOT_CLEAN"/>
        </portResult>
        <portResult address="1-1-5-2-3">
          <auditResult errorCode="PORT_NOT_CLEAN"/>
        </portResult>
        <portResult address="1-1-5-2-6">
          <auditResult errorCode="PORT_NOT_CLEAN"/>
        </portResult>
        <portResult address="1-1-5-2-5">
          <auditResult errorCode="PORT_NOT_CLEAN"/>
        </portResult>
      </commandResult>
    </ns:getNodeAuditResultResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

```

    <portResult address="1-1-5-4-3">
      <auditResult errorCode="PORT_NOT_CLEAN"/>
    </portResult>
    <portResult address="1-1-5-2-8">
      <auditResult errorCode="PORT_NOT_CLEAN"/>
    </portResult>
    <portResult address="1-1-5-2-7">
      <auditResult errorCode="PORT_NOT_CLEAN"/>
    </portResult>
    <portResult address="1-1-5-2-9">
      <auditResult errorCode="PORT_NOT_CLEAN"/>
    </portResult>
    <portResult address="1-1-9-1-1">
      <auditResult errorCode="PORT_NOT_CLEAN"/>
    </portResult>
    <portResult address="1-1-9-3-1">
      <auditResult errorCode="PORT_NOT_CLEAN"/>
    </portResult>
    <portResult address="1-1-9-1-3">
      <auditResult errorCode="PORT_NOT_CLEAN"/>
    </portResult>
    <portResult address="1-1-9-1-2">
      <auditResult errorCode="PORT_NOT_CLEAN"/>
    </portResult>
    <portResult address="1-1-5-2-2">
      <auditResult errorCode="PORT_NOT_CLEAN"/>
    </portResult>
    <portResult address="1-1-5-2-1">
      <auditResult errorCode="PORT_NOT_CLEAN"/>
    </portResult>
    <portResult address="1-1-1-1-1">
      <auditResult errorCode="PORT_NOT_CLEAN"/>
    </portResult>
    <portResult address="1-1-6-2-1">
      <auditResult errorCode="PORT_NOT_CLEAN"/>
    </portResult>
    <portResult address="1-1-2-1-3">
      <auditResult errorCode="PORT_NOT_CLEAN"/>
    </portResult>
    <portResult address="1-1-2-3-1">
      <auditResult errorCode="PORT_NOT_CLEAN"/>
    </portResult>
    <portResult address="1-1-2-1-2">
      <auditResult errorCode="PORT_NOT_CLEAN"/>
    </portResult>
    <portResult address="1-1-6-4-3">
      <auditResult>
        <serviceTemplateId>
          <templateName>10M10M</templateName>
          <templateVersion>1</templateVersion>
          <instanceLabel>card1-port1-service2</instanceLabel>
        </serviceTemplateId>
        <auditReport>
          <templateName>VZ_Ont</templateName>
          <templateType>G7342 ONT</templateType>
          <configured>true</configured>
        </auditReport>
        <auditReport>
          <templateName>VZ_Ethernet_Card</templateName>

```

```

        <templateType>G7342 ONT Card</templateType>
        <configured>true</configured>
    </auditReport>
    <auditReport>
        <templateName>VZ_Ethernet_Port</templateName>
        <templateType>G7342 ONT Ethernet Port</templateType>
        <configured>true</configured>
    </auditReport>
    <auditReport>
        <templateName>VZ_BridgePort</templateName>
        <templateType>G7342 Bridge Port</templateType>
        <configured>true</configured>
    </auditReport>
    <auditReport>
        <templateName>VZ_HSI_10M10M</templateName>
        <templateType>G7342 HSI Service</templateType>
        <configured>true</configured>
    </auditReport>
</auditResult>
</portResult>
<portResult address="1-1-10-4-1">
    <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-2-1-1">
    <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-6-4-2">
    <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-10-2-2">
    <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-6-4-1">
    <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-10-2-1">
    <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-3-3-1">
    <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-11-1-2">
    <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-11-1-1">
    <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-3-1-1">
    <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-7-4-2">
    <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-7-4-1">
    <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-8-2-3">
    <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>

```



```
<portResult address="1-1-8-4-1">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-8-2-2">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-12-1-2">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-8-2-1">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-12-1-1">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-8-2-5">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-8-4-3">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-8-4-2">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-4-1-1">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-5-2-10">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-5-1-4">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-5-2-11">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-5-2-14">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-5-2-15">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-5-2-12">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-5-2-13">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-9-2-2">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-9-2-1">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-5-1-1">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-9-2-6">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
```

```
</portResult>
<portResult address="1-1-5-3-1">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-5-1-2">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-6-1-2">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-6-1-1">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-1-2-2">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-10-1-9">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-2-4-3">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-5-2-18">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-10-1-8">
  <auditResult errorCode="TEMPLATE_NOT_CONFIGURED">
    <serviceTemplateId>
      <templateName>0_10M10M</templateName>
      <templateVersion>1</templateVersion>
    </serviceTemplateId>
    <auditReport>
      <templateName>VZ_Ont</templateName>
      <templateType>G7342 ONT</templateType>
      <configured>true</configured>
      <remoteFailedAssertion>
        <property>Suspended</property>
        <expected>false</expected>
        <actual>true</actual>
      </remoteFailedAssertion>
    </auditReport>
    <auditReport>
      <templateName>VZ_Ethernet_Card</templateName>
      <templateType>G7342 ONT Card</templateType>
      <configured>true</configured>
    </auditReport>
    <auditReport>
      <templateName>VZ_Ethernet_Port</templateName>
      <templateType>G7342 ONT Ethernet Port</templateType>
      <configured>true</configured>
    </auditReport>
    <auditReport>
      <templateName>VZ_BridgePort</templateName>
      <templateType>G7342 Bridge Port</templateType>
      <configured>true</configured>
    </auditReport>
    <auditReport>
      <templateName>VZ_HSI_10M10M</templateName>
      <templateType>G7342 HSI Service</templateType>
```

```

        <configured>true</configured>
      </auditReport>
    </auditResult>
  </portResult>
<portResult address="1-1-2-4-2">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-10-1-7">
  <auditResult errorCode="TEMPLATE_NOT_CONFIGURED">
    <serviceTemplateId>
      <templateName>20M20M</templateName>
      <templateVersion>1</templateVersion>
      <instanceLabel>1-1-1002-100</instanceLabel>
    </serviceTemplateId>
    <auditReport>
      <templateName>VZ_Ont</templateName>
      <templateType>G7342 ONT</templateType>
      <configured>true</configured>
      <remoteFailedAssertion>
        <property>Suspended</property>
        <expected>false</expected>
        <actual>true</actual>
      </remoteFailedAssertion>
    </auditReport>
  </auditReport>
  <auditReport>
    <templateName>VZ_Ethernet_Card</templateName>
    <templateType>G7342 ONT Card</templateType>
    <configured>true</configured>
  </auditReport>
  <auditReport>
    <templateName>VZ_Ethernet_Port</templateName>
    <templateType>G7342 ONT Ethernet Port</templateType>
    <configured>true</configured>
  </auditReport>
  <auditReport>
    <templateName>VZ_BridgePort</templateName>
    <templateType>G7342 Bridge Port</templateType>
    <configured>true</configured>
  </auditReport>
  <auditReport>
    <templateName>VZ_HSI_20M20M</templateName>
    <templateType>G7342 HSI Service</templateType>
    <configured>true</configured>
  </auditReport>
  <auditReport>
    <templateName/>
    <templateType>apc.g7342.ont.card</templateType>
    <configured>true</configured>
    <remoteFailedAssertion>
      <property>In excess instance:
1-1-10-1-7-1</property>
      <expected>false</expected>
      <actual>true</actual>
    </remoteFailedAssertion>
  </auditReport>
</auditResult>
</portResult>
<portResult address="1-1-2-4-1">
  <auditResult errorCode="PORT_NOT_CLEAN"/>

