Subscription Repository

User Guide

v.7.0

Legal Notices

Warranty.
Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.
A copy of the specific warranty terms applicable to your Hewlett-Packard product can be obtained from your local Sales and Service Office.
Restricted Rights Legend.
Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.
Hewlett-Packard Company United States of America
Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).
Copyright Notices.

Trademark Notices.
Java™ is a registered trademark of Oracle and/or its affiliates.
Linux is a U.S. registered trademark of Linus Torvalds
Microsoft® is a U.S. registered trademark of Microsoft Corporation.
Oracle® is a registered U.S. trademark of Oracle Corporation, Redwood City, California.
UNIX® is a registered trademark of the Open Group.
Windows® and MS Windows® are U.S. registered trademarks of Microsoft Corporation.
All other product names are the property of their respective trademark or service mark holders and are hereby acknowledged.

Table of Contents

In This Guide

This guide explains how to interact with the Subscription Repository (SR) through exposed web services.

Audience

The audience for this guide is the Solutions Integrator (SI). The SI has a combination of some or all of the following capabilities:

- Understands and has a solid working knowledge of:
  - UNIX® commands
  - Windows® system administration
- Understands networking concepts and language
- Is able to program in Java™ and XML
- Understands security issues
- Understands the customer's problem domain

References

Manual Organization

This guide contains the following chapters:

- Chapter 1, "Introduction", provides a brief explanation about the purpose, the scope and the definitions involved in this document.
- Chapter 2, "System Architecture", provides a description of its architecture.
- Chapter 3, "Compatibility with previous versions"
- Chapter 4, "General Description", gives a general description of the Subscription Repository.
- Chapter 5, "Web Services", describes the web services to interact with the system.
- Chapter 6, "Appendix"

Conventions

The following typographical conventions are used in this guide:

Font	What the Font Represents
Italic	Book or manual titles, and man page names
	Provides emphasis
	Specifies a variable that you must supply when entering a command
	Parameters to a method
Bold	New terms
`Computer`	Text and items on the computer screen
	Command names
	Method names
	File and directory names
	Process names
	Window/dialog box names
	XML tag references
`Computer Bold`	Text that you must type
Keycap	Keyboard keys
[Button]	Buttons on the user interface
Menu Items	A menu name followed by a colon (:) means that you select the menu, then the item. When the item is followed by an arrow (->), a cascading menu follows

Install Location Descriptors

The following names are used throughout this guide to define install locations:

Descriptor	What the Descriptor Represents
`$JBOSS_HOME`	HOME The install location for JBoss. The install location for JBoss depends on the path defined in the SR installation. For example: The Windows location could be C:\SR\Jboss

1. Introduction

1.1. Purpose

This document is a manual for the users of the Subscription Respository. It explains every feature of the mechanism to interact with the Subscription Repository.

1.2. Document Scope

This document is oriented for the users of the Subscription Repository.

1.3. Definitions

1.3.1. Acronyms

SR:	Subscription Repository
SID:	Shared Information Model
CRUD:	Creation Read Update Delete
OM:	Order Management
WS:	Web Service
WSDL:	Web Service Definition Language
CFS:	Customer Facing Service
RFS:	Resource Facing Service
MTOM:	Message Transmission Optimization Mechanism

2. System Architecture

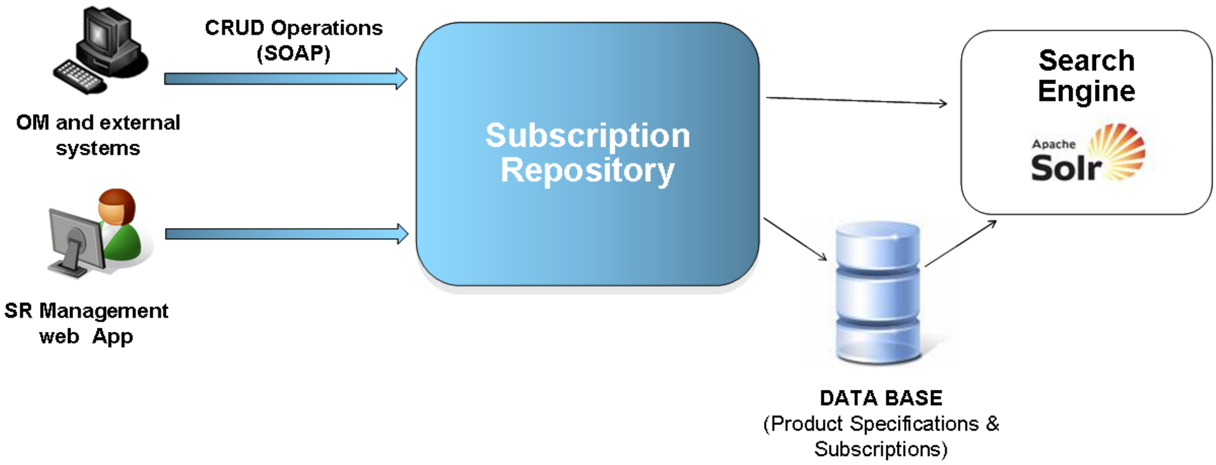
This chapter gives a general description of the architecture of the SR Application.

The SR application is a repository for subscribed products based on SID. It's written in java and runs as a standalone search server within a selvlet container. It uses a DataBase to store the product information and Solr (enterprise search platform from Apache) for providing advanced searching mechanisms.

Based on SID, it stores two kind of information:

- **Product Specifications**. Describe the structure of one product, including the services and resources that this product contains, and the different characteristics for each of them.
- **Product Subscriptions**. Main elements of the system. They contain specific information about a concrete product acquired by a customer. They follow the structure defined in the specification.

This diagram shows an overview about SR Application.



The main parts of SR Application are:

- **CRUD operations**. Basic read, create, update and remove operations (for product specifications and subscriptions). Exposed as web services. Product information is structured following the SID principles.
- **Advanced Search**. Execute advances searches against the system. The information in the Data Base is indexed in a Search Engine for this purpose.
- **External Processes**. Possibility to define java 'plugins' that interact with the system to export information to other systems and migrate the existing products.
- **Management Application**. A GUI is provided to manage the SR system.

3. Compatibility with previous versions

SR v7.0 is fully compatible with previous versions and maintains format of interfaces. Only it has been added new filters in current WS methods, new fields, new element “Resource Facing Service” and now a new databases are supported, Postgres Plus and MySQL database.

4. General Description

The main objective of subscription repository is to store product subscriptions, for this purpose, the subscription repository manages two types of information: specifications and product subscriptions.
The specifications define the product structure following the SID principles. SID proposes a model in wich each product is composed of services (customer facing services) and resources. The services and resources have their own specification that can be reused in different products.
The product subscriptions are instances of the specifications. The contain information of a concrete product subscribed to a user. They have to follow the product specification with concrete values assigned to the different characteristics.

A set of operations are defined for the user to interact with the system. These operations are exposed in two web services, *management* and *operations* web services.

4.1. CRUD operations (for specifications and subscriptions) and advanced search subscriptions

The *operations* web service contains the CRUD operations to manage the specifications and subscriptions, also contains the operations necessary to perform advanced search. The wsdl file (describes the web service) is located in http://localhost:8080/subscriptionrepository/operations/?wsdl.

4.1.1. Operations list

`createSpecification`: Operation to create specifications.
`updateSpecification`: Operation to update specifications.
`deleteSpecification`: Operation to delete specifications.
`querySpecification`: Operation to search specifications.
`createSubscription`: Operation to create subscriptions.
`updateSubscription`: Operation to update subscriptions.
`deleteSubscription`: Operation to delete subscriptions.
`querySubscription`: Operation to search subscriptions.
`createBasicSubscriptionElement`: Operation to create elements inside subscriptions.
`updateBasicSubscriptionElement`: Operation to update elements inside subscriptions.
`deleteBasicSubscriptionElement`: Operation to delete elements inside subscriptions.
`queryBasicSubscriptionElement`: Operation to search elements inside subscriptions.
`advancedQuerySubscription`: Operation for advanced searching subscriptions.
`advancedQuerySpecification`: Operation for advanced searching specifications.
`advancedQueryBasicSubscriptionElement`: Operation for advanced elements searching inside a subscription.
`saveAdvancedQueryConfiguration:` Operation to save SRAdvQuery.
`getAdvancedQueryConfiguration:` Operation to retrieve SRAdvQuery.
`listAdvancedQueriesConfigurations:` Operation to list all SRAdvQueries.
`deleteAdvancedQueryConfiguration:` Operation to delete a SRAdvQuery.
`deleteSpecificationByQuery:` Operation to delete specifications
`getAttachmentQuery:` Operation to retrieve one attached file.
`deleteAttachment:` Operation to delete attached files.
`createDependencies`: Operation to create subscription dependencies.
`deleteDependencies`: Operation to delete subscription dependencies.

`updateUsedDependencies`: Operation to update used value in dependency.
`getSourceDependencies`: Operation to retrieve source dependencies from one to many subscription productCodes.
`getTargetDependencies`: Operation to retrieve target dependencies from one to many subscription productCodes.
`requestSetterSpecificationProperties`: Operation to set properties or characteristic in specifications
`requestGetterSpecificationProperties`: Operation to retrieve properties or characteristic from specifications
`requestSetterSubscriptionProperties`: Operation to set properties or characteristic in subscriptions
`requestGetterSubscriptionProperties`: Operation to retrieve properties or characteristic from subscriptions

4.2. Management operations

The *management* web service contains the operations necessary to manage the bulk processes (external processes defined to interact with the system), index the subscription information and obtain a general status of the SR System. The wsdl file (describes the web service) is located in http://localhost:8080/subscriptionrepository/management/?wsdl.

4.2.1. Operations list

`listAvailableProcesses`: Operation to list all available processes.
`executeProcessByName`: Operation to execute a process identified by its name.
`executeProcess`: Operation to execute a process.
`finishProcess`: Operation to finish a running bulk process.
`getProcessStatus`: Operation for getting a given process status.
`getAllProcessStatus`: Operation for getting all process status.
`cleanHistoricProcesses`: Operation to clear the historic of executed processes.
`executeIndexProcess`: Operation to execute indexing process.
`executeReindexProcess`: Operation to execute reindexing process.
`getSRStatus`: Operation to get the subscription repository status.
`getConfigurationParameter`: Operation to get a SR configuration param.
`setConfigurationParameter`: Operation to set a SR configuration param.
`reloadCache`: Operation to reload local SR Specification cache.

5. Web Services

5.1. Data Model

The datamodel is based on SID principles. Following you can see datamodel description.

5.1.1. Specifications

5.1.1.1. Specification Description

The specification defines the product structure following the SID principles.

5.1.1.2. Product Specification

A Product specification must be created following this scheme:

Tag Name	*Type*[1]	*Max. length*[2]	*Description*
name	String	200	Specification name.
version	String	200	Specification version.
Parents	ComplexType – A sequence of parent tags.		See **parents** table.
description (optional)	String		Specification description.
creationDate	dateTime		Creation date.
updateDate	dateTime		Last modification date.
characteristics (optional)	ComplexType – A sequence of characteristic tags.		See **characteristic** table.
composite (optional)	ComplexType – A sequence of specificationRef tags.		See **specificationRef** table.
customerFacingServices (optional)	ComplexType - A sequence of customerFacingService tags.		See 5.1.1.3(CFS specification)
resources (optional)	ComplexType – A sequence of resource tags.		See 5.1.1.66 (Resource specification)
associatedItems (optional)	ComplexType – A sequence of item tags.		See 5.1.1.6 (Item specification)
AttachFiles (optional)	ComplexType – A sequence of file tags.		See **file** table.
usage (optional)	String	200	Specification usage, it can be DEFINITION or TEMPLATE
resource (optional)	Boolean		Specification is a resource if it is set to true
capacity (optional)	Integer		Specification capacity

characteristic		
name	String	Characteristic name.
description	String	Characteristic description.

[1] Charset is limited by the enconding.of the database.

[2] Maximum number of characters. If it´s undefined, there is not limit for the field.

(optional)		
type	String	Define the characteristic type. The possible values are String, Integer, Decimal, Date, Boolean.
multivalue (optional)	Boolean	Define if the characteristic value can be multivalue.
restriction (optional)	ComplexType	See **restriction** table

restriction		
length (optional)	String	Defines the exact length of the characteristic value.
range (optional)	String	Defines a range of values for the characteristic.
values (optional)	Enumeration	Defines a set of valid values for the characteristic.
pattern (optional)	Boolean	Defines a pattern. If characteristic is a "Date" type, the pattern attribute can be set to specify an specific format. SOLR format is set by default: "yyyy-MM-ddTHH:mm:ss.SSSZ".

specificationRef		
name	String	Name of the specification referenced.
version	String	Version of the specification referenced.
label (optional)	String	Label specification.

parent		
name	String	Name of the specification referenced.
version	String	Version of the specification referenced.

file		
fileName	String	Name of file.
content	String	Content of file (Use MTOM technology, so it´s a reference to external attached file).

Example:

```
<spec:products>
            <spec:product>
               <spec:name>DSLProduct</spec:name>
               <spec:version>1.0</spec:version>
               <spec:description>DSL product</spec:description>
               <spec:creationDate>2001-12-31T12:00:00</spec:creationDate>
               <spec:updateDate>2001-12-31T12:00:00</spec:updateDate>
               <spec:usage>DEFINITION</spec:usage>
               <spec:attachFiles>
                       <spec:file>
                               <spec:fileName>contract.pdf</spec:fileName>
                               <spec:content>cid:contract.pdf</spec:content>
                       </spec:file>
               </spec:attachFiles>
                <spec:customerFacingServices>
```

```
                  <spec:specificationRef>
                     <spec:name>DSLService</spec:name>
                     <spec:version>1.0</spec:version>
                  </spec:specificationRef>
               </spec:customerFacingServices>
               <spec:resources>
                  <spec:specificationRef>
                     <spec:name>DSLRouter</spec:name>
                     <spec:version>1.0</spec:version>
                  </spec:specificationRef>
               </spec:resources>
            </spec:product>
</spec:products>
```

5.1.1.3. CFSs Specification

A CFS specification must be created following this scheme:

Tag Name	*Type*[3]	*Max. length*[4]	*Description*
name	String	200	Specification name.
version	String	200	Specification version.
Parents	ComplexType – A sequence of parent tags.		See **parents** table.
description (optional)	String		Specification description.
creationDate	dateTime		Creation date.
updateDate	dateTime		Last modification date.
characteristics (optinal)	ComplexType – A sequence of characteristic tags.		See **characteristic** table.
composite (optional)	ComplexType – A sequence of specificationRef tags.		See **specificationRef** table.
resourceFacingServices (optional)	ComplexType – A sequence of resourceFacingService tags.		See 5.1.1.3 (RFS specification)
associatedItems (optional)	ComplexType – A sequence of item tags.		See 5.1.1.6 (Item specification)
attachFiles (optional)	ComplexType – A sequence of file tags.		See **file** table.
usage (optional)	String	200	Specification usage, it can be DEFINITION or TEMPLATE

characteristic		
name	String	Characteristic name.
description (optional)	String	Characteristic description.
type	String	Define the characteristic type. The possible values are String, Integer, Decimal, Date, Boolean.
multivalue (optional)	Boolean	Define if the characteristic value can be multivalue.
restriction	ComplexType	See **restriction** table

[3] Charset is limited by the enconding.of the database.

[4] Maximum number of characters. If it´s undefined, there is not limit for the field.

(optional)		

restriction		
length (optional)	String	Defines the exact length of the characteristic value.
range (optional)	String	Defines a range of values for the characteristic.
values (optional)	Enumeration	Defines a set of valid values for the characteristic.
pattern (optional)	Boolean	Defines a pattern.

specificationRef		
name	String	Name of the specification referenced.
version	String	Version of the specification referenced.
label (optional)	String	Label specification.

parent		
name	String	Name of the specification referenced.
version	String	Version of the specification referenced.

file		
fileName	String	Name of file.
content	String	Content of file (Use MTOM technology, so it´s a reference to external attached file).