```

```

</portResult>
<portResult address="1-1-5-2-16">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-10-1-6">
  <auditResult errorCode="TEMPLATE_NOT_CONFIGURED">
    <serviceTemplateId>
      <templateName>20M20M</templateName>
      <templateVersion>1</templateVersion>
      <instanceLabel>card1-port1-service1</instanceLabel>
    </serviceTemplateId>
    <auditReport>
      <templateName>VZ_Ont</templateName>
      <templateType>G7342 ONT</templateType>
      <configured>true</configured>
      <remoteFailedAssertion>
        <property>Suspended</property>
        <expected>false</expected>
        <actual>true</actual>
      </remoteFailedAssertion>
    </auditReport>
    <auditReport>
      <templateName>VZ_Ethernet_Card</templateName>
      <templateType>G7342 ONT Card</templateType>
      <configured>true</configured>
    </auditReport>
    <auditReport>
      <templateName>VZ_Ethernet_Port</templateName>
      <templateType>G7342 ONT Ethernet Port</templateType>
      <configured>true</configured>
    </auditReport>
    <auditReport>
      <templateName>VZ_BridgePort</templateName>
      <templateType>G7342 Bridge Port</templateType>
      <configured>true</configured>
    </auditReport>
    <auditReport>
      <templateName>VZ_HSI_20M20M</templateName>
      <templateType>G7342 HSI Service</templateType>
      <configured>true</configured>
    </auditReport>
  </auditResult>
</portResult>
<portResult address="1-1-5-2-17">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-10-1-5">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-2-2-1">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-10-1-4">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-10-1-3">
  <auditResult>
    <serviceTemplateId>
      <templateName>20M20M</templateName>

```

```

        <templateVersion>1</templateVersion>
        <instanceLabel>no.2</instanceLabel>
    </serviceTemplateId>
    <auditReport>
        <templateName>VZ_Ont</templateName>
        <templateType>G7342 ONT</templateType>
        <configured>true</configured>
    </auditReport>
    <auditReport>
        <templateName>VZ_Ethernet_Card</templateName>
        <templateType>G7342 ONT Card</templateType>
        <configured>true</configured>
    </auditReport>
    <auditReport>
        <templateName>VZ_Ethernet_Port</templateName>
        <templateType>G7342 ONT Ethernet Port</templateType>
        <configured>true</configured>
    </auditReport>
    <auditReport>
        <templateName>VZ_BridgePort</templateName>
        <templateType>G7342 Bridge Port</templateType>
        <configured>true</configured>
    </auditReport>
    <auditReport>
        <templateName>VZ_HSI_20M20M</templateName>
        <templateType>G7342 HSI Service</templateType>
        <configured>true</configured>
    </auditReport>
</auditResult>
<auditResult>
    <serviceTemplateId>
        <templateName>10M10M</templateName>
        <templateVersion>1</templateVersion>
        <instanceLabel>card1-port1-service1</instanceLabel>
    </serviceTemplateId>
    <auditReport>
        <templateName>VZ_Ont</templateName>
        <templateType>G7342 ONT</templateType>
        <configured>true</configured>
    </auditReport>
    <auditReport>
        <templateName>VZ_Ethernet_Card</templateName>
        <templateType>G7342 ONT Card</templateType>
        <configured>true</configured>
    </auditReport>
    <auditReport>
        <templateName>VZ_Ethernet_Port</templateName>
        <templateType>G7342 ONT Ethernet Port</templateType>
        <configured>true</configured>
    </auditReport>
    <auditReport>
        <templateName>VZ_BridgePort</templateName>
        <templateType>G7342 Bridge Port</templateType>
        <configured>true</configured>
    </auditReport>
    <auditReport>
        <templateName>VZ_HSI_10M10M</templateName>
        <templateType>G7342 HSI Service</templateType>
        <configured>true</configured>
    </auditReport>

```

```

        </auditReport>
    </auditResult>
</portResult>
<portResult address="1-1-6-1-4">
    <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-10-1-2">
    <auditResult errorCode="TEMPLATE_NOT_CONFIGURED">
        <serviceTemplateId>
            <templateName>10M10M</templateName>
            <templateVersion>1</templateVersion>
            <instanceLabel>card1-port1-service1</instanceLabel>
        </serviceTemplateId>
        <auditReport>
            <templateName>VZ_Ont</templateName>
            <templateType>G7342 ONT</templateType>
            <configured>true</configured>
            <remoteFailedAssertion>
                <property>Suspended</property>
                <expected>false</expected>
                <actual>true</actual>
            </remoteFailedAssertion>
        </auditReport>
    </auditResult>
    <auditReport>
        <templateName>VZ_Ethernet_Card</templateName>
        <templateType>G7342 ONT Card</templateType>
        <configured>true</configured>
    </auditReport>
    <auditReport>
        <templateName>VZ_Ethernet_Port</templateName>
        <templateType>G7342 ONT Ethernet Port</templateType>
        <configured>true</configured>
    </auditReport>
    <auditReport>
        <templateName>VZ_BridgePort</templateName>
        <templateType>G7342 Bridge Port</templateType>
        <configured>true</configured>
    </auditReport>
    <auditReport>
        <templateName>VZ_HSI_10M10M</templateName>
        <templateType>G7342 HSI Service</templateType>
        <configured>true</configured>
    </auditReport>
    <auditReport>
        <templateName/>
        <templateType>apc.g7342.ont.card</templateType>
        <configured>true</configured>
        <remoteFailedAssertion>
            <property>In excess instance:
1-1-10-1-2-2</property>
            <expected>false</expected>
            <actual>true</actual>
        </remoteFailedAssertion>
    </auditReport>
    </auditResult>
</portResult>
<portResult address="1-1-6-1-3">
    <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>

```

```

<portResult address="1-1-6-3-1">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-10-1-1">
  <auditResult errorCode="TEMPLATE_NOT_CONFIGURED">
    <serviceTemplateId>
      <templateName>10M10M</templateName>
      <templateVersion>1</templateVersion>
      <instanceLabel>card1-port1-service1</instanceLabel>
    </serviceTemplateId>
    <auditReport>
      <templateName>VZ_Ont</templateName>
      <templateType>G7342 ONT</templateType>
      <configured>true</configured>
      <remoteFailedAssertion>
        <property>Suspended</property>
        <expected>false</expected>
        <actual>true</actual>
      </remoteFailedAssertion>
    </auditReport>
    <auditReport>
      <templateName>VZ_Ethernet_Card</templateName>
      <templateType>G7342 ONT Card</templateType>
      <configured>true</configured>
    </auditReport>
    <auditReport>
      <templateName>VZ_Ethernet_Port</templateName>
      <templateType>G7342 ONT Ethernet Port</templateType>
      <configured>true</configured>
    </auditReport>
    <auditReport>
      <templateName>VZ_BridgePort</templateName>
      <templateType>G7342 Bridge Port</templateType>
      <configured>true</configured>
    </auditReport>
    <auditReport>
      <templateName>VZ_HSI_10M10M</templateName>
      <templateType>G7342 HSI Service</templateType>
      <configured>true</configured>
    </auditReport>
  </auditResult>
</portResult>
<portResult address="1-1-3-2-1">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-3-4-1">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-8-1-4">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-8-1-3">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-8-3-1">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
<portResult address="1-1-8-1-2">
  <auditResult errorCode="PORT_NOT_CLEAN"/>

```

```
        </portResult>
        <portResult address="1-1-8-1-1">
          <auditResult errorCode="PORT_NOT_CLEAN"/>
        </portResult>
        <portResult address="1-1-8-1-5">
          <auditResult errorCode="PORT_NOT_CLEAN"/>
        </portResult>
      </commandResult>
    </ns:getNodeAuditResultResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

7.2.7 getTemplateMetaData operation response

The following is an example of a getTemplateMetaData operation response.

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:getTemplateMetaDataResponse xmlns:ns="uri://alcatel.com/apc/9.6">
      <templateMetaInfo>
        <templateName>24SIP_FTP</templateName>
        <type>apc.stack.type.label</type>
        <version>1</version>
        <state>RELEASED</state>
        <instanceLabelUsePermitted>true</instanceLabelUsePermitted>
        <attributesList>
          <attributeMetaData>
            <nameAndStringValue>
              <name>attributeModifiable</name>
              <value>YES</value>
            </nameAndStringValue>
            <nameAndStringValue>
              <name>attributeMinValue</name>
              <value/>
            </nameAndStringValue>
            <nameAndStringValue>
              <name>attributeValidValues</name>
              <value/>
            </nameAndStringValue>
            <nameAndStringValue>
              <name>attributeRequired</name>
              <value>OPTIONAL</value>
            </nameAndStringValue>
            <nameAndStringValue>
              <name>attributeGroupName</name>
              <value>spfenoshow</value>
            </nameAndStringValue>
            <nameAndStringValue>
              <name>attributeDataType</name>
              <value>STRING</value>
            </nameAndStringValue>
            <nameAndStringValue>
              <name>attributeRelevantPortType</name>
              <value>ontPotsPort</value>
            </nameAndStringValue>
          </attributeMetaData>
        </attributesList>
      </templateMetaInfo>
    </ns:getTemplateMetaDataResponse>
  </soapenv:Body>
</soapenv:Envelope>
```



```

</nameAndStringValue>
<nameAndStringValue>
  <name>attributeProperties</name>
  <value>ARGUMENTNATIVE</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeRelevantCardType</name>
  <value>EthernetPotsComboPotsVdsl2PotsCombo</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeIsInstanceArgument</name>
  <value>NO</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeName</name>
  <value>ontSubscriberLocationId</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeLength</name>
  <value>12</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeDefaultValue</name>
  <value>WILDCARD</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeMaxValue</name>
  <value/>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeSpfeAlias</name>
  <value>Subscriber Location ID</value>
</nameAndStringValue>
</attributeMetaData>
<attributeMetaData>
  <nameAndStringValue>
    <name>attributeModifiable</name>
    <value>YES</value>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeMinValue</name>
    <value/>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeValidValues</name>
    <value>GSFUGMDUGSBUNone</value>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeRequired</name>
    <value>OPTIONAL</value>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeDataType</name>
    <value>ENUM</value>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeRelevantPortType</name>
    <value>ontPotsPort</value>
  </nameAndStringValue>

```

```

<nameAndStringValue>
  <name>attributeProperties</name>
  <value>ARGUMENTNATIVE</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeRelevantCardType</name>
  <value>EthernetPotsComboPotsVdsl2PotsCombo</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeIsInstanceArgument</name>
  <value>NO</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeName</name>
  <value>ontType</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeExcludeInformation</name>
  <value>excludeFromAudit</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeLength</name>
  <value/>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeDefaultValue</name>
  <value>GMDU</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeMaxValue</name>
  <value/>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeSpfeAlias</name>
  <value>ONT Type</value>
</nameAndStringValue>
</attributeMetaData>
<attributeMetaData>
  <nameAndStringValue>
    <name>attributeModifiable</name>
    <value>NO</value>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeMinValue</name>
    <value/>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeValidValues</name>
    <value/>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeRequired</name>
    <value>MANDATORY</value>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeGroupName</name>
    <value>spfenoshow</value>
  </nameAndStringValue>
  <nameAndStringValue>

```

```

        <name>attributeDataType</name>
        <value>STRING</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeRelevantPortType</name>
        <value>ontPotsPort</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeProperties</name>
        <value>ARGUMENTNATIVE</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeRelevantCardType</name>
        <value>EthernetPotsComboPotsVdsl2PotsCombo</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeIsInstanceArgument</name>
        <value>NO</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeName</name>
        <value>config_Client_IpAddress</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeLength</name>
        <value>15</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeDefaultValue</name>
        <value>0.0.0.0</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeMaxValue</name>
        <value/>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeSpfeAlias</name>
        <value>Client IP Address</value>
    </nameAndStringValue>
</attributeMetaData>
<attributeMetaData>
    <nameAndStringValue>
        <name>attributeModifiable</name>
        <value>YES</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeMinValue</name>
        <value/>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeValidValues</name>
        <value/>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeRequired</name>
        <value>OPTIONAL</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeGroupName</name>

```

```

        <value>Customer Information</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeDataType</name>
        <value>STRING</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeRelevantPortType</name>
        <value>ontPotsPort</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeProperties</name>
        <value>ARGUMENTNATIVE</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeRelevantCardType</name>
        <value>EthernetPotsComboPotsVdsl2PotsCombo</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeIsInstanceArgument</name>
        <value>NO</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeName</name>
        <value>dhcp_option61ClientId</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeLength</name>
        <value>25</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeDefaultValue</name>
        <value/>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeMaxValue</name>
        <value/>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeSpfeAlias</name>
        <value>Option 61 Client ID</value>
    </nameAndStringValue>
</attributeMetaData>
<attributeMetaData>
    <nameAndStringValue>
        <name>attributeModifiable</name>
        <value>YES</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeMinValue</name>
        <value/>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeValidValues</name>
        <value/>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeRequired</name>
        <value>OPTIONAL</value>