Example:

```
<spec:customerFacingServices>
            <spec:customerFacingService>
               <spec:name>DSLService</spec:name>
               <spec:version>1.0</spec:version>
               <spec:description>DSL service</spec:description>
               <spec:creationDate>2001-12-31T12:00:00</spec:creationDate>
               <spec:updateDate>2001-12-31T12:00:00</spec:updateDate>
               <spec:usage>DEFINITION</spec:usage>
               <spec:attachFiles>
                       <spec:file>
                              <spec:fileName>contract.pdf</spec:fileName>
                              <spec:content>cid:contract.pdf</spec:content>
                       </spec:file>
               </spec:attachFiles>
               <spec:resourceFacingServices>
                       <spec:specificationRef>
                             <spec:name>DSLServiceStatistics</spec:name>
                             <spec:version>1.0</spec:version>
                             <spec:label>DSLServiceStatistics</spec:label>
                       </spec:specificationRef>
               </spec:resourceFacingServices>
               <spec:characteristics>
                  <spec:characteristic>
                     <spec:name>upSpeed</spec:name>
                     <spec:type>Integer</spec:type>
                  </spec:characteristic>
                  <spec:characteristic>
                     <spec:name>downSpeed</spec:name>
                     <spec:type>Integer</spec:type>
                  </spec:characteristic>
                  <spec:characteristic>
                     <spec:name>reboot_date</spec:name>
                     <spec:type>Date</spec:type>
                      <spec:restriction>
                          <spec:pattern>yyyy/MM/dd</spec:pattern>
                      </spec:restriction>
                  </spec:characteristic>
               </spec:characteristics>
            </spec:customerFacingService>
</spec:customerFacingServices>
```

5.1.1.4. RFSs Specification

A RFS specification must be created following this scheme:

Tag Name	*Type*[5]	*Max. length*[6]	*Description*
name	String	200	Specification name.
version	String	200	Specification version.
Parents	ComplexType – A sequence of parent tags.		See **parents** table.
description (optional)	String		Specification description.
creationDate	dateTime		Creation date.
updateDate	dateTime		Last modification date.
characteristics (optinal)	ComplexType – A sequence of characteristic tags.		See **characteristic** table.
composite (optional)	ComplexType – A sequence of specificationRef tags.		See **specificationRef** table.
associatedItems (optional)	ComplexType – A sequence of item tags.		See 5.1.1.6 (Item specification)
AttachFiles (optional)	ComplexType – A sequence of file tags.		See **file** table.
usage (optional)	String	200	Specification usage, it can be DEFINITION or TEMPLATE

characteristic		
name	String	Characteristic name.
description (optional)	String	Characteristic description.
type	String	Define the characteristic type. The possible values are String, Integer, Decimal, Date, Boolean.
multivalue (optional)	Boolean	Define if the characteristic value can be multivalue.
restriction (optional)	ComplexType	See **restriction** table

restriction		
length	String	Defines the exact length of the characteristic value.

[5] Charset is limited by the enconding.of the database.
[6] Maximum number of characters. If it´s undefined, there is not limit for the field.

(optional)		
range (optional)	String	Defines a range of values for the characteristic.
values (optional)	Enumeration	Defines a set of valid values for the characteristic.
pattern (optional)	Boolean	Defines a pattern.

specificationRef		
name	String	Name of the specification referenced.
version	String	Version of the specification referenced.
label (optional)	String	Label specification.

parent		
name	String	Name of the specification referenced.
version	String	Version of the specification referenced.

file		
fileName	String	Name of file.
content	String	Content of file (Use MTOM technology, so it´s a reference to external attached file).

Example:

```
<spec:resourceFacingServices>
            <spec: resourceFacingService>
               <spec:name>DSLServiceStatistics</spec:name>
               <spec:version>1.0</spec:version>
               <spec:description>DSL service statistics</spec:description>
               <spec:creationDate>2001-12-31T12:00:00</spec:creationDate>
               <spec:updateDate>2001-12-31T12:00:00</spec:updateDate>
               <spec:usage>DEFINITION</spec:usage>
               <spec:attachFiles>
                      <spec:file>
                             <spec:fileName>statistics.pdf</spec:fileName>
                             <spec:content>cid:statistics.pdf</spec:content>
                      </spec:file>
               </spec:attachFiles>
               <spec:characteristics>
                  <spec:characteristic>
                     <spec:name>clientSatisfaction</spec:name>
                     <spec:type>Integer</spec:type>
                  </spec:characteristic>
                  <spec:characteristic>
                     <spec:name>servicePerformance</spec:name>
                     <spec:type>Integer</spec:type>
                  </spec:characteristic>
               </spec:characteristics>
            </spec: resourceFacingService >
</spec: resourceFacingServices >
```

5.1.1.5. Resource Specification

A Resource specification must be created following this scheme:

Tag Name	*Type*[7]	*Max. length*[8]	*Description*
name	String	200	Specification name.
version	String	200	Specification version.
Parents	ComplexType – A sequence of parent tags.		See **parents** table.
description (optional)	String		Specification description.
creationDate	dateTime		Creation date.
updateDate	dateTime		Last modification date.
characteristics (optinal)	ComplexType – A sequence of characteristic tags.		See **characteristic** table.
composite (optional)	ComplexType – A sequence of specificationRef tags.		See **specificationRef** table.
associatedItems (optional)	ComplexType – A sequence of item tags.		See 5.1.1.6 (Item specification)
attachFiles (optional)	ComplexType – A sequence of file tags.		See **file** table.
usage (optional)	String	200	Specification usage, it can be DEFINITION or TEMPLATE

characteristic		
name	String	Characteristic name.
description (optional)	String	Characteristic description.
type	String	Define the characteristic type. The possible values are String, Integer, Decimal, Date, Boolean.
multivalue (optional)	Boolean	Define if the characteristic value can be multivalue.
restriction (optional)	ComplexType	See **restriction** table

restriction		
length (optional)	String	Defines the exact length of the characteristic value.
range (optional)	String	Defines a range of values for the characteristic.
values (optional)	Enumeration	Defines a set of valid values for the characteristic.
pattern (optional)	Boolean	Defines a pattern.

[7] Charset is limited by the enconding.of the database.

[8] Maximum number of characters. If it´s undefined, there is not limit for the field.

specificationRef		
name	String	Name of the specification referenced.
version	String	Version of the specification referenced.
label (optional)	String	Label specification.

parent		
name	String	Name of the specification referenced.
version	String	Version of the specification referenced.

file		
fileName	String	Name of file.
content	String	Content of file (Use MTOM technology, so it´s a reference to external attached file).

Example:

```
<spec:resources>
            <spec:resource>
               <spec:name>DSLRouter</spec:name>
               <spec:version>1.0</spec:version>
               <spec:description>DSL router</spec:description>
               <spec:creationDate>2001-12-31T12:00:00</spec:creationDate>
               <spec:updateDate>2001-12-31T12:00:00</spec:updateDate>
               <spec:usage>DEFINITION</spec:usage>
               <spec:attachFiles>
                       <spec:file>
                              <spec:fileName>contract.pdf</spec:fileName>
                              <spec:content>cid:contract.pdf</spec:content>
                       </spec:file>
               </spec:attachFiles>
               <spec:characteristics>
                  <spec:characteristic>
                     <spec:name>manufacturer</spec:name>
                     <spec:type>String</spec:type>
                  </spec:characteristic>
                  <spec:characteristic>
                     <spec:name>model</spec:name>
                     <spec:type>String</spec:type>
                  </spec:characteristic>
                  <spec:characteristic>
                     <spec:name>version</spec:name>
                     <spec:type>String</spec:type>
                  </spec:characteristic>
               </spec:characteristics>
            </spec:resource>
</spec:resources>
```

5.1.1.6. Item Specification

An Item specification must be created following this scheme:

Tag Name	*Type*[9]	*Max. length*[10]	*Description*
name	String	200	Specification name.
version	String	200	Specification version.

[9] Charset is limited by the enconding.of the database.

[10] Maximum number of characters. If it´s undefined, there is not limit for the field.

Parents	ComplexType – A sequence of parent tags.		See **parents** table.
description (optional)	String		Specification description.
creationDate	dateTime		Creation date.
updateDate	dateTime		Last modification date.
characteristics (optinal)	ComplexType – A sequence of characteristic tags.		See **characteristic** table.
composite (optional)	ComplexType – A sequence of specificationRef tags.		See **specificationRef** table.
attachFiles (optional)	ComplexType – A sequence of file tags.		See **file** table.
usage (optional)	String	200	Specification usage, it can be DEFINITION or TEMPLATE

characteristic		
name	String	Characteristic name.
description (optional)	String	Characteristic description.
type	String	Define the characteristic type. The possible values are String, Integer, Decimal, Date, Boolean.
multivalue (optional)	Boolean	Define if the characteristic value can be multivalue.
restriction (optional)	ComplexType	See **restriction** table

restriction		
length (optional)	String	Defines the exact length of the characteristic value.
range (optional)	String	Defines a range of values for the characteristic.
values (optional)	Enumeration	Defines a set of valid values for the characteristic.
pattern (optional)	Boolean	Defines a pattern.

specificationRef		
name	String	Name of the specification referenced.
version	String	Version of the specification referenced.
label (optional)	String	Label specification.

parent		
name	String	Name of the specification referenced.
version	String	Version of the specification referenced.

file		
fileName	String	Name of file.
content	String	Content of file (Use MTOM technology, so it´s a reference to external attached file).

Example:

```
<spec:items>
            <spec:item>
               <spec:name>DSLRouter</spec:name>
               <spec:version>1.0</spec:version>
               <spec:description>DSL router</spec:description>
               <spec:creationDate>2001-12-31T12:00:00</spec:creationDate>
               <spec:updateDate>2001-12-31T12:00:00</spec:updateDate>
               <spec:usage>DEFINITION</spec:usage>
               <spec:attachFiles>
                       <spec:file>
                              <spec:fileName>contract.pdf</spec:fileName>
                              <spec:content>cid:contract.pdf</spec:content>
                       </spec:file>
               </spec:attachFiles>
               <spec:characteristics>
                  <spec:characteristic>
                     <spec:name>manufacturer</spec:name>
                     <spec:type>String</spec:type>
                  </spec:characteristic>
                  <spec:characteristic>
                     <spec:name>model</spec:name>
                     <spec:type>String</spec:type>
                  </spec:characteristic>
                  <spec:characteristic>
                     <spec:name>version</spec:name>
                     <spec:type>String</spec:type>
                  </spec:characteristic>
               </spec:characteristics>
            </spec:item>
</spec:items>
```

5.1.2. Subscriptions

5.1.2.1. Subscription Description

A subscription must be created following this scheme:

Tag Name	*Type*[11]	*Max. length*[12]	*Description*
name	String	200	Ref. to specification name
version	String	200	Ref. to specification version
label (optional)	String	200	Subscription label.
status (optional)	String	200	Subscription status.
administrativeStatus (optional)	String	200	Subscription administrativeStatus
operationalStatus (optional)	String	200	Subscription operationalStatus

[11] Charset is limited by the enconding.of the database.

[12] Maximum number of characters. If it´s undefined, there is not limit for the field.

subStatus (optional)	String	200	Subscription subStatus
resource (optional)	Boolean		Subscription is resource or not. Only for product subscriptions.
capacity (optional)	Integer		Subscription capacity only for product subscriptions.
used (optional)	Integer		Subscription used by other dependent subscriptions. Only for product subscriptions.
characteristics (optional)	ComplexType – A sequence of characteristic tags.		See **characteristic** table.
productCode	String	200	Product code.
customerCode	String	200	Customer code.
customerLabel	String	200	Customer label
creationDate (mandatory to insert)	dateTime		Creation date.
updateDate (optional)	dateTime		Last modification date.
composite (optional)	ComplexType – A sequence of product tags.		Defines a sequence of products.
customerFacingServices (optional)	ComplexType - A sequence of customerFacingService tags.		See **customerFacingService** table.
resources (optional)	ComplexType – A sequence of resource tags.		See **resource** table.
associatedItems (optional)	ComplexType – A sequence of item tags.		See **item** table.
pricing (optional)	ComplexType		See **pricing** table.
attachFiles (optional)	ComplexType – A sequence of file tags.		See **file** table.

customerFacingService		
name	String	CFS name.
version	String	CFS version.
label (optional)	String	CFS label.
characteristics (optional)	ComplexType – A sequence of characteristic tags.	See **characteristic** table.
composite (optional)	ComplexType - A sequence of customerFacingService tags.	See **customerFacingService** table.
resourceFacingServices (optional)	ComplexType - A sequence of resourceFacingService tags.	See **resourceFacingService** table.
associatedItems (optional)	ComplexType – A sequence of item tags.	See **item** table.
customerFacingServiceStatus	String	CFS component status

(optional)		

resourceFacingService		
name	String	RFS name.
version	String	RFS version.
label (optional)	String	RFS label.
characteristics (optional)	ComplexType – A sequence of characteristic tags.	See **characteristic** table.
composite (optional)	ComplexType - A sequence of resourceFacingService tags.	See **resourceFacingService** table.
associatedItems (optional)	ComplexType - A sequence of resourceFacingService tags.	See **item** table.
resourceFacingServiceStatus (optional)	String	RFS status

resource		
name	String	Resource name.
version	String	Resource version.
label (optional)	String	Resource label.
characteristics (optional)	ComplexType – A sequence of characteristic tags.	See **characteristic** table.
composite (optional)	ComplexType – A sequence of resource tags.	See **resource** table.
associatedItems (optional)	ComplexType - A sequence of resourceFacingService tags.	See **item** table.
resourceStatus (optional)	String	Resource component status

item		
name	String	Item name.
version	String	Item version.
label (optional)	String	Item label.
characteristics (optional)	ComplexType – A sequence of characteristic tags.	See **characteristic** table.
composite (optional)	ComplexType – A sequence of item tags.	See **item** table.

characteristic		
name	String	Characteristic name.
value	String	Characteristic value. Also, you can define a set

		of values adding a parent tag named values and a set of value tags.

pricing		
offerCode (optional)	String	Offer code.
price (optional)	ComplexType – A sequence of price tags.	See **price** table.

price		
name	String	Price name.
description (optional)	String	Price description.
validFor (optional)	ComplexType	See **validForInterval**. Define a date range for the validity of charge.
price (optional)	ComplexType – A sequence of children price policy included in price tag.	See **price** table.
charge (optional)	ComplexType	See **charge** table.
allowance (optional)	ComplexType	See **allowance** table.
discount (optional)	ComplexType	See **discount** table.

validForInterval		
from (optional)	Date	Init date.
to (optional)	Date	Finish date.

charge		
name	String	Charge name.
description (optional)	String	Charge description.
type	String	Define the charge type. The possible values are OneTimeCharge, RecurringCharge, and UsageCharge.
periodicity	String	Define the charge periodicity. The possible values are OneTime, Monthly, Quarterly and Yearly.
units	String	Units of charges.
amount	Decimal	Amount of charges.

allowance		
name	String	Allowance name.
description (optional)	String	Allowance description.
units	String	Units of allowances.
amount	Decimal	Amount of allowances.

discount		
name	String	Discount name.
description (optional)	String	Discount description.
units	String	Units of discounts.
amount	Decimal	Amount of discounts.

file		
fileName	String	Name of file.
content	String	Content of file (Use MTOM technology, so it´s a reference to external attached file).

Example:

```
<sub:product>
            <sub:name>DSLProduct</sub:name>
            <sub:version>1.0</sub:version>
            <sub:productCode>SN-0001</sub:productCode>
            <sub:customerCode>Javi</sub:customerCode>
            <sub:creationDate>2001-12-31T12:00:00</sub:creationDate>
            <sub:updateDate>2001-12-31T12:00:00</sub:updateDate>
             <sub:attachFiles>
                       <sub:file>
                              <sub:fileName>contract.pdf</sub:fileName>
                              <sub:content>cid:contract.pdf</sub:content>
                       </sub:file>
               </sub:attachFiles>
            <sub:customerFacingServices>
               <sub:customerFacingService>
                  <sub:name>DSLService</sub:name>
                  <sub:version>1.0</sub:version>
                  <sub:characteristics>
                     <sub:characteristic>
                        <sub:name>upSpeed</sub:name>
                        <sub:value>10</sub:value>
                     </sub:characteristic>
                     <sub:characteristic>
                        <sub:name>downSpeed</sub:name>
                        <sub:value>10</sub:value>
                     </sub:characteristic>
                  </sub:characteristics>
               </sub:customerFacingService>
            </sub:customerFacingServices>
            <sub:resources>
               <sub:resource>
                  <sub:name>DSLRouter</sub:name>
                  <sub:version>1.0</sub:version>
                  <sub:characteristics>
                     <sub:characteristic>
                        <sub:name>manufacturer</sub:name>
                        <sub:value>Alcatel</sub:value>
                     </sub:characteristic>
                     <sub:characteristic>
                        <sub:name>model</sub:name>
                        <sub:value>Alcatel-DSL</sub:value>
                     </sub:characteristic>
                     <sub:characteristic>
                        <sub:name>version</sub:name>
                        <sub:value>1.1</sub:value>
                     </sub:characteristic>
                  </sub:characteristics>
               </sub:resource>
            </sub:resources>
         </sub:product>
```

5.2. Operations

5.2.1. Operations List

5.2.1.1. Create Specification

Operation for creating specifications.

Description:
Operation for creating specifications in the repository.

- *customerFacingServices (optional)* - Valid set of customerFacingService objects (at least one).
- *resources (optional)* – Valid set of resource objects (at least one).
- *products (optional)* – Valid set of product objects (at least one).

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:spec="http://www.sr.om.hp.com/Specifications">
   <soapenv:Header/>
   <soapenv:Body>
      <spec:createSpecification>
         <spec:customerFacingServices>
            <spec:customerFacingService>
               .....
            </spec:customerFacingService>
         </spec:customerFacingServices>
         <spec:resources>
            <spec:resource>
               .....
            </spec:resource>
         </spec:resources>
         <spec:products>
            <spec:product>
               .....
            </spec:product>
         </spec:products>
      </spec:createSpecification>
   </soapenv:Body>
</soapenv:Envelope>
```

Response
Generate a **Result** response with a Status tag that contains the type of operation, the description of the result and the result code.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns4:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns4:status>
            <ns4:operation>create</ns4:operation>
            <ns4:description>Success</ns4:description>
            <ns4:resultCode>OK</ns4:resultCode>
         </ns4:status>
      </ns4:result>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<ns4:error>
```

```
               <ns4:errorCode>100011</ns4:errorCode>
               <ns4:errorDescription>Could not find specification.</ns4:errorDescription>
</ns4:error>
```

5.2.1.2. Update Specification

Operation for updating specifications.

Description:
Operation for updating specifications in the repository.

- *customerFacingServices (optional)* - Valid set of customerFacingService objects (at least one).
- *resources (optional)* – Valid set of resource objects (at least one).
- *products (optional)* – Valid set of product objects (at least one).

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:spec="http://www.sr.om.hp.com/Specifications">
   <soapenv:Header/>
   <soapenv:Body>
      <spec:updateSpecification>
         <spec:resources>
            <spec:resource>
               .....
            </spec:resource>
         </spec:resources>
      </spec:updateSpecification>
   </soapenv:Body>
</soapenv:Envelope>
```

Response
Generate a **Result** response with a Status tag that contains the type of operation, the description of the result and the result code.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns4:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns4:status>
            <ns4:operation>update</ns4:operation>
            <ns4:description>Success</ns4:description>
            <ns4:resultCode>OK</ns4:resultCode>
         </ns4:status>
      </ns4:result>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<ns4:error>
               <ns4:errorCode>100011</ns4:errorCode>
               <ns4:errorDescription>Could not find specification.</ns4:errorDescription>
</ns4:error>
```

Operation can be done, returning a resultCode with “WARNING” value. For instance, with a specification update with subscriptions depending of it. Description field contains detailed message. (When a specification has depending subscriptions, then specification must be compatible with existent version (it´s compatible if has al least old data (chars, composite, items, and other spec references) and new characteristics must be "not mandatory")).

5.2.1.3. Delete Specification

Operation for deleting specifications.

Description:
Operation for deleting specifications in the repository.

- *specificationRef (at least one)* - Contains the name and the version of the specification to delete.

Example with specific name and version:

```
<spec:specificationRef>
            <spec:name>DSLProduct</spec:name>
            <spec:version>1.0</spec:version>
</spec:specificationRef>
```

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:spec="http://www.sr.om.hp.com/Specifications">
   <soapenv:Header/>
   <soapenv:Body>
      <spec:deleteSpecification>
         <spec:specificationRef>
            <spec:name>DSLProduct</spec:name>
            <spec:version>1.0</spec:version>
         </spec:specificationRef>
         <spec:specificationRef>
            <spec:name>DSLService</spec:name>
            <spec:version>1.0</spec:version>
         </spec:specificationRef>
         <spec:specificationRef>
            <spec:name>DSLRouter</spec:name>
            <spec:version>1.0</spec:version>
         </spec:specificationRef>
      </spec:deleteSpecification>
   </soapenv:Body>
</soapenv:Envelope>
```

Response
Generate a **Result** response with a Status tag that contains the type of operation, the description of the result and the result code.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns4:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns4:status>
            <ns4:operation>delete</ns4:operation>
            <ns4:description>Success</ns4:description>
            <ns4:resultCode>OK</ns4:resultCode>
         </ns4:status>
      </ns4:result>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<ns4:error>
```

```
               <ns4:errorCode>100030</ns4:errorCode>
               <ns4:errorDescription>Cannot delete specification while existing specifications
reference it.</ns4:errorDescription>
</ns4:error>
```

5.2.1.4. Delete Specification By Query

Operation for deleting specifications, using a query. Only can exist one query in the request.

Description:
Operation for deleting specifications in the repository. Fields to filter:

- *type* - Define specification type to delete.
- *name* – Define specification name to delete.
- *version* – Define specification version to delete.
- *creationDate* – Define creation date range to delete.
- *updateDate* – Define update date range to delete.
- *indexed* – Define if it should delete indexed specifications to delete.

Example with specific name and version:

```
<spec:query>
            <spec:name>DSLProduct</spec:name>
            <spec:version>1.0</spec:version>
</spec:query>
```

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:spec="http://www.sr.om.hp.com/Specifications">
   <soapenv:Header/>
   <soapenv:Body>
      <spec:deleteSpecificationByQuery>
         <spec:query>
            <spec:name>DSLProduct</spec:name>
            <spec:version>1.0</spec:version>
         </spec:query>
      </spec:deleteSpecificationByQuery>
   </soapenv:Body>
</soapenv:Envelope>
```

Response
Generate a **Result** response with a Status tag that contains the type of operation, the description of the result and the result code.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns4:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns4:status>
            <ns4:operation>delete</ns4:operation>
            <ns4:description>Success</ns4:description>
            <ns4:resultCode>OK</ns4:resultCode>
         </ns4:status>
      </ns4:result>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<ns4:error>
               <ns4:errorCode>100030</ns4:errorCode>
               <ns4:errorDescription>Cannot delete specification while existing specifications
reference it.</ns4:errorDescription>
</ns4:error>
```

5.2.1.5. Query Specification

Operation for searching specifications.

Description:
Operation for searching specifications in the repository.

- *querySpecification (required)* - Lists all specifications in the database unless you include any of these options to limit the search.

Filter Name	*Required*	*Type*	*Description*	*Example*
type	No	String	Type of specification. Possible values: Service, Resource or Product.	<type>Service</type>
name	No	String	Specification name.	<name>DSLService</name>
version	No	String	Specification version.	<version>1.0</version>
usage	No	String	Specification usage.	<usage>TEMPLATE</usage>
creationDate	No	Date	Creation date. Use two attributes (minValue and maxValue) to filter by date range. You can set a minValue or a maxValue (or both).	<creationDate> <minValue> 2000-12-30T12:00:00 </minValue> <maxValue> 2001-01-01T12:00:00 </maxValue> </creationDate>
updateDate	No	Date	Last modification date. Use two attributes (minValue and maxValue) to filter by date range. You can set a minValue or a maxValue (or both).	<updateDate <minValue> 2000-12-30T12:00:00 </minValue> <maxValue> 2001-01-01T12:00:00 </maxValue> </updateDate>

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:spec="http://www.sr.om.hp.com/Specifications">
   <soapenv:Header/>
   <soapenv:Body>
      <spec:querySpecification>
         <spec:name>DSLProduct</spec:name>
         <spec:version>1.0</spec:version>
      </spec:querySpecification>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

Generate a **result** response with a Status tag and an Output tag which contains the total string and a set of specifications.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns4:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns4:status>
            <ns4:operation>query</ns4:operation>
            <ns4:description>Success</ns4:description>
            <ns4:resultCode>OK</ns4:resultCode>
         </ns4:status>
         <ns4:output>
            <ns4:total>1</ns4:total>
            <ns4:specifications>
               <ns4:product>
                  <ns4:name>DSLProduct</ns4:name>
                  <ns4:version>1.0</ns4:version>
                  <ns4:description>DSL product</ns4:description>
                  .....
               </ns4:product>
            </ns4:specifications>
         </ns4:output>
      </ns4:result>
   </env:Body>
</env:Envelope>
```

5.2.1.6. Create Subscription

Operation for creating subscriptions.

Description:

- *product (at least one)* – A valid product.

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soapenv:Header/>
   <soapenv:Body>
      <sub:createSubscription>
         <sub:product>
            <sub:name>DSLProduct</sub:name>
            <sub:version>1.0</sub:version>
            <sub:productCode>SN-0001</sub:productCode>
            .....
         </sub:product>
         <sub:product>
            …..
         </sub:product>
            …..
      </sub:createSubscription>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

Generate a **Result** response with a Status tag that contains the type of operation, the description of the result and the result code.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
```

```
            <ns3:operation>create</ns3:operation>
            <ns3:description>Success</ns3:description>
            <ns3:resultCode>OK</ns3:resultCode>
         </ns3:status>
      </ns3:result>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<ns3:error>
               <ns3:errorCode>100062</ns3:errorCode>
               <ns3:errorDescription>There is a subscription with the same product code [SN-
0001]</ns3:errorDescription>
</ns3:error>
```

5.2.1.7. Update Subscription

This operation updates subscriptions.

Description:
Operation for updating subscriptions in the repository and updating operational statuses..

- *product (0 or more)* – A valid product to update. It updates the whole subscription both in database and SOLR.
- *productOperationalStatus (0 or more)* – It updates the fields operational status and substatus just in database for the desired subscription. Both fields are not indexed and can be updated in individually with this operation. Please notice that this sub-operation "*productOperationalStatus"* is executed after "product" update, so it is possible that a subscription is updated 2 times if required.

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soapenv:Header/>
   <soapenv:Body>
      <sub:updateSubscription>

         <sub:product>
            <sub:name>DSLProduct</sub:name>
            <sub:version>1.0</sub:version>
            <sub:productCode>SN-0001</sub:productCode>
            .....
         </sub:product>
         <sub:product>
            .....
         </sub:product>
          .....
         <sub:productOperationalStatus>
            <sub:operationalStatus>ACTIVE</sub:operationalStatus>
            <sub:subStatus>READY</sub:subStatus>
            <sub:productCode>SN-0001</sub:productCode>
         </sub:productOperationalStatus>
          .....
         <sub:productOperationalStatus>
         </sub:productOperationalStatus>

      </sub:updateSubscription>
   </soapenv:Body>
</soapenv:Envelope>
```

Response
Generate a **Result** response with a Status tag that contains the type of operation, the description of the result and the result code.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>update</ns3:operation>
            <ns3:description>Success</ns3:description>
            <ns3:resultCode>OK</ns3:resultCode>
         </ns3:status>
      </ns3:result>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<ns3:error>
               <ns3:errorCode>100064</ns3:errorCode>
               <ns3:errorDescription>
                Could not find subscription with product code [SN-0001].
              </ns3:errorDescription>
</ns3:error>
```

5.2.1.8. Delete Subscription

Operation for deleting subscriptions.

Description:
Operation for deleting subscriptions in the repository.

- *productRef (at least one)* – Contains a valid productCode.

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soapenv:Header/>
   <soapenv:Body>
      <sub:deleteSubscription>
         <sub:productRef>
            <sub:productCode>SN-0001</sub:productCode>
         </sub:productRef>
      </sub:deleteSubscription>
   </soapenv:Body>
</soapenv:Envelope>
```

Response
Generate a **Result** response with a Status tag that contains the type of operation, the description of the result and the result code.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
```

```
         <ns3:status>
            <ns3:operation>delete</ns3:operation>
            <ns3:description>Success</ns3:description>
            <ns3:resultCode>OK</ns3:resultCode>
         </ns3:status>
      </ns3:result>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<ns3:error>
               <ns3:errorCode>100064</ns3:errorCode>
               <ns3:errorDescription>Could not find subscription with product code [SN-
0001].</ns3:errorDescription>
</ns3:error>
```

5.2.1.9. Query Subscription

Operation for searching subscriptions.

Description:
Operation for searching subscriptions in the repository.

- *querySubscription (required)* - Lists all subscriptions in the database unless you include any of these options to limit the search.

FilterName	Required	Type	Default	Description	Example
historic	No	Boolean	False	Boolean indicating if the query should be performed into the Subscription_History table.	< historic >true</ historic >
indexed	No	Boolean	False	Boolean indicating if the subscription is indexed in Solr.	<indexed>true</indexed>
id	No	String	None	Subscription id.	<id>1</id>
productCode	No	String	None	Product serial number.	<productCode> SN-0001 </ productCode>
name	No	String	None	Subscription name.	<name>DSLService</name>
version	No	String	None	Subscription version.	<version>1.0</version>
customerCode	No	String	None	Customer.	<customerCode> Michael </ customerCode>

customerLabel	No	String	None	CustomerLabel	<customerLabel>123ABC</customerLabel>
creationDate	No	Date range.	None	Creation date. Use two attributes (minValue and maxValue) to filter by date range. You can set a minValue or a maxValue (or both).	<creationDate> <minValue> 2000-12-30T12:00:00 </minValue> <maxValue> 2001-01-01T12:00:00 </maxValue> </creationDate>
updateDate	No	Date range.	None	Last modification date. Use two attributes (minValue and maxValue) to filter by date range. You can set a minValue or a maxValue (or both).	<updateDate <minValue> 2000-12-30T12:00:00 </minValue> <maxValue> 2001-01-01T12:00:00 </maxValue> </updateDate>
Status	No	String	None	Status of subscription.	<status>statusTest</status>
Administrative Status	No	String	None	Administrative status of subscription.	<administrativeStatus>ACTIVE </administrativeStatus>
OperationalStatus	No	String	None	Operational status of subscription.	<operationalStatus>DELIVERED</operationalStatus>
SubStatus	No	String	None	Substatus of subscription.	<subStatus>FINISHED</operationalStatus>
Label	No	String	None	User identifier of product	<label>product1</label>
Sorts	No	Complex type (of two mandatory strings).	None	Results order. Order type can be only: "ASC" or "DESC"	<sorts> <sort> <fieldName>name</fieldName> <order>ASC</order> </sort> <sort>

					<fieldName>creationDate</fieldName> <order>DESC</order> </sort> </sorts>
OtherFilters	No	Complex type (of optionalstrings).	None	Additional filters for special/internal purposes	<otherFilters> <minId>100000</minId> </otherFilters>
Pagination	No	Complex type (of two mandatory numbers).	None	Pagination parameters "from" and "to"	<sub:pagination> <sub:from>0</sub:from> <sub:to>5</sub:to> </sub:pagination>
QueryCount flag	No	Boolean	false	Boolean parameter that determines if queryCount is being executed.	<sub:queryCountFlag>true</sub:queryCountFlag>

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soapenv:Header/>
   <soapenv:Body>
      <sub:querySubscription>
         <sub:productCode>SN-1</sub:productCode>
      </sub:querySubscription>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

Generate a **result** response with a Status tag and an Output tag which contains the total string and a set of subscriptions.

If flag queryCount is set to true, the query is going to return also the total of subscriptions found with the selected filters, but it's not affected by pagination, so for example if there are 150 subscriptions in database and pagination 0-5 is selected, the answer would be like this:

- Total: 5 because only 5 subscriptions are retrieved.
- QueryCount: 20 because there are 20 results found.

If selected pagination range is 0-0, it is possible to execute just the queryCount and skip the subscription search if we do not need to retrieve any result and have better counting performance. The returned answer would be like this:

- Total: 0 because no subscription is searched. No results, better performance.
- QueryCount: 20 because there are 20 results found.