```

```

</nameAndStringValue>
<nameAndStringValue>
  <name>attributeGroupName</name>
  <value>spfenoshow</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeDataType</name>
  <value>STRING</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeRelevantPortType</name>
  <value>ontPotsPort</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeProperties</name>
  <value>ARGUMENTNATIVE</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeRelevantCardType</name>
  <value>EthernetPotsComboPotsVdsl2PotsCombo</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeIsInstanceArgument</name>
  <value>NO</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeName</name>
  <value>dhcp_option82CustomerId</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeLength</name>
  <value>64</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeDefaultValue</name>
  <value/>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeMaxValue</name>
  <value/>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeSpfeAlias</name>
  <value>Option 82 Customer ID</value>
</nameAndStringValue>
</attributeMetaData>
(...)
<attributeMetaData>
  <nameAndStringValue>
    <name>attributeModifiable</name>
    <value>NO</value>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeMinValue</name>
    <value>1</value>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeValidValues</name>
    <value/>
  </nameAndStringValue>

```

```

</nameAndStringValue>
<nameAndStringValue>
  <name>attributeRequired</name>
  <value>MANDATORY</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeGroupName</name>
  <value>spfenoshow</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeDataType</name>
  <value>INTEGER</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeRelevantPortType</name>
  <value>ontPotsPort</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeProperties</name>
  <value>ARGUMENTNATIVE</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeRelevantCardType</name>
  <value>EthernetPotsComboPotsVdsl2PotsCombo</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeIsInstanceArgument</name>
  <value>YES</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeName</name>
  <value>voipServiceId</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeLength</name>
  <value/>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeDefaultValue</name>
  <value>1</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeMaxValue</name>
  <value>1</value>
</nameAndStringValue>
<nameAndStringValue>
  <name>attributeSpfeAlias</name>
  <value>VoIP Service ID</value>
</nameAndStringValue>
</attributeMetaData>
<attributeMetaData>
  <nameAndStringValue>
    <name>attributeModifiable</name>
    <value>YES</value>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeMinValue</name>
    <value>0</value>
  </nameAndStringValue>

```

```

    <nameAndStringValue>
      <name>attributeValidValues</name>
      <value/>
    </nameAndStringValue>
    <nameAndStringValue>
      <name>attributeRequired</name>
      <value>OPTIONAL</value>
    </nameAndStringValue>
    <nameAndStringValue>
      <name>attributeGroupName</name>
      <value>VLAN Information</value>
    </nameAndStringValue>
    <nameAndStringValue>
      <name>attributeDataType</name>
      <value>LONG</value>
    </nameAndStringValue>
    <nameAndStringValue>
      <name>attributeRelevantPortType</name>
      <value>ontPotsPort</value>
    </nameAndStringValue>
    <nameAndStringValue>
      <name>attributeProperties</name>
      <value>ARGUMENTNATIVE</value>
    </nameAndStringValue>
    <nameAndStringValue>
      <name>attributeRelevantCardType</name>
      <value>EthernetPotsComboPotsVdsl2PotsCombo</value>
    </nameAndStringValue>
    <nameAndStringValue>
      <name>attributeIsInstanceArgument</name>
      <value>YES</value>
    </nameAndStringValue>
    <nameAndStringValue>
      <name>attributeName</name>
      <value>voipStaticCVlanID</value>
    </nameAndStringValue>
    <nameAndStringValue>
      <name>attributeLength</name>
      <value/>
    </nameAndStringValue>
    <nameAndStringValue>
      <name>attributeDefaultValue</name>
      <value>0</value>
    </nameAndStringValue>
    <nameAndStringValue>
      <name>attributeMaxValue</name>
      <value>4095</value>
    </nameAndStringValue>
    <nameAndStringValue>
      <name>attributeSpfeAlias</name>
      <value>Static C-VLAN ID</value>
    </nameAndStringValue>
  </attributeMetaData>
  <attributeMetaData>
    <nameAndStringValue>
      <name>attributeModifiable</name>
      <value>NO</value>
    </nameAndStringValue>
    <nameAndStringValue>

```

```

        <name>attributeMinValue</name>
        <value>0</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeValidValues</name>
        <value/>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeRequired</name>
        <value>MANDATORY</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeGroupName</name>
        <value>VLAN Information</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeDataType</name>
        <value>LONG</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeRelevantPortType</name>
        <value>ontPotsPort</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeProperties</name>
        <value>ARGUMENTNATIVE</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeRelevantCardType</name>
        <value>EthernetPotsComboPotsVdsl2PotsCombo</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeIsInstanceArgument</name>
        <value>NO</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeName</name>
        <value>voipVlanId</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeLength</name>
        <value/>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeDefaultValue</name>
        <value>0</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeMaxValue</name>
        <value>4093</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeSpfeAlias</name>
        <value>S-VLAN ID</value>
    </nameAndStringValue>
</attributeMetaData>
<attributeMetaData>
    <nameAndStringValue>
        <name>attributeModifiable</name>

```



```

        <value>NO</value>
      </nameAndStringValue>
      <nameAndStringValue>
        <name>attributeMinValue</name>
        <value>1</value>
      </nameAndStringValue>
      <nameAndStringValue>
        <name>attributeValidValues</name>
        <value/>
      </nameAndStringValue>
      <nameAndStringValue>
        <name>attributeRequired</name>
        <value>MANDATORY</value>
      </nameAndStringValue>
      <nameAndStringValue>
        <name>attributeDataType</name>
        <value>INTEGER</value>
      </nameAndStringValue>
      <nameAndStringValue>
        <name>attributeRelevantPortType</name>
        <value>ontPotsPort</value>
      </nameAndStringValue>
      <nameAndStringValue>
        <name>attributeProperties</name>
        <value>ARGUMENTNATIVE</value>
      </nameAndStringValue>
      <nameAndStringValue>
        <name>attributeRelevantCardType</name>
        <value>Vdsl2PotsCombo</value>
      </nameAndStringValue>
      <nameAndStringValue>
        <name>attributeIsInstanceArgument</name>
        <value>YES</value>
      </nameAndStringValue>
      <nameAndStringValue>
        <name>attributeName</name>
        <value>ontCardId</value>
      </nameAndStringValue>
      <nameAndStringValue>
        <name>attributeLength</name>
        <value/>
      </nameAndStringValue>
      <nameAndStringValue>
        <name>attributeDefaultValue</name>
        <value>1</value>
      </nameAndStringValue>
      <nameAndStringValue>
        <name>attributeMaxValue</name>
        <value>15</value>
      </nameAndStringValue>
      <nameAndStringValue>
        <name>attributeSpfeAlias</name>
        <value>Card Number</value>
      </nameAndStringValue>
    </attributeMetaData>
  </attributeMetaData>
  <nameAndStringValue>
    <name>attributeModifiable</name>

```

(...)

```

        <value>YES</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeMinValue</name>
        <value>1</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeValidValues</name>
        <value/>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeRequired</name>
        <value>MANDATORY</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeGroupName</name>
        <value>spfenoshow</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeDataType</name>
        <value>LONG</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeRelevantPortType</name>
        <value>ontPotsPort</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeProperties</name>
        <value>ARGUMENTNATIVE</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeRelevantCardType</name>
        <value>EthernetPotsComboPotsVdsl2PotsCombo</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeIsInstanceArgument</name>
        <value>NO</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeName</name>
        <value>potsPortVoipServiceIndex</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeLength</name>
        <value/>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeDefaultValue</name>
        <value>1</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeMaxValue</name>
        <value>1</value>
    </nameAndStringValue>
    <nameAndStringValue>
        <name>attributeSpfeAlias</name>
        <value>VoIP Service Index</value>
    </nameAndStringValue>
</attributeMetaData>