Please notice also that queryCount has as maximum the "max_number_results parameter", so for example if this parameter is set to 100, the query will count subscriptions until it reach this maximum. If there are more it will display 100+.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
```

```
      <ns3:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>query</ns3:operation>
            <ns3:description>Success</ns3:description>
            <ns3:resultCode>OK</ns3:resultCode>
         </ns3:status>
         <ns3:output>
            <ns3:total>1</ns3:total>
            <ns3:queryCount>100+</ns3:queryCount>
            <ns3:products>
               <ns3:product>
                  <ns3:name>DSLProduct</ns3:name>
                  <ns3:version>1.0</ns3:version>
                  <ns3:productCode>SN-0001</ns3:productCode>
                  .....
               </ns3:product>
            </ns3:products>
         </ns3:output>
      </ns3:result>
   </env:Body>
</env:Envelope>
```

5.2.1.10. Advanced Subscription Search

Information contained in the DataBase is indexed in Solr for providing the advanced search functionality. The final user can execute advanced searches, it provides:

- Queries specifying pairs attribute-value, defining ranges in case of numeric and date values.
- Case insensitive and umlaut.
- Search on the whole text. The user can provide different values (like in a real google search), and obtains the product instances containing this information.
- Possibility of sorting by any field.
- Pagination and total results provided.
- Facet searches. User can obtain the number of products associated with any of the fields.
- Sort searches (optional). User can order the results by one or more fields (even kind of order).
- File searches (optional). User can filter elements by attachment or finding into the content.

Description

- *Pagination (optional)* - Provides pagination, specifies the range of pages to recover. Is defined as:

Filter Name	*Required*	*Type*	*Default*	*Description*	*Example*
From	Yes	String	False	Define the first page. (starts in 0)	<from>0</from>
To	Yes	String	False	Define the last page.	<to>7</to>

Example pagination:

```
<sub:pagination>
            <sub:from>0</sub:from>
            <sub:to>7</sub:to>
</sub:pagination>
```

- *commonFields (optional)* - Sorting by any subscription field.

Filter Name	*Required*	*Type*	*Default*	*Description*	*Example*

historic	No	Boolean	False	Boolean indicating if the query should be performed into the Subscription_History table.	< historic >true</ historic >
indexed	No	Boolean	False	Boolean indicating if the subscription is indexed in Solr.	<indexed>true</indexed>
id	No	String	None	Subscription id.	<id>1</id>
productCode	No	String	None	Product serial number.	<productCode> SN-0001 </ productCode>
name	No	String	None	Subscription name.	<name>DSLService</name>
version	No	String	None	Subscription version.	<version>1.0</version>
customerCode	No	String	None	Customer.	<customerCode> Michael </customerCode>
customerLabel	No	String	None	Customer label	<customerLabel>LABEL</customerLabel>
status	No	String	None	Status	<status>OK</status>
administrativeStatus	No	String	None	Administrative status	<administrativeStatus>ACTIVE</administrativeStatus>
operationalStatus	No*	*This field is ignored because operationalStatus is only stored in database but never indexed in SOLR.			
sublStatus	No*	*This field is ignored because subStatus is only stored in database but never indexed in SOLR.			
creationDate	No	Date range.	None	Creation date. Use two attributes (minValue and maxValue) to filter by date range. You can set a minValue or a maxValue (or both).	<creationDate> <minValue> 2000-12-30T12:00:00 </minValue> <maxValue> 2001-01-01T12:00:00 </maxValue> </creationDate>
updateDate	No	Date range.	None	Last modification date. Use two attributes (minValue and maxValue) to filter by date range. You can set a minValue or a maxValue	<updateDate <minValue> 2000-12-30T12:00:00 </minValue> <maxValue> 2001-01-01T12:00:00 </maxValue> </updateDate>

				(or both).	

Example of commonFields with interval in creationDate:

```
          <sub:commonFields>
             <sub:creationDate>
                <sub:minValue>2001-12-30T12:00:00</sub:minValue>
                <sub:maxValue>2002-01-01T12:00:00</sub:maxValue>
             </sub:creationDate>
          </sub:commonFields>
```

Example of commonFields with productCode:

```
          <sub:commonFields>
             <sub:productCode>SN-0001</sub:productCode>
          </sub:commonFields>
```

- *characteristics (optional)* – A list of characteristic objects (at least one) which allows queries specifying pairs attribute-value, defining ranges in case of numeric and date values. You must indicate a range or a value, not both.
  Characteristic is defined as:

Filter Name	Required	Type	Default	Description	Example
name	Yes	String	False	Characteristic name.	< name >true</ name >
range	Yes	Range.	False	Defines a range values for characteristic value. You can set a minValue or a maxValue (or both).	<range> <minValue> 5 </minValue> <maxValue> 20 </maxValue> </sub:range>
value	Yes	String	None	Characteristic value. Also, you can define a set of values adding a parent tag named values.	<value>5</value> Or <values> <value>5</value> <value>10</value> <value>15</value> </values>

Example of characteristic with range:

```
        <sub:characteristics>
             <sub:characteristic>
                <sub:name>downSpeed</sub:name>
                <sub:range>
                   <sub:minValue>5</sub:minValue>
                   <sub:maxValue>20</sub:maxValue>
                </sub:range>
            </sub:characteristic>
          </sub:characteristics>
```

Example of characteristic with value:

```
        <sub:characteristics>
             <sub:characteristic>
                <sub:name>downSpeed</sub:name>
                <sub:value>10</sub:value>
             </sub:characteristic>
          </sub:characteristics>
```

Example of characteristic with values:

```
        <sub:characteristics>
             <sub:characteristic>
                <sub:name>downSpeed</sub:name>
                <sub:values>
                   <sub:value>5</sub:value>
                   <sub:value>10</sub:value>
                   <sub:value>15</sub:value>
                </sub:values>
             </sub:characteristic>
          </sub:characteristics>
```

- *charge (optional)* – Define the subscription charge:

Filter Name	Required	Type	Default	Description	Example
periodicity	Yes	String	False	Periodicity charge. Possible values: OneTime, Monthly, Quarterly or Yearly	<periodicity> Monthly </ periodicity >
value	Yes	Decimal Interval.	False	Defines a range values for charge value. Use two attributes (minValue and maxValue). You can set a minValue or a maxValue (or both).	<value> <minValue> 10 </minValue> </value>

Example of *charge* with maxValue:

```
        <sub:charge>
                <sub:periodicity>Monthly</sub: periodicity >
         <sub:values>
                <maxValue>30<maxValue>
         </sub:values>
        </sub: charge >
```

- *Text (optional)* – The text to search into all subscriptions (case insensitive). You can use special characters like wildcard. Some examples:

  Example with specific text:

```
        <sub:text>downSpeed</sub:text>
```

Example starts with wildcard:

```
        <sub:text>*speed</sub:text>
```

Example ends with wildcard:

```
        <sub:text>downs*</sub:text>
```

- *Facets (optional)* – Facet search is similar to use count agregator combined with group by modificator in an bbdd sql statement.

  Facets search xml element is optional. You must indicate al least one field name. It´s optional to indicate one value, several values or range values (these options are exclusive, so you can select only one of them). Each facet xml element can have several fields, in this case, it would be like a searchs with AND clause.

  Example:

```
<advancedQuery>
  …
  <facets>
    <facet>
    // facet.field for customer
    <field isCharacteristic=”false”>
      <name>customercode</name>
    </field>
    </facet>
    <facet>
    // facet.query for version:1.0
    <field>
      <name>version</name>
      <value>1.0</value>
    </field>
    </facet>
    <facet>
    // facet.query for version:(1.0 OR 4.0)
    <field>
      <name>version</name>
      <values>
        <value>1.0</value>
        <value>4.0</value>
      </values>
    </field>
    </facet>
    <facet>
    // facet.query for version:[1.0 TO 4.0]
    <field>
      <name>version</name>
      <range>
        <minValue>1.0</minValue>
        <maxValue>4.0</maxValue>
      </range>
    </field>
   </facet>
  </facets>
  …
</advancedQuery>
```

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soapenv:Header/>
   <soapenv:Body>
      <sub:advancedQuerySubscription>

         <sub:pagination>
            <sub:from>0</sub:from>
            <sub:to>7</sub:to>
         </sub:pagination>

         <sub:commonFields>
            <sub:productCode>SN-0001</sub:productCode>
               <sub:minValue>2000-12-30T12:00:00</sub:minValue>
               <sub:maxValue>2001-01-01T12:00:00</sub:maxValue>
         </sub:commonFields>

         <sub:characteristics>
            <sub:characteristic>
               <sub:name>downSpeed</sub:name>
               <sub:range>
                  <sub:minValue>5</sub:minValue>
                  <sub:maxValue>20</sub:maxValue>
               </sub:range>
            </sub:characteristic>
         </sub:characteristics>

        <sub:charge>
               <sub:periodicity>Monthly</sub: periodicity >
       <sub:values>
               <maxValue>30<maxValue>
       </sub:values>
       </sub: charge >

       <sub:text>SN-000*</sub:text>

       <sub:facets>
            <sub:field>
               <sub:name>version</sub:name>
            </sub:field>
       </sub:facets>

      </sub:advancedQuerySubscription>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

Generate a **result** response with a Status tag and an Output tag which contains the number of subscriptions recovered as string and a set of subscriptions.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>query</ns3:operation>
            <ns3:description>Success</ns3:description>
            <ns3:resultCode>OK</ns3:resultCode>
         </ns3:status>
         <ns3:output>
            <ns3:total>1</ns3:total>
            <ns3:products>
               <ns3:product>
                  <ns3:name>DSLProduct</ns3:name>
                  <ns3:version>1.0</ns3:version>
                  <ns3:productCode>SN-0001</ns3:productCode>
       ......
               </ns3:product>
            </ns3:products>
            <ns3:facetCounts>
               <ns3:name>version:1.0</ns3:name>
               <ns3:count>10000</ns3:count>
            </ns3:facetCounts>
            <ns3:facetCounts>
               <ns3:name>version:2.0</ns3:name>
               <ns3:count>10000</ns3:count>
            </ns3:facetCounts>
            <ns3:facetCounts>
               <ns3:name>version:3.0</ns3:name>
               <ns3:count>10000</ns3:count>
            </ns3:facetCounts>
            <ns3:facetCounts>
               <ns3:name>version:4.0</ns3:name>
               <ns3:count>10000</ns3:count>
            </ns3:facetCounts>
            <ns3:facetCounts>
               <ns3:name>version:5.0</ns3:name>
               <ns3:count>10000</ns3:count>
            </ns3:facetCounts>
         </ns3:output>
      </ns3:result>
   </env:Body>
</env:Envelope>
```

- *Sorts (optional)* – Sort searches let obtain result with desire order.

  Sort search xml element is optional. You must indicate al least one field name. It´s mandatory to indicate one kind of order: “asc” or “desc”.

  Available fields are: productcode, id, customercode, customerlabel, name, version, creationdate, updatedate, label and administrativeStatus.

  Example:

```
<advancedQuery>
  …
           <!--Optional:-->
         <sorts>
            <!--1 or more repetitions:-->
            <sort isCharacteristic=”false”>
               <fieldName>customercode</fieldName>
               <order>asc</order>
            </sort>
         </sorts>
  …
</advancedQuery>
```

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soapenv:Header/>
   <soapenv:Body>
      <sub:advancedQuerySubscription>

         <sub:pagination>
            <sub:from>0</sub:from>
            <sub:to>7</sub:to>
         </sub:pagination>

         <sub:commonFields>
            <sub:productCode>SN-0001</sub:productCode>
               <sub:minValue>2000-12-30T12:00:00</sub:minValue>
               <sub:maxValue>2001-01-01T12:00:00</sub:maxValue>
         </sub:commonFields>

         <sub:characteristics>
            <sub:characteristic>
               <sub:name>downSpeed</sub:name>
               <sub:range>
                  <sub:minValue>5</sub:minValue>
                  <sub:maxValue>20</sub:maxValue>
               </sub:range>
            </sub:characteristic>
         </sub:characteristics>

        <sub:charge>
               <sub:periodicity>Monthly</sub: periodicity >
<sub:values>
       <maxValue>30<maxValue>
</sub:values>
       </sub: charge >

       <sub:text>SN-000*</sub:text>

           <!--Optional:-->
         <sub:sorts>
            <!--1 or more repetitions:-->
            <sub:sort>
               <sub:fieldName>customercode</sub:fieldName>
               <sub:order>asc</sub:order>
            </sub:sort>
         </sub:sorts>
      </sub:advancedQuerySubscription>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

Generate a normal **result** response with result list ordered by introduced params.

- *file (optional)* – Define the subscription attached files:

Filter Name	Required	Type	Default	Description	Example
hasAttachment	No	String	false	Valid values: “true” or “false”. Search elements with attachments.	<sub:hasAttachment>?</sub:hasAttachment>
text	No	String		Define a text to search into file content.	<sub:text>

Example of *file*:

```
         <sub:file>
            <!--Optional:-->
            <sub:hasAttachment>?</sub:hasAttachment>
            <!--Optional:-->
            <sub:text>?</sub:text>
         </sub:file>
```

5.2.1.11. Advanced Specification Search

Information contained in the DataBase is indexed in Solr for providing the advanced search functionality. The final user can execute advanced searches, it provides:

- Queries specifying pairs attribute-value, defining ranges in case of numeric and date values.
- Case insensitive and umlaut.
- Search on the whole text. The user can provide different values (like in a real google search), and obtains the product instances containing this information.
- Possibility of sorting by any field.
- Pagination and total results provided.

- Facet searches. User can obtain the number of products associated with any of the fields.
- Sort searches (optional). User can order the results by one or more fields (even kind of order).
- File searches (optional). User can filter elements by attachment or finding into the content.

Description

- *Pagination (optional)* - Provides pagination, specifies the range of pages to recover. Is defined as:

Filter Name	*Required*	*Type*	*Default*	*Description*	*Example*
From	Yes	String	False	Define the first page. (starts in 0)	<from>0</from>
To	Yes	String	False	Define the last page.	<to>7</to>

Example pagination:

```
<sub:pagination>
            <sub:from>0</sub:from>
            <sub:to>7</sub:to>
</sub:pagination>
```

- *Specification fields (optional)* - Search by any specification field.

Filter Name	*Required*	*Type*	*Default*	*Description*	*Example*
id	No	String	None	Specification id.	<id>1</id>
name	No	String	None	Specification name.	<name>DSLService</name>
version	No	String	None	Specification version.	<version>1.0</version>
usage	No	String	None	Specification usage	<usage>TEMPLATE</usage>
creationDate	No	Date range.	None	Creation date. Use two attributes (minValue and maxValue) to filter by date range. You can set a minValue or a maxValue (or both).	<creationDate> <minValue> 2000-12-30T12:00:00 </minValue> <maxValue> 2001-01-01T12:00:00 </maxValue> </creationDate>
updateDate	No	Date range.	None	Last modification date. Use two attributes (minValue and maxValue) to filter by date range. You can set a minValue or a maxValue (or both).	<updateDate <minValue> 2000-12-30T12:00:00 </minValue> <maxValue> 2001-01-01T12:00:00 </maxValue> </updateDate>

Example interval in creationDate:

```
<sub:creationDate>
   <sub:minValue>2001-12-30T12:00:00</sub:minValue>
   <sub:maxValue>2002-01-01T12:00:00</sub:maxValue>
</sub:creationDate>
```

- *characteristics (optional)* – A list of characteristic objects (at least one) which allows queries specifying pairs attribute-value, defining ranges in case of numeric and date values. You must indicate a range or a value, not both.
  Characteristic is defined as:

Filter Name	Required	Type	Default	Description	Example
name	Yes	String	False	Characteristic name.	< name >true</ name >
range	Yes	Range.	False	Defines a range values for characteristic value. You can set a minValue or a maxValue (or both).	<range> <minValue> 5 </minValue> <maxValue> 20 </maxValue> </sub:range>
value	Yes	String	None	Characteristic value. Also, you can define a set of values adding a parent tag named values.	<value>5</value> Or <values> <value>5</value> <value>10</value> <value>15</value> </values>
components	Optionally	String	None	Component name and version	<components> <component> <name>?</ name> <version>?</version> </component> </components>

Example of characteristic with range:

```
<sub:characteristics>
     <sub:characteristic>
        <sub:name>downSpeed</sub:name>
        <sub:range>
           <sub:minValue>5</sub:minValue>
           <sub:maxValue>20</sub:maxValue>
        </sub:range>
    </sub:characteristic>
  </sub:characteristics>
```

Example of characteristic with value:

```
<sub:characteristics>
     <sub:characteristic>
        <sub:name>downSpeed</sub:name>
```

```
               <sub:value>10</sub:value>
            </sub:characteristic>
         </sub:characteristics>
```

Example of characteristic with values:

```
       <sub:characteristics>
            <sub:characteristic>
               <sub:name>downSpeed</sub:name>
               <sub:values>
                  <sub:value>5</sub:value>
                  <sub:value>10</sub:value>
                  <sub:value>15</sub:value>
               </sub:values>
            </sub:characteristic>
         </sub:characteristics>
```

- *components (optional)* – Define the specification composite search:

Filter Name	Required	Type	Default	Description
name	Yes	String	False	Other spec name.
value	Yes	String	False	Other spec version.

Example of *charge* with maxValue:

```
<components>
<component>
     <name>?</ name>
     <version>?</version>
  </component>
</components>
```

- *Text (optional)* – The text to search into all specifications (case insensitive). You can use special characters like wildcard. Some examples:

Example with specific text:

```
       <sub:text>downSpeed</sub:text>
```

Example starts with wildcard:

```
       <sub:text>*speed</sub:text>
```

Example ends with wildcard:

```
       <sub:text>downs*</sub:text>
```

- *Facets (optional)* – Facet searches is similar to use count agregator combined with group by modificator in an bbdd sql statement.

  Facets search xml element is optional. You must indicate al least one field name. It´s optional to indicate one value, several values or range values (these options are exclusive, so you can select only one of them). Each facet xml element can have several fields, in this case, it would be like a searchs with AND clause.