```

```

<attributeMetaData>
  <nameAndStringValue>
    <name>attributeModifiable</name>
    <value>NO</value>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeMinValue</name>
    <value/>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeValidValues</name>
    <value>ontPotsPort</value>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeRequired</name>
    <value>OPTIONAL</value>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeDataType</name>
    <value>STRING</value>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeProperties</name>
    <value>ARGUMENTORSELECTOR</value>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeIsInstanceArgument</name>
    <value>NO</value>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeName</name>
    <value>portType</value>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeLength</name>
    <value/>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeDefaultValue</name>
    <value/>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeMaxValue</name>
    <value/>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeSpfeAlias</name>
    <value>Port Type</value>
  </nameAndStringValue>
</attributeMetaData>
<attributeMetaData>
  <nameAndStringValue>
    <name>attributeModifiable</name>
    <value>NO</value>
  </nameAndStringValue>
  <nameAndStringValue>
    <name>attributeMinValue</name>
    <value/>
  </nameAndStringValue>

```

```

        <nameAndStringValue>
          <name>attributeValidValues</name>
          <value>EthernetPotsComboPotsVdsl2PotsCombo</value>
        </nameAndStringValue>
        <nameAndStringValue>
          <name>attributeRequired</name>
          <value>OPTIONAL</value>
        </nameAndStringValue>
        <nameAndStringValue>
          <name>attributeDataType</name>
          <value>STRING</value>
        </nameAndStringValue>
        <nameAndStringValue>
          <name>attributeProperties</name>
          <value>ARGUMENTORSELECTOR</value>
        </nameAndStringValue>
        <nameAndStringValue>
          <name>attributeIsInstanceArgument</name>
          <value>NO</value>
        </nameAndStringValue>
        <nameAndStringValue>
          <name>attributeName</name>
          <value>cardType</value>
        </nameAndStringValue>
        <nameAndStringValue>
          <name>attributeLength</name>
          <value/>
        </nameAndStringValue>
        <nameAndStringValue>
          <name>attributeDefaultValue</name>
          <value/>
        </nameAndStringValue>
        <nameAndStringValue>
          <name>attributeMaxValue</name>
          <value/>
        </nameAndStringValue>
        <nameAndStringValue>
          <name>attributeSpfeAlias</name>
          <value>Card Type</value>
        </nameAndStringValue>
      </attributeMetaData>
    </attributesList>
    <attributeGroupMetaData/>
  </templateMetaInfo>
</ns:getTemplateMetaDataResponse>
</soapenv:Body>
</soapenv:Envelope>

```

7.3 Examples of imported bulk request files

This section contains examples of imported bulk request files.

7.3.1 Example imported bulk request file for template migration

The following is an example of the file format of a bulk request that is imported during template migration.

```
<commands>
  <command>
    <commandId>1</commandId>
    <action>MIGRATE</action>
    <objectName>ISAM190:1-1-3-1</objectName>
    <templateName>Migrate-2</templateName>
    <oldTemplateName>Migrate-1</oldTemplateName>
    <argument>
      <name>customerId</name>
      <value>secondMig</value>
    </argument>
    <argument>
      <name>spectrumProfileName</name>
      <value>xd_spec_1</value>
    </argument>
  </command>
  <command>
    <commandId>2</commandId>
    <action>MIGRATE</action>
    <objectName>ISAM190:1-1-2-1</objectName>
    <templateName>Migrate-2</templateName>
    <oldTemplateName>Migrate-1</oldTemplateName>
    <argument>
      <name>customerId</name>
      <value>secondMig</value>
    </argument>
    <argument>
      <name>spectrumProfileName</name>
      <value>xd_spec_1</value>
    </argument>
  </command>
</commands>
```

7.4 Examples of audit results

This section describes the different results for an audit operation.

- [Example of an audit result for a clean port](#)
- [Example of an audit result for a dirty port](#)
- [Example of an audit result for an aligned port](#)
- [Example of an audit result for a misaligned port](#)
- [Example of an audit result where a port is intended to be configured but remains unconfigured](#)

See section [4.3](#) for a description of other template-related and argument-related error codes that might occur in an audit result.

7.4.1 Example of an audit result for a clean port

The following is an example of an audit result that indicates that a port is clean. The port is not configured and the intended configuration for the service in the template matches the actual configuration in the NE.

```
<portResult address="1-1-2-10">
  <auditResult/>
</portResult>
```

7.4.2 Example of an audit result for a dirty port

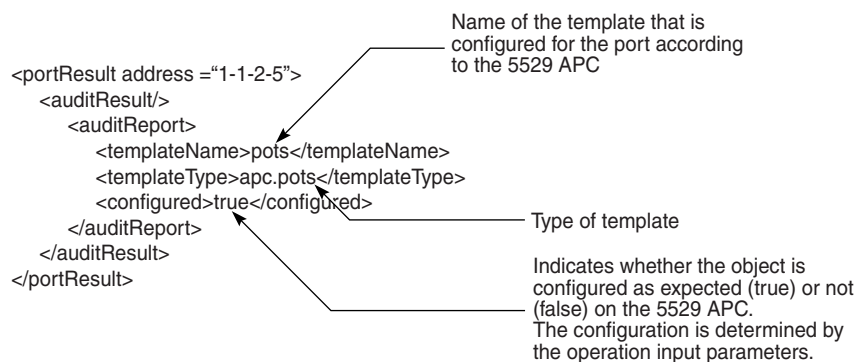
The following is an example of an audit result that indicates that a port is dirty. The port is configured on the NE but the configuration was not made by the 5529 APC or OSS.

```
<portResult address="1-1-2-2">
  <auditResult errorCode="PORT_NOT_CLEAN"/>
</portResult>
```

7.4.3 Example of an audit result for an aligned port

Figure 3 shows an example of an audit result that indicates that a port is aligned. The port is configured and the intended configuration for the service in the template matches the actual configuration in the NE.