Example:

```
<advancedQuery>
  …
  <facets>
    <facet>
    // facet.field for customer
    <field>
      <name>customercode</name>
    </field>
    </facet>
    <facet>
    // facet.query for version:1.0
    <field>
      <name>version</name>
      <value>1.0</value>
    </field>
    </facet>
    <facet>
    // facet.query for version:(1.0 OR 4.0)
    <field>
      <name>version</name>
      <values>
        <value>1.0</value>
        <value>4.0</value>
      </values>
    </field>
    </facet>
    <facet>
    // facet.query for version:[1.0 TO 4.0]
    <field>
      <name>version</name>
      <range>
        <minValue>1.0</minValue>
        <maxValue>4.0</maxValue>
      </range>
    </field>
   </facet>
  </facets>
  …
</advancedQuery>
```

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soapenv:Header/>
   <soapenv:Body>
      <sub:advancedQuerySpecification>

         <sub:pagination>
            <sub:from>0</sub:from>
            <sub:to>7</sub:to>
         </sub:pagination>

       <sub:text>DSLRouter</sub:text>

        <sub:facets>
            <sub:field>
               <sub:name>version</sub:name>
            </sub:field>
        </sub:facets>

      </sub:advancedQuerySpecification>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

Generate a **result** response with a Status tag and an Output tag which contains the number of specifications recovered as string and a set of specifications.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns4:resultExtended xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns4:status>
            <ns4:operation>query</ns4:operation>
            <ns4:description>Success</ns4:description>
            <ns4:resultCode>OK</ns3:resultCode>
         </ns4:status>
         <ns4:output>
            <ns4:total>0</ns4:total>
            <ns4:specifications/>
         </ns4:output>
      </ns4:resultExtended>
   </env:Body>
</env:Envelope>
```

- *Sorts (optional)* – Sort searchs lets obtain result with desire order.

  Sort search xml element is optional. You must indicate al least one field name. It´s mandatory to indicate one kind of order: “asc” or “desc”.

  Example:

```
<advancedQuerySpecification>
  …
           <!--Optional:-->
         <sorts>
            <!--1 or more repetitions:-->
            <sort>
               <fieldName>customercode</fieldName>
               <order>asc</order>
            </sort>
         </sorts>
  …
</advancedQuery>
```

- *file (optional)* – Define the subscription attached files:

Filter Name	Required	Type	Default	Description	Example
hasAttachment	No	String	false	Valid values: “true” or “false”. Search elements with attachments.	<sub:hasAttachment>?</sub:hasAttachment>
text	No	String		Define a text to search into file content.	<sub:text>

Example of *file*:

```
         <spec:file>
            <!--Optional:-->
            <spec:hasAttachment>?</spec:hasAttachment>
            <!--Optional:-->
            <spec:text>?</spec:text>
         </spec:file>
```

Response

Generate a normal **result** response with result list ordered by introduced params.

5.2.1.12. saveAdvancedQueryConfiguration

Operation for saving advanceQuery.

Description:

- *label (at least one)* – Contains a valid user identifier. Must be unique.
- *advancedQuery (at least one)* – Contains a query to persist.

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soapenv:Header/>
   <soapenv:Body>
      <sub:saveAdvancedQueryConfiguration>
         <label>query1</label>
         <sub:advancedQuery>
            <sub:pagination>
               <sub:from>1</sub:from>
               <sub:to>100</sub:to>
            </sub:pagination>

           <!—More params →

         </sub:advancedQuery>
      </sub:saveAdvancedQueryConfiguration>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

Generate a **Result** response with a Status tag that contains the type of operation, the description of the result and the result code.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>create</ns3:operation>
            <ns3:description>Success</ns3:description>
            <ns3:resultCode>OK</ns3:resultCode>
         </ns3:status>
      </ns3:result>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>create</ns3:operation>
            <ns3:description>Error</ns3:description>
            <ns3:resultCode>ERROR</ns3:resultCode>
            <ns3:error>
               <ns3:errorCode>201119</ns3:errorCode>
               <ns3:errorDescription>SQL Error inserting
AdvanceQueryConf.</ns3:errorDescription>
            </ns3:error>
         </ns3:status>
      </ns3:result>
   </env:Body>
</env:Envelope>
```

5.2.1.13. getAdvancedQueryConfiguration

Operation for obtain advancedQuery from repository. It can be specify id or/and label to identify query.

Description:
Operation for obtain advancedQuery from repository.

- *Label (mandatory if id not specified)* – Contains a valid label.
- *id (mandatory if label not specified)* – Contains a valid id.

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soapenv:Header/>
   <soapenv:Body>
      <sub:getAdvancedQueryConfiguration>
         <id>1</id>
         <label>test</label>
      </sub:getAdvancedQueryConfiguration>
   </soapenv:Body>
</soapenv:Envelope>
```

Response
Generate an **advancedQuery** response that contains the requested information. If there is not data, then return *<advancedQuery/>* element.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:getAdvancedQueryConfigurationResponse xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:advancedQuery>
            <ns3:pagination>
               <ns3:from>1</ns3:from>
               <ns3:to>100</ns3:to>
            </ns3:pagination>

            <!—More data →

         </ns3:advancedQuery>
      </ns3:getAdvancedQueryConfigurationResponse>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an exception is throwed. See error appendix.

```
<ns3:error>
               <ns3:errorCode>100121</ns3:errorCode>
               <ns3:errorDescription> SQL Error retrieving AdvanceQueryConf
</ns3:errorDescription>
</ns3:error>
```

5.2.1.14. listAdvancedQueriesConfigurations

Operation for list all queries from repository.

Description:
Operation for list all queries from repository. Without parameters.

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soapenv:Header/>
   <soapenv:Body>
      <sub:listAdvancedQueriesConfigurations/>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

Generate a **listAdvancedQueriesConfigurationsResponse** response with a advancedQueryConfigurationRef tag that contains the label and id of each query.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:listAdvancedQueriesConfigurationsResponse xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <advancedQueryConfigurationRef>
            <id>1</id>
            <label>test</label>
         </advancedQueryConfigurationRef>
         <advancedQueryConfigurationRef>
            <id>2</id>
            <label>test2</label>
         </advancedQueryConfigurationRef>
      </ns3:listAdvancedQueriesConfigurationsResponse>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error is throwed. See error appendix.

```
<ns3:error>
               <ns3:errorCode>100118</ns3:errorCode>
               <ns3:errorDescription> SQL Error occurs during list of
advancedQueryConf.</ns3:errorDescription>
</ns3:error>
```

5.2.1.15. deleteAdvancedQueryConfiguration

This operation deletes one or more AdvancedQueries in the repository.

Description:

- *id (required if label isn´t specified)* – Contains a valid id.
- *label (required if id isn´t specified)* – Contains a valid label.

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soapenv:Header/>
   <soapenv:Body>
      <sub:deleteAdvancedQueryConfiguration>
         <!--1 or more repetitions:-->
         <advancedQueryConfigurationRef>
            <id>1</id>
            <label>test</label>
         </advancedQueryConfigurationRef>
      </sub:deleteAdvancedQueryConfiguration>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

Generate a **Result** response with a Status tag that contains the type of operation, the description of the result and the result code.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>delete</ns3:operation>
            <ns3:description>Success</ns3:description>
            <ns3:resultCode>OK</ns3:resultCode>
         </ns3:status>
      </ns3:result>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<ns3:error>
               <ns3:errorCode>100123</ns3:errorCode>
               <ns3:errorDescription> SQL Error during AdvancedQueryConf
delete</ns3:errorDescription>
</ns3:error>
```

5.2.1.16. getAttachmentQuery

Operation for obtain one attachment data, filtering by productcode (per subscription) or name-version (per specification).

Description:
Operation for obtain one attachment data. Withparameters to find related component: subscription or specification.

SOAP API Format:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soap:Header/>
   <soap:Body>
      <sub:getAttachmentQuery>
         <!--Optional:-->
         <subscriptionFilter>
            <productcode>?</productcode>
            <filename>?</filename>
         </subscriptionFilter>
         <!--Optional:-->
         <specificationFilter>
            <name>?</name>
            <version>?</version>
            <filename>?</filename>
         </specificationFilter>
      </sub:getAttachmentQuery>
   </soap:Body>
</soap:Envelope>
```

Response
Generate **SRAttachmentQueryResponse** response with the filename and content.

If an exception occurs during the process execution, an error is throwed. See error appendix.

```
<ns3:error>
               <ns3:errorCode>100163</ns3:errorCode>
               <ns3:errorDescription> SQL Error retrieving attached file.
<ns3:errorDescription>
```

```
</ns3:error>
```

5.2.1.17. deleteAttachment

Operation for delete attached files using a filter clause.

Description:
Operation for delete one or more attached files. With parameters: subscription filter or specification filter. In both scenarios it´s possible to limit the operation scope only for one file, using "filename" clause.

SOAP API Format:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soap:Header/>
   <soap:Body>
      <sub:deleteAttachment>
         <!--Optional:-->
         <subscriptionFilter>
            <productcode>?</productcode>
            <!--Optional:-->
            <filename>?</filename>
         </subscriptionFilter>
         <!--Optional:-->
         <specificationFilter>
            <name>?</name>
            <version>?</version>
            <!--Optional:-->
            <filename>?</filename>
         </specificationFilter>
      </sub:deleteAttachment>
   </soap:Body>
</soap:Envelope>
```

Response
Generate a **SRAttachmentDeleteResponse** response with a string with the result.

If an exception occurs during the process execution, an error is throwed. See error appendix.

```
<ns3:error>
               <ns3:errorCode>100161</ns3:errorCode>
               <ns3:errorDescription> SQL Error deleting attached file in the database.
</ns3:errorDescription>
</ns3:error>
```

5.2.1.18. Create Basic Subscription Element

This operation can create basic components inside an existing subscription.

Description:
It can create multiple components in one single request. Each component is created inside a subscription, it is defined by a productCode (subscription) and a label (parent component).

Request parameters:

- *productCode* – Subscription identifier where the new component is inserted.
- *parentLabel* – Parent component label where the new component is inserted.
- *parentType* – (optional) If there are multiple components with the same label, it's possible to filter the parent by type.

- **component* – (one required) The component to be inserted.

SOAP API Format:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soap:Header/>
   <soap:Body>
      <sub:createBasicSubscriptionElement>
         <!--1 or more repetitions:-->
         <sub:createElement>
            <sub:productCode>?</sub:productCode>
            <sub:parentLabel>?</sub:parentLabel>
            <!--Optional:-->
            <sub:parentType>?</sub:parentType>

            <!-- *One of the next elements:-->
            <sub:product></sub:product>
            <sub:customerFacingService></sub:customerFacingService>
            <sub:resourceFacingService></sub:resourceFacingService>
            <sub:resource></sub:resource>
            <sub:item></sub:item>

         </sub:createElement>
      </sub:createBasicSubscriptionElement>
   </soap:Body>
</soap:Envelope>
```

Response

Generate a **ResultComponent** response with a Status tag that contains the type of operation, the description of the result and the result code.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:resultComponent xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>create</ns3:operation>
            <ns3:description>Success</ns3:description>
            <ns3:resultCode>OK</ns3:resultCode>
         </ns3:status>
      </ns3:resultComponent>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<ns3:error>
        <ns3:errorCode>100176</ns3:errorCode>
        <ns3:errorDescription>Invalid element to be inserted inside
[ResourceFacingService]</ns3:errorDescription>
</ns3:error>
```

5.2.1.19. Delete Basic Subscription Element

This operation can remove basic components from an existing subscription.

Description:

It can remove multiple components in one single request. Each component is removed from a subscription, it is defined by a productCode (subscription) and a label (component identifier).

Request parameters:

- *productCode* – Subscription identifier where the component is being removed from.
- *label* – Component identifier.
- *type* – (optional) If there are multiple components with the same label, it's possible to filter the desired one by type.

SOAP API Format:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soap:Header/>
   <soap:Body>
      <sub:deleteBasicSubscriptionElement>
         <!--1 or more repetitions:-->
         <sub:deleteElement>
            <sub:productCode>?</sub:productCode>
            <sub:label>?</sub:label>
            <!--Optional:-->
            <sub:type>?</sub:type>
         </sub:deleteElement>
      </sub:deleteBasicSubscriptionElement>
   </soap:Body>
</soap:Envelope>
```

Response
Generate a **ResultComponent** response with a Status tag that contains the type of operation, the description of the result and the result code.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:resultComponent xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>delete</ns3:operation>
            <ns3:description>Success</ns3:description>
            <ns3:resultCode>OK</ns3:resultCode>
         </ns3:status>
      </ns3:resultComponent>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<ns3:error>
       <ns3:errorCode>100170</ns3:errorCode>
       <ns3:errorDescription>Could not find the object labeled as
[testComponent]</ns3:errorDescription>
</ns3:error>
```

5.2.1.20. Update Basic Subscription Element

This operation can update basic components inside an existing subscription.

Description:
It can update multiple components in one single request. Each component is updated inside a subscription, it is defined by a productCode (subscription) and a label (component identifier).

Request parameters:

- *productCode* – Subscription identifier where the component is updated.
- *label* – Component label that identifies the component to be updated.
- **component* – (one required) The component to be updated.

SOAP API Format:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soap:Header/>
   <soap:Body>
      <sub:updateBasicSubscriptionElement>
         <!--1 or more repetitions:-->
         <sub:updateElement>
            <sub:productCode>?</sub:productCode>
            <sub:label>?</sub:label>

            <!–- *One of the next elements:-->
            <sub:product></sub:product>
            <sub:customerFacingService></sub:customerFacingService>
            <sub:resourceFacingService></sub:resourceFacingService>
            <sub:resource></sub:resource>
            <sub:item></sub:item>

          </sub:updateElement>
      </sub:updateBasicSubscriptionElement>
   </soap:Body>
</soap:Envelope>
```

Response
Generate a **ResultComponent** response with a Status tag that contains the type of operation, the description of the result and the result code.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:resultComponent xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>update</ns3:operation>
            <ns3:description>Success</ns3:description>
            <ns3:resultCode>OK</ns3:resultCode>
         </ns3:status>
      </ns3:resultComponent>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<ns3:error>
        <ns3:errorCode>100171</ns3:errorCode>
        <ns3:errorDescription>More than one object was found with the selected label
[testComponent]</ns3:errorDescription>
</ns3:error>
```

5.2.1.21. Query Basic Subscription Element

This operation can search basic components inside an existing subscription.

Description:
It can search a single component inside a subscription. The query is defined by a productCode (subscription) and a label (component identifier).
If there are multiple child components with the same label, all of them will be retrieved in the response.

Request parameters:

- *productCode* – Subscription identifier.
- *label* – Identifies the component to be found.

SOAP API Format:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soap:Header/>
   <soap:Body>
      <sub:queryBasicSubscriptionElement>
         <sub:productCode>?</sub:productCode>
         <sub:label>?</sub:label>
      </sub:queryBasicSubscriptionElement>
   </soap:Body>
</soap:Envelope>
```

Response
Generate a **ResultComponent** response with a Status tag that contains the type of operation, the description of the result, the result code and the search result output.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:resultComponent xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>query</ns3:operation>
            <ns3:description>Success</ns3:description>
            <ns3:resultCode>OK</ns3:resultCode>
         </ns3:status>
         <ns3:output>
            <ns3:total>1</ns3:total>
            <ns3:resourceFacingServices>
               <ns3:resourceFacingService>
                  <ns3:name>DSLServiceStatistics</ns3:name>
                  <ns3:version>1.0</ns3:version>
                  <ns3:label>DSLServiceStatistics 1.0</ns3:label>
                  <ns3:characteristics>
                     <ns3:characteristic>
                        <ns3:name>ClientSatisfaction</ns3:name>
                        <ns3:value>8</ns3:value>
                     </ns3:characteristic>
                     <ns3:characteristic>
                        <ns3:name>ServiceQuality</ns3:name>
                        <ns3:value>7</ns3:value>
                     </ns3:characteristic>
                  </ns3:characteristics>
               </ns3:resourceFacingService>
            </ns3:resourceFacingServices>
         </ns3:output>
      </ns3:resultComponent>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).
This operation should not respond with an exception, if there is a mistake in the request probably we will get a "Not found" response:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:resultComponent xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>query</ns3:operation>
            <ns3:description>Success</ns3:description>
            <ns3:resultCode>OK</ns3:resultCode>
         </ns3:status>
         <ns3:output>
            <ns3:total>0</ns3:total>
         </ns3:output>
      </ns3:resultComponent>
   </env:Body>
</env:Envelope>
```

5.2.1.22. Create Dependencies

This operation can create dependencies between subscriptions or subscription and a compoment inside a subscription.

Description:
It can create a dependency between two subscription or between a subscription and a component inside another subscription indicating wether the subscription involved are required to be existing in the system or not.

Request parameters:

- *List of dependency subscription*

SOAP API Format:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soap:Header/>
   <soap:Body>
      <sub:createDependencies>
         <!--1 or more repetitions:-->
         <sub:dependency mandatory="true" used="10" type="use">
            <sub:nodeA>
               <!--1 or more repetitions:-->
               <sub:productCode>SN-0001</sub:productCode>
            </sub:nodeA>
            <sub:nodeZ>
               <sub:productCode>SN-0002</sub:productCode>
            </sub:nodeZ>
         </sub:dependency>
      </sub:createDependencies>
   </soap:Body>
</soap:Envelope>
```

Response

Generate a **ResultDependencies** response with a Status tag that contains the type of operation, the description of the result, the result code and the output containing subscriptions which used has been modified.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:resultDependencies xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>create</ns3:operation>
            <ns3:description>Success</ns3:description>
            <ns3:resultCode>OK</ns3:resultCode>
         </ns3:status>
         <ns3:output>
            <ns3:total>1</ns3:total>
            <ns3:subscriptionStats>
               <ns3:subscriptionStat>
                  <ns3:productCode>SN-0001</ns3:productCode>
                  <ns3:capacity>50</ns3:capacity>
                  <ns3:used>10</ns3:used>
               </ns3:subscriptionStat>
            </ns3:subscriptionStats>
         </ns3:output>
      </ns3:resultDependencies>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).
This operation should not respond with an exception, if there is a mistake in the request probably we will get a “Not found” response:

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>create</ns3:operation>
            <ns3:description>Error</ns3:description>
            <ns3:resultCode>ERROR</ns3:resultCode>
            <ns3:error>
               <ns3:errorCode>201064</ns3:errorCode>
               <ns3:errorDescription>SQL Could not find subscription with product code SN-
0005</ns3:errorDescription>
            </ns3:error>
         </ns3:status>
      </ns3:result>
   </env:Body>
</env:Envelope>
```

5.2.1.23. Delete Dependencies

This operation can delete dependencies between subscriptions or subscription and a compoment inside a subscription.