Figure 3 Example of an audit result for an aligned port

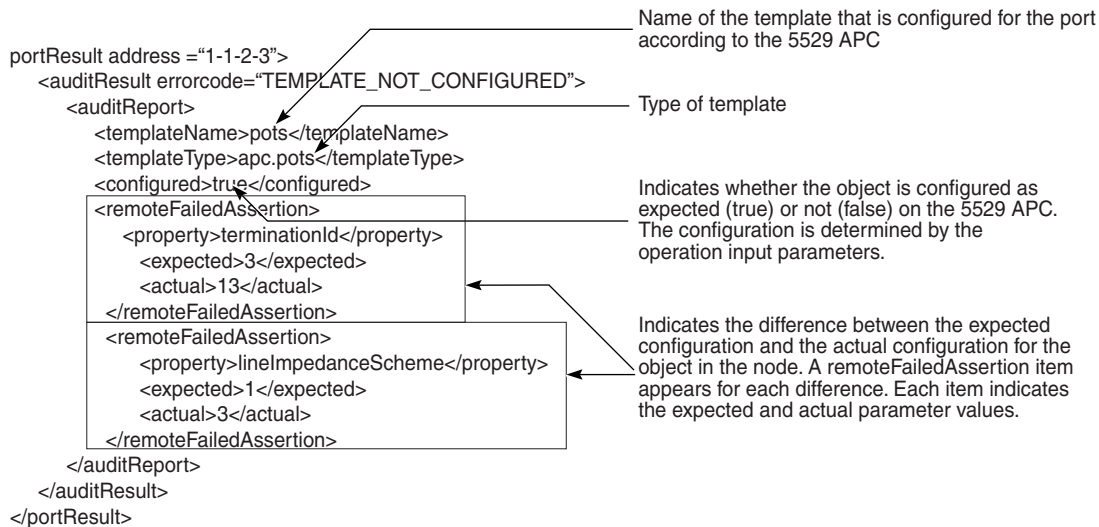


23836

7.4.4 Example of an audit result for a misaligned port

Figure 4 shows an example of an audit result that indicates that a port is misaligned. The port is configured and the intended configuration for the service in the template does not match the actual configuration in the NE.

Figure 4 Example of an audit result for a misaligned port



20027

7.4.5 Example of an audit result where a port is intended to be configured but remains unconfigured

The following is an example of an audit result that indicates that a template was used in an operation for the port, but the template is not configured on the port. The information in the 5529 APC indicates that the port is configured but the NE indicates that the port is not configured.

```
<portResult address="1-1-2-3">
  <auditResult errorCode="TEMPLATE_NOT_CONFIGURED"/>
</portResult>
```


8 Configuration change notification message and web service examples

8.1 Examples of change notification messages

8.2 Example of a getNotificationTopic operation request and response

8.1 Examples of change notification messages

The following is an example of a change notification message for a configure operation.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <tmf854:header xmlns:tmf854="tmf854.v1" tmf854Version="1.0">
      <tmf854:communicationPattern>Notification</tmf854:communication
Pattern>
      <tmf854:msgType>NOTIFICATION</tmf854:msgType>
      <tmf854:destinationURI>apc/ng/XTNnotifGroupHWJA</tmf854:destina
tionURI>
      <tmf854:activityName>notify</tmf854:activityName>
      <tmf854:msgName>notify</tmf854:msgName>
      <tmf854:communicationStyle>MSG</tmf854:communicationStyle>
      <tmf854:senderURI>192.168.92.123</tmf854:senderURI>
    </tmf854:header>
  </soapenv:Header>
  <soapenv:Body>
    <tmf854:notify xmlns:alu="alu.v1"
xmlns:apcType="uri://alcatel.com/apc/9.2"
xmlns:ns1="http://xml.apache.org/axis/" xmlns:tmf854="tmf854.v1"
extAuthor="alu" tmf854Version="1.0">
      <tmf854:topic>apc/ng/XTNnotifGroupHWJA</tmf854:topic>
      <tmf854:message>
        <tmf854:VendorNotification>
          <tmf854:notificationId>166</tmf854:notificationId>
          <tmf854:vendorNotificationType>ApcServiceProvisioningNo
tification</tmf854:vendorNotificationType>
          <tmf854:vendorExtensions>
            <alu:ApcServiceProvisioningNotification>
              <operationDate>2017/02/20</operationDate>
              <operationTime>15:19:00</operationTime>
              <operationMode>Automated</operationMode>
              <operationKind>Service
Provisioning</operationKind>
              <clientName>nbi</clientName>
              <operationName>configure</operationName>
              <apcRequest>
```

```

/objectName>
<apcType:configure>
  <objectName>135.249.41.119:1-1-10-1-11<
    <templateName>20M20M</templateName>
    <templateVersion>1</templateVersion>
    <instanceLabel>no.3</instanceLabel>
    <argument>
      <name>ethPortId</name>
      <value>1</value>
    </argument>
    <argument>
      <name>ontCardId</name>
      <value>2</value>
    </argument>
    <argument>
      <name>portType</name>
      <value>ontEthernetPort</value>
    </argument>
    <argument>
      <name>cvlanId</name>
      <value>1002</value>
    </argument>
    <argument>
      <name>cardType</name>
      <value>Ethernet</value>
    </argument>
    <argument>
      <name>ontType</name>
      <value>GMDU</value>
    </argument>
    <argument>
      <name>hsiVlanId</name>
      <value>100</value>
    </argument>
    <operationInitiator>Admin</operationIni
tiator>
  </apcType:configure>
</apcRequest>
<apcResponse>
  <apcType:configureResponse/>
</apcResponse>
</alu:ApcServiceProvisioningNotification>
</tmf854:vendorExtensions>
</tmf854:VendorNotification>
</tmf854:message>
</tmf854:notify>
</soapenv:Body>
</soapenv:Envelope>

```

The following is an example of a change notification message for a clean operation.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <tmf854:header xmlns:tmf854="tmf854.v1" tmf854Version="1.0">
      <tmf854:communicationPattern>Notification</tmf854:communication
Pattern>
      <tmf854:msgType>NOTIFICATION</tmf854:msgType>

```

```

tionURI>
    <tmf854:destinationURI>apc/ng/XTNnotifGroupHWJA</tmf854:destina
    <tmf854:activityName>notify</tmf854:activityName>
    <tmf854:msgName>notify</tmf854:msgName>
    <tmf854:communicationStyle>MSG</tmf854:communicationStyle>
    <tmf854:senderURI>192.168.92.123</tmf854:senderURI>
  </tmf854:header>
</soapenv:Header>
<soapenv:Body>
  <tmf854:notify xmlns:alu="alu.v1"
xmlns:apcType="uri://alcatel.com/apc/9.2"
xmlns:ns1="http://xml.apache.org/axis/" xmlns:tmf854="tmf854.v1"
extAuthor="alu" tmf854Version="1.0">
    <tmf854:topic>apc/ng/XTNnotifGroupHWJA</tmf854:topic>
    <tmf854:message>
      <tmf854:VendorNotification>
        <tmf854:notificationId>167</tmf854:notificationId>
        <tmf854:vendorNotificationType>ApcServiceProvisioningNo
tification</tmf854:vendorNotificationType>
        <tmf854:vendorExtensions>
          <alu:ApcServiceProvisioningNotification>
            <operationDate>2017/02/20</operationDate>
            <operationTime>15:20:42</operationTime>
            <operationMode>Automated</operationMode>
            <operationKind>Service
Provisioning</operationKind>
            <clientName>nbi</clientName>
            <operationName>clean</operationName>
            <apcRequest>
              <apcType:clean>
                <objectName>135.249.41.119:1-1-10-1-11<
/objectName>
                <operationInitiator>Admin</operationIni
tiator>
              </apcType:clean>
            </apcRequest>
            <apcResponse>
              <apcType:cleanResponse/>
            </apcResponse>
          </alu:ApcServiceProvisioningNotification>
        </tmf854:vendorExtensions>
      </tmf854:VendorNotification>
    </tmf854:message>
  </tmf854:notify>
</soapenv:Body>
</soapenv:Envelope>