Description:

It can delete a dependency between two subscription or between a subscription and a component inside another subscription indicating wether the subscription involved are required to be existing in the system or not.

Request parameters:

- *List of dependency subscription*

SOAP API Format:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soap:Header/>
   <soap:Body>
      <sub:deleteDependencies>
         <sub:dependencies>
            <sub:dependency>
               <sub:nodeZ>
                  <sub:references>
                     <sub:productCode>SN-0003</sub:productCode>
                     <sub:componentLabel>DSL Service 1.0</sub:componentLabel>
                  </sub:references>
               </sub:nodeZ>
               <sub:nodeA>
                  <sub:productCode>SN-0002</sub:productCode>
               </sub:nodeA>
            </sub:dependency>
         </sub:dependencies>
      </sub:deleteDependencies>
   </soap:Body>
</soap:Envelope>
```

Response

Generate a **ResultDependencies** response with a Status tag that contains the type of operation, the description of the result, the result code and the output containing subscriptions which used has been modified.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:resultDependencies xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>delete</ns3:operation>
            <ns3:description>Success</ns3:description>
            <ns3:resultCode>OK</ns3:resultCode>
         </ns3:status>
         <ns3:output>
            <ns3:total>1</ns3:total>
            <ns3:subscriptionStats>
               <ns3:subscriptionStat>
                  <ns3:productCode>SN-0002</ns3:productCode>
                  <ns3:capacity>50</ns3:capacity>
                  <ns3:used>0</ns3:used>
               </ns3:subscriptionStat>
            </ns3:subscriptionStats>
         </ns3:output>
      </ns3:resultDependencies>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>delete</ns3:operation>
            <ns3:description>Error</ns3:description>
            <ns3:resultCode>ERROR</ns3:resultCode>
            <ns3:error>
               <ns3:errorCode>201064</ns3:errorCode>
               <ns3:errorDescription>SQL Could not find subscription with product code SN-
0005</ns3:errorDescription>
            </ns3:error>
         </ns3:status>
      </ns3:result>
   </env:Body>
</env:Envelope>
```

5.2.1.24. Update Used Dependencies

This operation can update used dependencies values between subscriptions or subscription and a compoment inside a subscription.

Description:
It can update used value in a dependency between two subscription or between a subscription and a component inside another subscription indicating wether the subscription involved are required to be existing in the system or not.

Request parameters:

- *List of dependency subscription*

SOAP API Format:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soap:Header/>
   <soap:Body>
      <sub:deleteDependencies>
         <sub:dependencies>
            <sub:dependency>
               <sub:nodeZ>
                  <sub:references>
                     <sub:productCode>SN-0003</sub:productCode>
                     <sub:componentLabel>DSL Service 1.0</sub:componentLabel>
                  </sub:references>
               </sub:nodeZ>
               <sub:nodeA>
                  <sub:productCode>SN-0002</sub:productCode>
               </sub:nodeA>
            </sub:dependency>
         </sub:dependencies>
      </sub:deleteDependencies>
   </soap:Body>
</soap:Envelope>
```

Response
Generate a **ResultDependencies** response with a Status tag that contains the type of operation, the description of the result, the result code and the output containing subscriptions which used has been modified.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
```

```
   <env:Header/>
   <env:Body>
      <ns3:resultDependencies xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>update</ns3:operation>
            <ns3:description>Success</ns3:description>
            <ns3:resultCode>OK</ns3:resultCode>
         </ns3:status>
         <ns3:output>
            <ns3:total>1</ns3:total>
            <ns3:subscriptionStats>
               <ns3:subscriptionStat>
                  <ns3:productCode>SN-0002</ns3:productCode>
                  <ns3:capacity>55</ns3:capacity>
                  <ns3:used>45</ns3:used>
               </ns3:subscriptionStat>
            </ns3:subscriptionStats>
         </ns3:output>
      </ns3:resultDependencies>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>delete</ns3:operation>
            <ns3:description>Error</ns3:description>
            <ns3:resultCode>ERROR</ns3:resultCode>
            <ns3:error>
               <ns3:errorCode>201064</ns3:errorCode>
               <ns3:errorDescription>SQL Could not find subscription with product code SN-
0005</ns3:errorDescription>
            </ns3:error>
         </ns3:status>
      </ns3:result>
   </env:Body>
</env:Envelope>
```

5.2.1.25. Get Source Dependencies

This operation can retrieve all existing dependencies from a specific subscription.

Description:
It can retirve the whole dependencies from a set of subscriptions.

Request parameters:

- *List of productCodes*

SOAP API Format:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soap:Header/>
   <soap:Body>
      <sub:getSourceDependencies>
```

```
         <sub:nodeA>
            <!--1 or more repetitions:-->
            <sub:productCode>SN-0001</sub:productCode>
            <sub:productCode>SN-0002</sub:productCode>
         </sub:nodeA>
      </sub:getSourceDependencies>
   </soap:Body>
</soap:Envelope>
```

Response

Generate a **ResultDependencies** response with a Status tag that contains the type of operation, the description of the result, the result code and the output containing subscriptions which used has been modified and a dependencies list for the source.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:resultDependencies xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>query</ns3:operation>
            <ns3:description>Success</ns3:description>
            <ns3:resultCode>OK</ns3:resultCode>
         </ns3:status>
         <ns3:output>
            <ns3:total>2</ns3:total>
            <ns3:subscriptionStats>
               <ns3:subscriptionStat>
                  <ns3:productCode>SN-0001</ns3:productCode>
                  <ns3:capacity>50</ns3:capacity>
                  <ns3:used>15</ns3:used>
                  <ns3:dependencyListType>
                     <ns3:dependency type="use" used="15" mandatory="false"
bidirectional="false">
                        <ns3:nodeA>
                           <ns3:productCode>SN-0001</ns3:productCode>
                        </ns3:nodeA>
                        <ns3:nodeZ>
                           <ns3:productCode>SN-0002</ns3:productCode>
                        </ns3:nodeZ>
                     </ns3:dependency>
                  </ns3:dependencyListType>
               </ns3:subscriptionStat>
               <ns3:subscriptionStat>
                  <ns3:productCode>SN-0002</ns3:productCode>
                  <ns3:capacity>55</ns3:capacity>
                  <ns3:used>15</ns3:used>
                  <ns3:dependencyListType>
                     <ns3:dependency type="use" used="15" mandatory="true"
bidirectional="false">
                        <ns3:nodeA>
                           <ns3:productCode>SN-0002</ns3:productCode>
                        </ns3:nodeA>
                        <ns3:nodeZ>
                           <ns3:references>
                              <ns3:productCode>SN-0003</ns3:productCode>
                              <ns3:componentLabel>DSL Service 1.0</ns3:componentLabel>
                           </ns3:references>
                        </ns3:nodeZ>
                     </ns3:dependency>
                  </ns3:dependencyListType>
               </ns3:subscriptionStat>
            </ns3:subscriptionStats>
         </ns3:output>
      </ns3:resultDependencies>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:resultDependencies xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>query</ns3:operation>
            <ns3:description>Error</ns3:description>
            <ns3:resultCode>ERROR</ns3:resultCode>
            <ns3:error>
               <ns3:errorCode>100021</ns3:errorCode>
               <ns3:errorDescription>Request is empty</ns3:errorDescription>
            </ns3:error>
         </ns3:status>
      </ns3:resultDependencies>
   </env:Body>
</env:Envelope>
```

5.2.1.26. Get Target Dependencies

This operation can retrieve all existing dependencies from a specific subscription or a component inside a subscription.

Description:
It can retirve the whole dependencies from a set of subscription based on their productCodes or their productCodes and componentLabel.

Request parameters:

- *List of productCode or component reference (productCode + componentLabel)*

SOAP API Format:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soap:Header/>
   <soap:Body>
      <sub:getTargetDependencies>
         <sub:nodeZ>
            <sub:productCode>SN-0002</sub:productCode>
            <sub:references>
               <sub:productCode>SN-0003</sub:productCode>
               <sub:componentLabel>DSL Service 1.0</sub:componentLabel>
            </sub:references>
         </sub:nodeZ>
      </sub:getTargetDependencies>
   </soap:Body>
</soap:Envelope>
```

Response
Generate a **ResultDependencies** response with a Status tag that contains the type of operation, the description of the result, the result code and the output containing subscriptions which used has been modified and a dependencies list for the target elements.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
```

```
      <ns3:resultDependencies xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>query</ns3:operation>
            <ns3:description>Success</ns3:description>
            <ns3:resultCode>OK</ns3:resultCode>
         </ns3:status>
         <ns3:output>
            <ns3:total>2</ns3:total>
            <ns3:subscriptionStats>
               <ns3:subscriptionStat>
                  <ns3:productCode>SN-0001</ns3:productCode>
                  <ns3:capacity>50</ns3:capacity>
                  <ns3:used>15</ns3:used>
                  <ns3:dependencyListType>
                     <ns3:dependency type="use" used="15" mandatory="false"
bidirectional="false">
                        <ns3:nodeA>
                           <ns3:productCode>SN-0001</ns3:productCode>
                        </ns3:nodeA>
                        <ns3:nodeZ>
                           <ns3:productCode>SN-0002</ns3:productCode>
                        </ns3:nodeZ>
                     </ns3:dependency>
                  </ns3:dependencyListType>
               </ns3:subscriptionStat>
               <ns3:subscriptionStat>
                  <ns3:productCode>SN-0002</ns3:productCode>
                  <ns3:capacity>55</ns3:capacity>
                  <ns3:used>15</ns3:used>
                  <ns3:dependencyListType>
                     <ns3:dependency type="use" used="15" mandatory="true"
bidirectional="false">
                        <ns3:nodeA>
                           <ns3:productCode>SN-0002</ns3:productCode>
                        </ns3:nodeA>
                        <ns3:nodeZ>
                           <ns3:references>
                              <ns3:productCode>SN-0003</ns3:productCode>
                              <ns3:componentLabel>DSL Service 1.0</ns3:componentLabel>
                           </ns3:references>
                        </ns3:nodeZ>
                     </ns3:dependency>
                  </ns3:dependencyListType>
               </ns3:subscriptionStat>
            </ns3:subscriptionStats>
         </ns3:output>
      </ns3:resultDependencies>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:resultDependencies xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>query</ns3:operation>
            <ns3:description>Error</ns3:description>
            <ns3:resultCode>ERROR</ns3:resultCode>
            <ns3:error>
               <ns3:errorCode>100021</ns3:errorCode>
               <ns3:errorDescription>Request is empty</ns3:errorDescription>
            </ns3:error>
         </ns3:status>
      </ns3:resultDependencies>
   </env:Body>
</env:Envelope>
```

5.2.1.27. Request Setter Specification Properties

This operation can set properties and characteristics in an existing specification without having to modify the whole specification. NOTE: characteristics must be prefix by “characteristic:” to distinguish between properties and characteristics.

Description:
It can set those requested properties or characteristics requested in an existing specification.

Request parameters:

- *A set of specification name and version with their own properties and their values to update.*

SOAP API Format:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:spec="http://www.sr.om.hp.com/Specifications">
   <soap:Header/>
   <soap:Body>
      <spec:requestSetterSpecificationProperties>
         <spec:requestSetterProperties>
            <spec:name>DSLService</spec:name>
            <spec:version>1.0</spec:version>
            <spec:properties>
               <spec:property>
                  <spec:name>characteristic:upSpeed</spec:name>
                  <spec:propertyValues>
                     <spec:value>300</spec:value>
                  </spec:propertyValues>
               </spec:property>
            </spec:properties>
         </spec:requestSetterProperties>
         <spec:requestSetterProperties>
            <spec:name>DSLProduct</spec:name>
            <spec:version>1.0</spec:version>
            <spec:properties>
               <spec:property>
                  <spec:name>resource</spec:name>
                  <spec:propertyValues>
                       <spec:value>true</spec:value>
                  </spec:propertyValues>
               </spec:property>
               <spec:property>
                  <spec:name>capacity</spec:name>
                  <spec:propertyValues>
                              <spec:value>57</spec:value>
                  </spec:propertyValues>
               </spec:property>
            </spec:properties>
         </spec:requestSetterProperties>
      </spec:requestSetterSpecificationProperties>
   </soap:Body>
</soap:Envelope>
```

Response
Generate a **Result** response with the response status.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
```

```
      <ns4:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns4:status>
            <ns4:operation>setProperties</ns4:operation>
            <ns4:description>Success</ns4:description>
            <ns4:resultCode>OK</ns4:resultCode>
         </ns4:status>
      </ns4:result>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns4:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns4:status>
            <ns4:operation>setProperties</ns4:operation>
            <ns4:description>Error: The property characteristic:manufacturer does not exist in
specification DSLProduct 1.0.</ns4:description>
            <ns4:resultCode>ERROR</ns4:resultCode>
            <ns4:error>
               <ns4:errorCode>100203</ns4:errorCode>
               <ns4:errorDescription>The property characteristic:manufacturer does not exist in
specification DSLProduct 1.0.</ns4:errorDescription>
            </ns4:error>
         </ns4:status>
      </ns4:result>
   </env:Body>
</env:Envelope>
```

5.2.1.28. Request Getter Specification Properties

This operation can retrieve only properties and characteristics from a specification. NOTE: characteristics must be prefix by “characteristic:” to distinguish between properties and characteristics.

Description:
It can retrive those requested properties or characteristics from a specification.

Request parameters:

- *A set of specification name and version with its own properties to retrieve.*

SOAP API Format:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:spec="http://www.sr.om.hp.com/Specifications">
   <soap:Header/>
   <soap:Body>
      <spec:requestGetterSpecificationProperties>
         <spec:requestGetterProperties>
            <spec:name>DSLService</spec:name>
            <spec:version>1.0</spec:version>
            <spec:properties>
               <spec:property>
                  <spec:name>characteristic:upSpeed</spec:name>
               </spec:property>
            </spec:properties>
         </spec:requestGetterProperties>
         <spec:requestGetterProperties>
            <spec:name>DSLProduct</spec:name>
            <spec:version>1.0</spec:version>
            <spec:properties>
               <spec:property>
                  <spec:name>resource</spec:name>
               </spec:property>
               <spec:property>
                  <spec:name>capacity</spec:name>
               </spec:property>
            </spec:properties>
         </spec:requestGetterProperties>
      </spec:requestGetterSpecificationProperties>
   </soap:Body>
</soap:Envelope>
```

Response
Generate a **Result** response with the response status and the search result output.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns4:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns4:status>
            <ns4:operation>getProperties</ns4:operation>
            <ns4:description>Success</ns4:description>
            <ns4:resultCode>OK</ns4:resultCode>
         </ns4:status>
         <ns4:output>
            <ns4:responseGetterProperties>
               <ns4:name>DSLService</ns4:name>
               <ns4:version>1.0</ns4:version>
               <ns4:properties>
                  <ns4:property>
                     <ns4:name>characteristic:upSpeed</ns4:name>
                     <ns4:propertyValue>
                        <ns4:value>null</ns4:value>
                     </ns4:propertyValue>
                  </ns4:property>
               </ns4:properties>
            </ns4:responseGetterProperties>
            <ns4:responseGetterProperties>
               <ns4:name>DSLProduct</ns4:name>
               <ns4:version>1.0</ns4:version>
               <ns4:properties>
                  <ns4:property>
                     <ns4:name>resource</ns4:name>
                     <ns4:propertyValue>
                        <ns4:value>true</ns4:value>
                     </ns4:propertyValue>
                  </ns4:property>
                  <ns4:property>
                     <ns4:name>capacity</ns4:name>
                     <ns4:propertyValue>
                        <ns4:value>50</ns4:value>
                     </ns4:propertyValue>
                  </ns4:property>
               </ns4:properties>
            </ns4:responseGetterProperties>
         </ns4:output>
      </ns4:result>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns4:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns4:status>
            <ns4:operation>getProperties</ns4:operation>
            <ns4:description>Error: The property manufacturer does not exist in specification
DSLProduct 1.0</ns4:description>
            <ns4:resultCode>ERROR</ns4:resultCode>
            <ns4:error>
               <ns4:errorCode>100203</ns4:errorCode>
               <ns4:errorDescription>The property manufacturer does not exist in specification
DSLProduct 1.0</ns4:errorDescription>
            </ns4:error>
         </ns4:status>
      </ns4:result>
   </env:Body>
</env:Envelope>
```

5.2.1.29. Request Setter Subscription Properties

This operation can set properties and characteristics in an existing subscription. NOTE: characteristics must be prefix by “characteristic:” to distinguish between properties and characteristics.

Description:
It can set those requested properties or characteristics in an existing subscription.

Request parameters:

- *A set of requestSetterProperties containing productCodes and a set of properties/characteristics and their values to update.*

SOAP API Format:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soap:Header/>
   <soap:Body>
      <sub:requestSetterSubscriptionProperties>
         <sub:requestSetterProperties>
            <sub:productCode>SN-0001</sub:productCode>
            <sub:properties>
               <sub:property>
                  <sub:name>customerCode</sub:name>
                  <sub:value>Javi_updated</sub:value>
               </sub:property>
               <sub:property>
                  <sub:name>characteristic:productCharateristic</sub:name>
                  <sub:value>3236</sub:value>
               </sub:property>
               <sub:property>
                  <sub:name>characteristic:upSpeed</sub:name>
                  <sub:component>
                     <sub:name>DSLService</sub:name>
                     <sub:version>1.0</sub:version>
                  </sub:component>
                  <sub:value>120</sub:value>
               </sub:property>
               <sub:property>
                  <sub:name>processName</sub:name>
                  <sub:component>
                     <sub:name>DSLServiceStatistics</sub:name>
                     <sub:version>1.0</sub:version>
                  </sub:component>
                  <sub:value>Test_updated</sub:value>
               </sub:property>
            </sub:properties>
         </sub:requestSetterProperties>
      </sub:requestSetterSubscriptionProperties>
   <sub:requestGetterSubscriptionProperties/></soap:Body>
</soap:Envelope>
```

Response
Generate a **Result** response with the response status.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>setProperties</ns3:operation>
            <ns3:description>Success</ns3:description>
            <ns3:resultCode>OK</ns3:resultCode>
         </ns3:status>
      </ns3:result>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>setProperties</ns3:operation>
            <ns3:description>Error: The property processName type can not be multivalued in
component DSLServiceStatistics - 1.0 in subscription with productCode SN-0001</ns3:description>
            <ns3:resultCode>ERROR</ns3:resultCode>
            <ns3:error>
               <ns3:errorCode>100201</ns3:errorCode>
               <ns3:errorDescription>The property processName type can not be multivalued in
component DSLServiceStatistics - 1.0 in subscription with productCode SN-
0001</ns3:errorDescription>
            </ns3:error>
         </ns3:status>
      </ns3:result>
   </env:Body>
</env:Envelope>
```

5.2.1.30. Request Getter Subscription Properties

This operation can retrieve only properties and characteristics from a subscription. NOTE: characteristics must be prefix by “characteristic:” to distinguish between properties and characteristics.

Description:
It can retrive those requested properties or characteristics from a subscription.

Request parameters:

- *A set of requestGetterProperties containing productCodes and a set of properties/characteristics to retrieve.*

SOAP API Format:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soap:Header/>
   <soap:Body>
      <sub:requestGetterSubscriptionProperties>
         <sub:requestGetterProperties>
            <sub:productCode>SN-0001</sub:productCode>
            <sub:properties>
               <sub:property>
                  <sub:name>customerCode</sub:name>
               </sub:property>
               <sub:property>
                  <sub:name>characteristic:productCharateristic</sub:name>
               </sub:property>
               <sub:property>
                  <sub:name>characteristic:upSpeed</sub:name>
                  <sub:component>
                     <sub:name>DSLService</sub:name>
                     <sub:version>1.0</sub:version>
                  </sub:component>
               </sub:property>
               <sub:property>
                  <sub:name>processName</sub:name>
                  <sub:component>
                     <sub:name>DSLServiceStatistics</sub:name>
                     <sub:version>1.0</sub:version>
                  </sub:component>
               </sub:property>
             </sub:properties>
         </sub:requestGetterProperties>
      </sub:requestGetterSubscriptionProperties>
   </soap:Body>
</soap:Envelope>
```

Response

Generate a **Result** response with the response status and the search result output.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>getProperties</ns3:operation>
            <ns3:description>Success</ns3:description>
            <ns3:resultCode>OK</ns3:resultCode>
         </ns3:status>
         <ns3:output>
            <ns3:responseGetterProperties>
               <ns3:productCode>SN-0001</ns3:productCode>
               <ns3:properties>
                  <ns3:property>
                     <ns3:name>customerCode</ns3:name>
                     <ns3:value>Javi_updated</ns3:value>
                  </ns3:property>
                  <ns3:property>
                     <ns3:name>characteristic:productCharateristic</ns3:name>
                     <ns3:values>
                        <ns3:value>3236</ns3:value>
                     </ns3:values>
                  </ns3:property>
                  <ns3:property>
                     <ns3:name>characteristic:upSpeed</ns3:name>
                     <ns3:value>120</ns3:value>
                  </ns3:property>
                  <ns3:property>
                     <ns3:name>processName</ns3:name>
                     <ns3:value>Test_updated</ns3:value>
                  </ns3:property>
               </ns3:properties>
            </ns3:responseGetterProperties>
         </ns3:output>
      </ns3:result>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns3:result xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns4="http://www.sr.om.hp.com/Specifications">
         <ns3:status>
            <ns3:operation>getProperties</ns3:operation>
            <ns3:description>Error: The property customerCode2 does not exist in subscription
with productCodeSN-0001</ns3:description>
            <ns3:resultCode>ERROR</ns3:resultCode>
            <ns3:error>
               <ns3:errorCode>100199</ns3:errorCode>
               <ns3:errorDescription>The property customerCode2 does not exist in subscription
with productCodeSN-0001</ns3:errorDescription>
            </ns3:error>
         </ns3:status>
      </ns3:result>
   </env:Body>
</env:Envelope>
```

5.3. Management

Through this web service the user can invoke and manages the bulk processes, index the subscription information and obtain a general status of the SR System.

5.3.1. Management List

5.3.1.1. List Available Processes

Return a list of available processes.

Description:
This operation lists all available processes.

- *Without parameters.*

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:end="http://endpoint.sr.om.hp.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <end:listAvailableProcesses/>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

Generate a response with a Status tag which contains the result of the operation, a Total tag which contains the total number of available processes and Processes tag which contains the list of available processes.
An available process is defined by: the process description, the process name, transactionality (indicates if the process is transactional or not transactional) and the process type.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns1:listAvailableProcessesResponse xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns1="http://www.sr.om.hp.com/Processes">
         <status>
            <resultCode>OK</resultCode>
         </status>
         <processes>
            <process>
               <description>Index subscriptions in search engine. The default operation is ADD
and the default query returns all current subscriptions.</description>
               <name>indexation</name>
               <transactional>true</transactional>
               <type>INDEXATION</type>
            </process>
            <process>
               <description>Process export to csv</description>
               <name>export csv</name>
               <transactional>false</transactional>
               <type>EXPORT</type>
            </process>
         </processes>
         <total>2</total>
      </ns1:listAvailableProcessesResponse>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<status>
            <error>
               <errorCode>100043</errorCode>
               <errorDescription> Processes with the same name exists.</errorDescription>
            </error>
            <resultCode>ERROR</resultCode>
</status>
```

5.3.1.2. Execute Process By Name

Invoke for executing a process by name, without parameters. It returns the code identifying the job.

Description:
This operation executes a process identified by its name.

- *name (required)* : The process name.

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:end="http://endpoint.sr.om.hp.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <end:executeProcessByName>
         <name>export csv</name>
      </end:executeProcessByName>
   </soapenv:Body>
</soapenv:Envelope>
```

Response
Generate a response with a Status tag which contains the result of the operation and JobCode tag which contains the code identifying the execute job.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns1:executeProcessByNameResponse xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns1="http://www.sr.om.hp.com/Processes">
         <status>
            <resultCode>OK</resultCode>
         </status>
         <jobCode>100901 092731_export csv@1</jobCode>
      </ns1:executeProcessByNameResponse>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<status>
            <error>
               <errorCode>100044</errorCode>
               <errorDescription>There is no process with name: not found
process</errorDescription>
            </error>
            <resultCode>ERROR</resultCode>
</status>
```

5.3.1.3. Execute Process

Invoke for executing a process. It returns the code identifying the job.

Description:
This operation executes a process.

- *name (required)* : The process name.
- *params (optional)* : Hashtable containing the process params, consists of a list of entries key-value.

  Example with two parameters:

  ```
  <params>
              <entry>
                 <key>separator</key>
                 <value>;</value>
              </entry>
              <entry>
                 <key>filePath</key>
                 <value>.........</value>
              </entry>
  </params>
  ```

- *query (optional)* : Query indicates the group of subscriptions to work.

Filter Name	Required	Type	Default	Description	Example

historic	No	Boolean	False	Boolean indicating if the query should be performed into the Subscription_History table.	< historic >true</ historic >
indexed	No	Boolean	False	Boolean indicating if the subscription is indexed in Solr.	<indexed>true</indexed>
id	No	String	None	Subscription id.	<id>1</id>
productCode	No	String	None	Product serial number.	<productCode> SN-0001 </ productCode>
name	No	String	None	Subscription name.	<name>DSLService</name>
version	No	String	None	Subscription version.	<version>1.0</version>
customerCode	No	String	None	Customer.	<customerCode> Michael </ customerCode>
creationDate	No	Contains two children nodes (minValue,maxValue) as dates.	None	Creation date. Use two attributes (minValue and maxValue) to filter by date range.	<creationDate> <minValue> 2000-12-30T12:00:00 </minValue> <maxValue> 2001-01-01T12:00:00 </maxValue> </creationDate>
updateDate	No	Contains two children nodes (minValue,maxValue) as dates.	None	Last modification date. Use two attributes (minValue and maxValue) to filter by date range.	<updateDate <minValue> 2000-12-30T12:00:00 </minValue> <maxValue> 2001-01-01T12:00:00 </maxValue> </updateDate>

Example of query with interval in creationDate:

```
<query>
   <sub:creationDate>
      <sub:minValue>2001-12-30T12:00:00</sub:minValue>
      <sub:maxValue>2002-01-01T12:00:00</sub:maxValue>
   </sub:creationDate>
</query>
```

Example of query with customerCode:

```
<query>
   <sub:customerCode>Michael</sub:customerCode>
</query>
```

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:end="http://endpoint.sr.om.hp.com/" xmlns:sub="http://www.sr.om.hp.com/Subscriptions">
   <soapenv:Header/>
   <soapenv:Body>
      <end:executeProcess>
         <name>export csv</name>
          <params>
            <entry>
               <key>separator</key>
               <value>;</value>
            </entry>
         </params>
         <query>
            <sub:customerCode>Michael</sub:customerCode>
         </query>
      </end:executeProcess>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

Generate a response with a Status tag which contains the result of the operation and JobCode tag which contains the code identifying the execute job.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns1:executeProcessByNameResponse xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns1="http://www.sr.om.hp.com/Processes">
         <status>
            <resultCode>OK</resultCode>
         </status>
         <jobCode>100901 092731_export csv@1</jobCode>
      </ns1:executeProcessByNameResponse>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<status>
            <error>
               <errorCode>100044</errorCode>
               <errorDescription>There is no process with name: not found
process</errorDescription>
            </error>
            <resultCode>ERROR</resultCode>
</status>
```

5.3.1.4. Finish Process

Invoke to force a process to finish.

Description:

This operation finishes a running bulk process.

- *jobCode (required)* : The code identifying the job.

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:end="http://endpoint.sr.om.hp.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <end:finishProcess>
         <jobcode>100901 092734_DummyProcess@8</jobcode>
      </end:finishProcess>
   </soapenv:Body>
</soapenv:Envelope>
```

Response

Generate a response with a Status tag which contains the result of the operation.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns1:finishProcessResponse xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns1="http://www.sr.om.hp.com/Processes">
         <status>
            <resultCode>OK</resultCode>
         </status>
      </ns1:finishProcessResponse>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<status>
            <error>
               <errorCode>100046</errorCode>
               <errorDescription>Could not find processResult with job code:
jobCode</errorDescription>
            </error>
            <resultCode>ERROR</resultCode>
</status>
```

5.3.1.5. Get Process Status

Obtain the status of the process execution represented by the job code.

Description:

This operation retrieves the process status.

- *jobCode (required)* : The code identifying the job.

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:end="http://endpoint.sr.om.hp.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <end:getProcessStatus>
         <jobcode>100901 092731_export csv@6</jobcode>
      </end:getProcessStatus>
   </soapenv:Body>
```

```
</soapenv:Envelope>
```

Response

Generate a response with a Status tag which contains the result of the operation, a ProcessesResult tag which contains the processResults retrieved (in this operation will always return zero or one) and a Total tag which contains the total number of processResults.

A processResult object represents information about the process execution. During the process execution it includes information about the current status and indicates the partial status. If the process execution is finished, then the processResult indicates the general status.

A processResult is defined by:

id	The process result id
initDate	The date when processing started
processedElements	The number of processed elements
processedError	The number of elements that failed to be processed
processedOK	The number of correctly processed elements
jobCode	The job code
processName	Process name
result	Process result code
status	Status of execution of the process: (INIT, RUNNING, FINISH)
processType	Type of process (SYSTEM, INDEXATION, MIGRATION, EXPORT)
launchType	Way of launch process (SCHEDULED, MANUAL)

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns1:getProcessStatusResponse xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns1="http://www.sr.om.hp.com/Processes">
         <status>
            <resultCode>OK</resultCode>
         </status>
         <processResults>
            <processResult>
               <id>6</id>
               <initDate>2010-09-01T09:27:31.562+02:00</initDate>
               <processedElements>0</processedElements>
               <processedError>0</processedError>
               <processedOK>0</processedOK>
               <jobCode>100901 092731_export csv@6</jobCode>
               <processName>export csv</processName>
               <result>OK</result>
               <status>RUNNING</status>
                <processType>EXPORT</processType>
                <launchType>MANUAL</launchType>
            </processResult>
         </processResults>
         <total>1</total>
      </ns1:getProcessStatusResponse>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<status>
            <error>
               <errorCode>100105</errorCode>
               <errorDescription>Could not find processResult to job with code 
noneCode</errorDescription>
            </error>
            <resultCode>ERROR</resultCode>
</status>
```

5.3.1.6. Get All Process Status

Obtain the status of all the process executions, the total number of results is limited.

Description:
This operation gets process status.

- *Without parameters.*

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope" 
xmlns:end="http://endpoint.sr.om.hp.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <end:getAllProcessStatus/>
   </soapenv:Body>
</soapenv:Envelope>
```

Response
Generate a response with a Status tag which contains the result of the operation, a ProcessesResult tag which contains the processResults retrieved and a Total tag which contains the total number of processResults.
A processResult object represents information about the process execution. During the process execution it includes information about the current status and indicates the partial status. If the process execution is finished, then the processResult indicates the general status.

A processResult is defined by:

id	The process result id
initDate	The date when processing started
processedElements	The number of processed elements
processedError	The number of elements that failed to be processed
processedOK	The number of correctly processed elements
jobCode	The job code
processName	Process name
result	Process result code
status	Status of execution of the process: (INIT, RUNNING, FINISH)

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
```

```
      <ns1:getAllProcessStatusResponse xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns1="http://www.sr.om.hp.com/Processes">
         <status>
            <resultCode>OK</resultCode>
         </status>
         <processResults>
            <processResult>
               <id>2</id>
               <initDate>2010-09-01T09:27:31+02:00</initDate>
               <processedElements>15</processedElements>
               <processedError>0</processedError>
               <processedOK>15</processedOK>
               <jobCode>100901 092731_export csv@3</jobCode>
               <processName>export csv</processName>
               <result>OK</result>
               <status>FINISH</status>
            </processResult>
            <processResult>
               <id>6</id>
               <initDate>2010-09-01T09:27:31+02:00</initDate>
               <processedElements>10</processedElements>
               <processedError>0</processedError>
               <processedOK>10</processedOK>
               <jobCode>100901 092731_export csv@6</jobCode>
               <processName>export csv</processName>
               <result>OK</result>
               <status>RUNNING</status>
            </processResult>
            .....
         </processResults>
         <total>7</total>
      </ns1:getAllProcessStatusResponse>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<status>
            <error>
               <errorCode>201038</errorCode>
               <errorDescription> SQL Error retrieving process result.</errorDescription>
            </error>
            <resultCode>ERROR</resultCode>
</status>
```

5.3.1.7. Clean Historic Processes

Removes form the data base the information of all already executed processes.

Description:
This operation clears historic processes.

- *Without parameters.*

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:end="http://endpoint.sr.om.hp.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <end:cleanHistoricProcesses/>
   </soapenv:Body>
```

```
</soapenv:Envelope>
```

Response
Generate a response with a Status tag which contains the result of the operation.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns1:cleanHistoricProcessesResponse xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns1="http://www.sr.om.hp.com/Processes">
         <status>
            <resultCode>OK</resultCode>
         </status>
      </ns1:cleanHistoricProcessesResponse>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<status>
            <error>
               <errorCode>201040</errorCode>
               <errorDescription> SQL Error deleting all process results with FINISH
status.</errorDescription>
            </error>
            <resultCode>ERROR</resultCode>
</status>
```

5.3.1.8. Execute Index Process

Execute the indexation of all pending subscriptions and specifications (including new, updated and deleted documents).

Description:
Operation to execute indexing process.