```

The following is an example of change notification for a heartbeat message.

```

<soapenv:Envelope xmlns="tmf854.v1"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <header>
      <activityName>notify</activityName>
      <msgName>notify</msgName>
      <msgType>NOTIFICATION</msgType>
      <senderURI>http://192.168.92.121</senderURI>
    </header>
  </soapenv:Header>
  <soapenv:Body>
    <tmf854:notify>
      <tmf854:topic>apc/ng/XTNnotifGroupHWJA</tmf854:topic>
      <tmf854:message>
        <tmf854:VendorNotification>
          <tmf854:notificationId>167</tmf854:notificationId>
          <tmf854:vendorNotificationType>ApcServiceProvisioningNo
tification</tmf854:vendorNotificationType>
          <tmf854:vendorExtensions>
            <alu:ApcServiceProvisioningNotification>
              <operationDate>2017/02/20</operationDate>
              <operationTime>15:20:42</operationTime>
              <operationMode>Automated</operationMode>
              <operationKind>Service
Provisioning</operationKind>
              <clientName>nbi</clientName>
              <operationName>clean</operationName>
              <apcRequest>
                <apcType:clean>
                  <objectName>135.249.41.119:1-1-10-1-11<
/objectName>
                  <operationInitiator>Admin</operationIni
tiator>
                </apcType:clean>
              </apcRequest>
              <apcResponse>
                <apcType:cleanResponse/>
              </apcResponse>
            </alu:ApcServiceProvisioningNotification>
          </tmf854:vendorExtensions>
        </tmf854:VendorNotification>
      </tmf854:message>
    </tmf854:notify>
  </soapenv:Body>
</soapenv:Envelope>

```

```

        <destinationURI>urn:topic/apc/ng/XTNnotifGroupHWJA</destinationURI>
        <communicationPattern>Notification</communicationPattern>
        <communicationStyle>MSG</communicationStyle>
        <timestamp>20170220152224.448+0700</timestamp>
    </header>
</soapenv:Header>
<soapenv:Body>
    <notify>
        <topic>topic/apc/ng/XTNnotifGroupHWJA</topic>
        <message>
            <Heartbeat>
                <notificationId>APC:7118</notificationId>
                <objectType>OT_OS</objectType>
                <objectName>
                    <osNm>Alcatel-Lucent/APC</osNm>
                </objectName>
                <osTime>20170220152224.448+0700</osTime>
            </Heartbeat>
        </message>
    </notify>
</soapenv:Body>
</soapenv:Envelope>

```

8.2 Example of a getNotificationTopic operation request and response

The following is an example of a getNotificationTopic operation request.

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:ns="uri://alcatel.com/apc/3.1">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:getNotificationTopic>
      <eventType>VendorNotification</eventType>
      <notificationType>ApcServiceProvisioningNotification</notification
Type>
    </ns:getNotificationTopic>
  </soapenv:Body>
</soapenv:Envelope>

```

The following is an example of a getNotificationTopic operation response. The response contains one JMS topic.

```

<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns:getNotificationTopicResponse
      xmlns:ns="uri://alcatel.com/apc/3.1">
      <topicName>topic/apc/ng/XTNnotifGroupHWJA</topicName>
    </ns:getNotificationTopicResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

9 New ONT notification

9.1 New ONT notifications

9.2 Example of a new ONT notification event

9.1 New ONT notifications

This chapter provides information about the new ONT notification event.

There are several end-to-end ONT service provisioning message flows. The most effective ones are the SLID based pre-provisioning (before ONT installation) and serial number (SERNUM) based pre-provisioning message flows. Some service providers, however, prefer the new ONT notification-triggered service provisioning flow. The event object name in the new ONT notification represents the PON port at which a new ONT has appeared.

Table 79 describes the new ONT notification event arguments.

Table 79 New ONT notification event arguments

Argument	Description
eventType	The type of notification event. The value is always newONT.
SLID	The ONT subscriber location ID
SERNUM	The ONT serial number
DISIND ⁽¹⁾	The ONT disable indicator. The supported values are: <ul style="list-style-type: none">DISABLED—indicates that a PLOAM message was sent to the ONT to turn off the ONT laserPWRERR—indicates that extraneous power bursts have been detected on the ONTDIFREACH—indicates that the ONT has been deactivated because the location of the ONT on the PON has exceeded its differential reach capabilitySNBUNDLE—indicates that the ONT failed to bundle the serial number with the LOID or SLIDnull string (no text)—indicates that none of the other DISIND values are applicable
LOID ⁽¹⁾	The logical ONU ID. The value is the logical authentication ID.

Note

⁽¹⁾ Argument is supported on 7342 ISAM FTTU and 7302/7330/7356/7360/7362/7363/7367 ISAM R5.2 or later.

9.1.1 Format of empty value element

The format of a value element with no value provided in the new ONT notification event is different between 5529 APC R9.5.0x and R9.6 or later. The R9.5.0x format uses two tags: <value> and </value>. The format for R9.6 or later uses one tag: <value/>.

Example of R9.5.0x format:

```
<argument>
  <name>LOID</name>
  <value></value>
</argument>
```

Example of format for R9.6 or later:

```
<argument>
  <name>LOID</name>
  <value/>
</argument>
```

9.2 Example of a new ONT notification event

The following is an example of the alarm for new ONTs.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <tmf854:header xmlns:tmf854="tmf854.v1" tmf854Version="1.0">
      <tmf854:communicationPattern>Notification</tmf854:communication
Pattern>
      <tmf854:msgType>NOTIFICATION</tmf854:msgType>
      <tmf854:destinationURI>apc/ng/MABnotifGroupTVNX</tmf854:destina
tionURI>
      <tmf854:activityName>notify</tmf854:activityName>
      <tmf854:msgName>notify</tmf854:msgName>
      <tmf854:communicationStyle>MSG</tmf854:communicationStyle>
      <tmf854:senderURI>192.168.95.151</tmf854:senderURI>
    </tmf854:header>
  </soapenv:Header>
  <soapenv:Body>
    <tmf854:notify xmlns:alu="alu.v1"
      xmlns:apcType="uri://alcatel.com/apc/9.2"
      xmlns:ns1="http://xml.apache.org/axis/" xmlns:tmf854="tmf854.v1"
      extAuthor="alu" tmf854Version="1.0">
      <tmf854:topic>apc/ng/MABnotifGroupTVNX</tmf854:topic>
      <tmf854:message>
        <tmf854:VendorNotification>
          <tmf854:notificationId>2</tmf854:notificationId>
          <tmf854:vendorNotificationType>ApcServiceProvisioningNo
tification</tmf854:vendorNotificationType>
          <tmf854:vendorExtensions>
            <alu:ApcServiceProvisioningNotification>
              <operationDate>2017/03/15</operationDate>
```

```

        <operationTime>14:47:00</operationTime>
        <operationMode>Automated</operationMode>
        <operationKind>Service
Provisioning</operationKind>
        <clientName>notifConsumer</clientName>
        <apcEvent>
entObjectName>        <eventObjectName>135.249.41.119:1-1-4-1</ev
        <eventInformation>
            <argument>
                <name>eventType</name>
                <value>newONT</value>
            </argument>
            <argument>
                <name>SLID</name>
                <value>DEFAULT</value>
            </argument>
            <argument>
                <name>SERNUM</name>
                <value>ALCLFADA ECC4</value>
            </argument>
            <argument>
                <name>LOID</name>
                <value/>
            </argument>
        </eventInformation>
        </apcEvent>
    </alu:ApcServiceProvisioningNotification>
</tmf854:vendorExtensions>
</tmf854:VendorNotification>
</tmf854:message>
</tmf854:notify>
</soapenv:Body>
</soapenv:Envelope>

```


10 NE reparenting notifications

10.1 NE reparenting notifications

10.2 Examples of NE reparenting notifications

10.1 NE reparenting notifications

The 5529 APC sends notifications of the following events to the OSS:

- Removal of NE services as part of an NE reparenting operation.
- Addition of NE services as part of an NE reparenting operation.



Note — NE reparenting notifications are triggered when the `apclImportExportServices.sh` script is run. These notifications are not authentic NBI operations.

10.2 Examples of NE reparenting notifications

The following is an example of a notification for an NE reparenting export operation:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Header>
    <tmf854:header xmlns:tmf854="tmf854.v1" tmf854Version="1.0">
      <tmf854:communicationPattern>Notification</tmf854:communication
Pattern>
      <tmf854:msgType>NOTIFICATION</tmf854:msgType>
      <tmf854:destinationURI>apc/ng/MABnotifGroupTVNX</tmf854:destina
tionURI>
      <tmf854:activityName>notify</tmf854:activityName>
      <tmf854:msgName>notify</tmf854:msgName>
      <tmf854:communicationStyle>MSG</tmf854:communicationStyle>
      <tmf854:senderURI>192.168.95.151</tmf854:senderURI>
    </tmf854:header>
  </soapenv:Header>
  <soapenv:Body>
    <tmf854:notify xmlns:alu="alu.v1"
xmlns:apcType="uri://alcatel.com/apc/9.2"
xmlns:ns1="http://xml.apache.org/axis/" xmlns:tmf854="tmf854.v1"
extAuthor="alu" tmf854Version="1.0">
      <tmf854:topic>apc/ng/MABnotifGroupTVNX</tmf854:topic>
      <tmf854:message>
        <tmf854:VendorNotification>
          <tmf854:notificationId>1</tmf854:notificationId>
        </tmf854:VendorNotification>
      </tmf854:message>
    </tmf854:notify>
  </soapenv:Body>
</soapenv:Envelope>
```

```

        <tmf854:vendorNotificationType>ApcServiceProvisioningNo
tification</tmf854:vendorNotificationType>
        <tmf854:vendorExtensions>
            <alu:ApcServiceProvisioningNotification>
                <operationDate>2017/03/15</operationDate>
                <operationTime>15:19:33</operationTime>
                <operationMode>Automated</operationMode>
                <operationKind>Service
Provisioning</operationKind>
                <clientName>system</clientName>
                <operationName>removeServices</operationName>
                <apcRequest>
                    <apcType:removeServices>
                        <m_sourceFile>/var/opt/ams/local/reposi
tory/app-apc-1.0_9.6.03-354177/bin/apcImportExportServices/.apcExportedSer
vices.zip</m_sourceFile>
                        <m_neNames>GPON_4157</m_neNames>
                        <objectName>GPON_4157</objectName>
                    </apcType:removeServices>
                </apcRequest>
                <apcResponse>
                    <apcType:removeServicesResponse/>
                </apcResponse>
            </alu:ApcServiceProvisioningNotification>
        </tmf854:vendorExtensions>
    </tmf854:VendorNotification>
</tmf854:message>
</tmf854:notify>
</soapenv:Body>
</soapenv:Envelope>

```

The following is an example of a notification for an NE reparenting import operation:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Header>
        <tmf854:header xmlns:tmf854="tmf854.v1" tmf854Version="1.0">
            <tmf854:communicationPattern>Notification</tmf854:communication
Pattern>
            <tmf854:msgType>NOTIFICATION</tmf854:msgType>
            <tmf854:destinationURI>apc/ng/MABnotifGroupTVNX</tmf854:destina
tionURI>
            <tmf854:activityName>notify</tmf854:activityName>
            <tmf854:msgName>notify</tmf854:msgName>
            <tmf854:communicationStyle>MSG</tmf854:communicationStyle>
            <tmf854:senderURI>192.168.95.151</tmf854:senderURI>
        </tmf854:header>
    </soapenv:Header>
    <soapenv:Body>
        <tmf854:notify xmlns:alu="alu.v1"
xmlns:apcType="uri://alcatel.com/apc/9.2"
xmlns:ns1="http://xml.apache.org/axis/" xmlns:tmf854="tmf854.v1"
extAuthor="alu" tmf854Version="1.0">
            <tmf854:topic>apc/ng/MABnotifGroupTVNX</tmf854:topic>
            <tmf854:message>
                <tmf854:VendorNotification>
                    <tmf854:notificationId>3</tmf854:notificationId>

```

```

        <tmf854:vendorNotificationType>ApcServiceProvisioningNo
tification</tmf854:vendorNotificationType>
        <tmf854:vendorExtensions>
            <alu:ApcServiceProvisioningNotification>
                <operationDate>2017/03/15</operationDate>
                <operationTime>15:28:42</operationTime>
                <operationMode>Automated</operationMode>
                <operationKind>Service
Provisioning</operationKind>
                <clientName>system</clientName>
                <operationName>importServices</operationName>
                <apcRequest>
                    <apcType:importServices>
                        <sourceFile>/var/opt/ams/local/reposito
ry/app-apc-1.0_9.6.03-354177/bin/apcImportExportServices/apcExportedService
s.zip</sourceFile>
                        </apcType:importServices>
                    </apcRequest>
                <apcResponse>
                    <apcType:importServicesResponse/>
                </apcResponse>
            </alu:ApcServiceProvisioningNotification>
        </tmf854:vendorExtensions>
    </tmf854:VendorNotification>
</tmf854:message>
</tmf854:notify>
</soapenv:Body>
</soapenv:Envelope>

```

In the 5529 APC, R9.4 and later, the format of the XML output file that is generated by the apcImportExportServices tool is changed to remove whitespace and looks like the following example:

```

<exec:argument><exec:name>ontSubscriberLocationId</exec:name><exec:value>WI
LDCARD</exec:value></exec:argument><exec:argument><exec:name>ontType</exec:
name><exec:value>GSBU</exec:value></exec:argument>

```

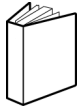
The old format looks like the following example:

```

<exec:argument>
    <exec:name>ontSubscriberLocationId</exec:name>
    <exec:value>WILDCARD</exec:value>
</exec:argument>
<exec:argument>
    <exec:name>ontType</exec:name>
    <exec:value>GSBU</exec:value>
</exec:argument>

```


Customer document and product support



Customer documentation

[Customer Documentation Welcome Page](#)



Technical Support

[Product Support Portal](#)



Documentation feedback

[Customer Documentation Feedback](#)