- *Without parameters.*

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:end="http://endpoint.sr.om.hp.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <end:executeIndexProcess/>
   </soapenv:Body>
</soapenv:Envelope>
```

Response
Generate a response with a Status tag which contains the result of the operation.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns1:executeIndexProcessResponse xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns1="http://www.sr.om.hp.com/Processes">
```

```
         <status>
            <resultCode>OK</resultCode>
         </status>
      </ns1:executeIndexProcessResponse>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<status>
            <error>
               <errorCode>100044</errorCode>
               <errorDescription>There is no process with name: indexation</errorDescription>
            </error>
            <resultCode>ERROR</resultCode>
</status>
```

5.3.1.9. Execute Reindex Process

Execute the reindexation of all subscriptions and specifications stored in the database (cleaning the index system before).

Description:
This operation executes reindexing process.

- *Without parameters.*

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:end="http://endpoint.sr.om.hp.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <end:executeReindexProcess/>
   </soapenv:Body>
</soapenv:Envelope>
```

Response
Generate a response with a Status tag which contains the result of the operation.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns1:executeReindexProcessResponse xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns1="http://www.sr.om.hp.com/Processes">
         <status>
            <resultCode>OK</resultCode>
         </status>
      </ns1:executeReindexProcessResponse>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<status>
            <error>
               <errorCode>100044</errorCode>
               <errorDescription>There is no process with name: indexation</errorDescription>
            </error>
            <resultCode>ERROR</resultCode>
</status>
```

5.3.1.10. Execute Specification Index Process

Execute the indexation of all pending specifications (including new, updated and deleted documents).

Description:
This operation executes indexing process.

- *Without parameters.*

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:end="http://endpoint.sr.om.hp.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <end:executeSpecificationIndexProcess/>
   </soapenv:Body>
</soapenv:Envelope>
```

Response
Generate a response with a Status tag which contains the result of the operation.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns1: executeSpecificationIndexProcessResponse xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Specifications"
xmlns:ns1="http://www.sr.om.hp.com/Processes">
         <status>
            <resultCode>OK</resultCode>
         </status>
      </ns1:executeSpecificationIndexProcessResponse>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<status>
            <error>
               <errorCode>100044</errorCode>
               <errorDescription>There is no process with name: indexation</errorDescription>
            </error>
            <resultCode>ERROR</resultCode>
</status>
```

5.3.1.11. Execute Subscription Index Process

Execute the indexation of all pending subscriptions (including new, updated and deleted documents).

Description:
This operation executes indexing process.

- *Without parameters.*

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:end="http://endpoint.sr.om.hp.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <end: executeSubscriptionIndexProcess/>
   </soapenv:Body>
</soapenv:Envelope>
```

Response
Generate a response with a Status tag which contains the result of the operation.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns1:executeSubscriptionIndexProcessResponse xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns1="http://www.sr.om.hp.com/Processes">
         <status>
            <resultCode>OK</resultCode>
         </status>
      </ns1:executeSubscriptionIndexProcessResponse>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<status>
            <error>
               <errorCode>100044</errorCode>
               <errorDescription>There is no process with name: indexation</errorDescription>
            </error>
            <resultCode>ERROR</resultCode>
</status>
```

5.3.1.12. Execute Specification Reindex Process

Execute the reindexation of all specifications stored in the database (cleaning the index system before).

Description:
This operation executes reindexing process.

- *Without parameters.*

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:end="http://endpoint.sr.om.hp.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <end: executeSpecificationReindexProcess/>
   </soapenv:Body>
</soapenv:Envelope>
```

Response
Generate a response with a Status tag which contains the result of the operation.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns1:executeSpecificationReindexProcessResponse xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Specifications"
xmlns:ns1="http://www.sr.om.hp.com/Processes">
         <status>
            <resultCode>OK</resultCode>
         </status>
      </ns1:executeSpecificationReindexProcessResponse>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<status>
            <error>
               <errorCode>100044</errorCode>
               <errorDescription>There is no process with name: indexation</errorDescription>
            </error>
            <resultCode>ERROR</resultCode>
</status>
```

5.3.1.13. Execute Subscription Reindex Process

Execute the reindexation of all subscriptions stored in the database (cleaning the index system before).

Description:
This operation executes reindexing process.

- *Without parameters.*

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:end="http://endpoint.sr.om.hp.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <end: executeSubscriptionReindexProcess/>
   </soapenv:Body>
</soapenv:Envelope>
```

Response
Generate a response with a Status tag which contains the result of the operation.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns1:executeSubscriptionReindexProcess xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns1="http://www.sr.om.hp.com/Processes">
         <status>
            <resultCode>OK</resultCode>
         </status>
      </ns1:executeSubscriptionReindexProcessResponse>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<status>
            <error>
               <errorCode>100044</errorCode>
               <errorDescription>There is no process with name: indexation</errorDescription>
            </error>
            <resultCode>ERROR</resultCode>
</status>
```

5.3.1.14. Get SR Status

Return the general status of the system, including the total number of specifications stored in the database, the total number of subscriptions stored in the database, the number of indexed subscriptions, the number of process running and the license expiration date.

Description:
This operation gets the subscription repository status.

- *Without parameters.*

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:end="http://endpoint.sr.om.hp.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <end:getSRStatus/>
   </soapenv:Body>
</soapenv:Envelope>
```

Response
Generate a response with a Status tag which contains the result of the operation and a srStatus.
A srStatus is defined by:

expirationDate	The expiration license date. If the product has a license without expiration date this information won't include.
indexingMode	Solr indexing mode. Two valid values: 'SCHEDULED' and 'ONLINE'.
lastSuccessfullyIndexation	The jobcode of last success process or 'UNKNOWN" in other case. (Only mandatory if indexingMode equal 'SCHEDULED').
numberProcessRunning	The number of process running.
numberSpecifications	The number of specifications in SR.
numberSubscriptionsInDatabase	The number of subscriptions in the database.
numberSubscriptionsInSolr	The number of subscriptions indexed in Solr.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns1:getSRStatusResponse xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns1="http://www.sr.om.hp.com/Processes">
         <status>
            <resultCode>OK</resultCode>
         </status>
         <srStatus>
            <expirationDate>2010-10-29T23:59:59+02:00</expirationDate>
            <indexingMode>SCHEDULED</indexingMode>
<lastSuccessfullyIndexation>110405124200_indexation@19</lastSuccessfullyIndexation>
            <numberProcessRunning>0</numberProcessRunning>
            <numberSpecifications>3</numberSpecifications>
            <numberSubscriptionsInDatabase>2</numberSubscriptionsInDatabase>
            <numberSubscriptionsInSolr>6</numberSubscriptionsInSolr>
         </srStatus>
      </ns1:getSRStatusResponse>
   </env:Body>
</env:Envelope>
```

If an exception occurs during the process execution, an error tag is added into Status tag. The error tag contains the errorDescription and the errorCode (For further information about error code, please see Appendix).

```
<status>
            <error>
               <errorCode>201089</errorCode>
               <errorDescription>SQL Error retrieving subscription count.</errorDescription>
            </error>
            <resultCode>ERROR</resultCode>
</status>
```

5.3.1.15. Get Configuration Parameter

Return the configuration parameter of the system.

Description:
This operation gets the configuration parameters.

- *Parameter name (Is user doesn't send parameter names, system returns all configuration parameters).*

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:end="http://endpoint.sr.om.hp.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <end:getConfigurationParameter>
         <!--Zero or more repetitions:-->
         <name>max_number_results</name>
      </end:getConfigurationParameter>
   </soapenv:Body>
</soapenv:Envelope>
```

Response
Generate a response with a list of configuration parameters.

A configurationParameters is defined by:

key	Configuration property name
value	Configuration property value.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns1:getConfigurationParameterResponse xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns1="http://www.sr.om.hp.com/Processes">
         <configurationParameters>
            <key>max_number_results</key>
            <value>100</value>
         </configurationParameters>
      </ns1:getConfigurationParameterResponse>
   </env:Body>
</env:Envelope>
/status>
```

5.3.1.16. Set Configuration Parameter

Set the configuration parameter of the system, if it´s possible. Method execution returns a result object with result status of operation. (If parameter exists, then updates is done; else insert operation is done).

Description:

This operation sets the configuration parameters.

- *Parameter name (mandatory).*
- *Parameter value (mandatory).*

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:end="http://endpoint.sr.om.hp.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <end:setConfigurationParameter>
         <configurationParameters>
            <key>max_number_results</key>
            <value>200</value>
         </configurationParameters>
      </end:setConfigurationParameter>
   </soapenv:Body>
</soapenv:Envelope>
```

Generate a response with the result information.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns1:setConfigurationParameterResponse xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns1="http://www.sr.om.hp.com/Processes">
         <status>
            <resultCode>OK</resultCode>
         </status>
      </ns1:setConfigurationParameterResponse>
   </env:Body>
</env:Envelope>
```

5.3.1.17. Reload Cache

Reload memory specification cache in node where is invoked. This method is invoked by other cluster nodes when any cache changes (by new or update specification in other node). This operation launchs a external process called “SpecificationCacheUpdater”.

Description:

- Operation does not require parameters.

SOAP API Format:

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"
xmlns:end="http://endpoint.sr.om.hp.com/">
   <soapenv:Header/>
   <soapenv:Body>
      <end:reloadCache/>
   </soapenv:Body>
</soapenv:Envelope>
```

Generate a response with the process launched information.

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header/>
   <env:Body>
      <ns1:reloadCacheResponse xmlns:ns2="http://endpoint.sr.om.hp.com/"
xmlns:ns3="http://www.sr.om.hp.com/Subscriptions"
xmlns:ns1="http://www.sr.om.hp.com/Processes">
         <status>
            <resultCode>OK</resultCode>
         </status>
         <jobCode>110623121228_SpecificationCacheUpdater@1</jobCode>
      </ns1:reloadCacheResponse>
   </env:Body>
```

```
</env:Envelope>
```

6. Appendix

6.1. Wsdls

6.1.1. Operations

You can access to the latest version in http://localhost:8080/subscriptionrepository/operations/?wsdl

6.1.2. Management

You can access to the latest version in http://localhost:8080/subscriptionrepository/management/?wsdl

6.2. Error codes

The error code adds additional information about the exception.
The code is divided into three parts:

- The first digit defines the system where exception occurred:

CODE	*DESCRIPTION*
0	UNKNOWN
1	SUBSCRIPTION RESPOSITORY
2	INVENTORY
3	SEARCH ENGINE (SOLR)

- The next two digits define the subsystem:

CODE	DESCRIPTION
00	UNKNOWN
01	ORACLE DBMS
02	MYSQL DBMS
03	SORL SEARCH ENGINE
04	POSTGRES PLUS A.S. DBMS

- The next three digits define the unique code for error identification.

CODE	*DESCRIPTION*
000	Unexpected exception
001	SQL Error deleting specification
002	SQL Error retrieving specification
003	SQL Error inserting specification
004	SQL Error updating specification
005	Failed to obtain last insertion id
006	Failed to parse product specification
007	Failed to parse service specification
008	Failed to parse resource specification
009	Could not obtain specification as XML
010	Specification already exists
011	Could not find specification
012	Could not find referenced specification
013	Duplicate characteristic name in specification definition
014	A field with the same name and different type already exists
015	Specification already exists
016	Tried to update non-existing field
017	Tried to update field while referenced by subscriptions
018	Field does not exist
019	Specification does not exist
020	Tried to update field while referenced by other specifications
021	Request is empty
022	Could not find specification to update
023	SQL Error inserting process result
024	Failed to obtain string representation of the given bean

025	Specification id should be set for cache operations
026	SQL Error inserting specification dependency
027	SQL Error deleting specification dependency
028	Cannot update specification while existing specifications reference it
029	SQL Error counting specification dependencies
030	Cannot delete specification while existing specifications reference it
031	Cannot delete specification while existing subscriptions reference it
032	Cannot update specification while existing subscriptions reference it
033	Unexpected error initiating process
034	Unexpected error executing process
035	Unexpected error finalizing process execution
036	Process stopped
037	SQL Error deleting process result
038	SQL Error retrieving process result
039	SQL Error updating process result
040	SQL Error deleting all process results with FINISH status
041	Could not find process result to update
042	Could not find process result to delete
043	There are processes with the same name
044	There is no process with name
045	There is another migration process in execution
046	Could not find process to finish with code
047	There is no query defined
048	The job is already running
049	Unexpected error executing process of loading jars from dynamic loading folders
050	Could not obtain advanced subscription query as XML
051	Could not obtain subscription query as XML
052	Could not obtain subscription as XML
053	Failed to parse subscription
054	Datasource for inventory system could not be instantiated
055	Indexing task for search engine could not be instantiated
056	Malformed URL for Solr server
057	Too many subscriptions have been found
058	There is no transaction
059	There is no specification registered with name [%s] and version [%s]
060	There are more than one specifications registered with name [%s] and version [%s]
061	This subscription does not verify the refered specification
062	There is a subscription with the same product code [%s]
063	The subscription could not be found in history
064	Could not find subscription with product code [%s]
065	Product Code not specified when trying to delete subscription
066	There are more than one subcriptions registered with product code [%s]
067	Could not find subscription with id [%s]
068	There is no subscription matching the query criteria in history
069	Too many subscriptions found
070	Too many subscriptions found in history
071	Could not find subscription
072	Connectivity problems with Solr server
073	I/O error with Solr

074	Unable to begin transaction
075	Unable to commit transaction
076	Unable to rollback transaction
077	Unable to end transaction
078	Type [%s] for characteristic [%s] is unknown
079	The string [%s] is not a composed field name
080	Type [%s] for field [%s] is unknown
081	Connection already in use
082	Transaction already in process
083	Could not stablish connection with Solr server
084	A new transaction cannot be set before the current one ends
085	Cannot set null transaction
086	Solr error retrieving subscriptions
087	SQL Error retrieving specification count
088	Unable to set transaction for newly created DAO
089	SQL Error retrieving subscription count
090	SQL Error inserting subscription
091	SQL Error deleting subscription
092	SQL sentence file [%s] could not be loaded
093	SQL Error retrieving subscription count
094	SQL Error updating subscription
095	Solr error retrieving subscription count
096	The pagination values are wrong
097	The required page size is greater than the maximum allowed number of results
098	Operation information for history entry was not provided
099	Could not find factory for system [%s] and type [%s]
100	Could not find system [%s]
101	System not specified
102	The maximum allowed number of results has been exceeded. Please adjust your query a bit more
103	Could not find subscription to update
104	Characteristic [%s] is unknown
105	Could not find processResult with job code.
106	Subscription does not exist
107	There isn't any license installed for this product
108	The license for this product has expirated
109	License checking task could not be instantiated
110	Optimization task for search engine could not be instantiated
111	The license checker couldn't be initialized successfully. (Verify that Jboss starts from correct command line base path (../jboss/bin))
112	The license isn't valid
113	The license couldn't be checked
114	Too many specifications found
115	Old specification type is incompatible with new specification type in specification with name [%s] and version [%s]
116	Subscription format is not correct. There are mandatory parameters with empty value: [%s]
117	There is another indexation process in execution.
118	SQL Error occurs during list of advancedQueryConf.

119	SQL Error inserting AdvanceQueryConf.
120	Error converting advanceQuery bean to XML.
121	SQL Error retrieving AdvanceQueryConf.
122	Error converting XML to advanceQuery bean.
123	SQL Error during AdvancedQueryConf delete.
124	You are using a forbidden name of characteristic.
125	SQL Error occurs during findAll configurations.
126	SQL Error occurs during findAll status.
127	SQL Error during configuration delete.
128	SQL Error during status delete.
129	SQL Error during configuration update.
130	SQL Error during status update.
131	SQL Error during configuration save.
132	SQL Error during status save.
133	SQL Error during configuration search.
134	SQL Error during status search.
135	Error on remote web service invocation to update specification cache in other cluster nodes.
136	Error during application start: it´s undefined mandatory configuration parameters to cluster mode starting.
137	Validation error:
138	This kind of process cannot be stopped.
139	Unplugged indexation checking task could not be instantiated
140	Error initializing database connection data.
141	Scope is not supported for System processes.
142	The plsql for specification [%s, %s] couldn't be generated successfully.
143	Information about %s couldn't be recovered.
144	SOLR Error retrieving count of type:
145	SOLR Error retrieving subscription.
146	SOLR error retrieving specifications.
147	Specification indexing task for search engine could not be instantiated
148	Could not obtain advanced specification query as XML.
149	Parent specification doesn´t exist.
150	Parent specification is of another type.
151	Error during specification validation.
152	Error during delete because there are specification whose parent is specification to delete.
153	There is another system process in execution.
154	Quartz process has incorrect trigger expression:
155	Error during subscription history update.
156	Specification cannot be updated because is not compatible with existent version.
157	Error during locking table.
158	Error during unlocking table.
159	SQL Error inserting attachment file (reading file content).
160	SQL Error inserting attachment file (into database).
161	SQL Error deleting attached file in the database.
162	SQL Error searching attached files.
163	SQL Error retrieving attached file.
164	SQL Error deleting attached file in the database (using fileNames).

165	There is not a file.
166	Error reading the attached file content to process.
167	Error processing the attached file content.
168	SQL It´s mandatory to complete at least one filter.
169	SQL Error deleting attached file.
170	Could not find the object labeled as [%s]
171	More than one object was found with the selected label [%s]
172	Invalid type. [Product,CustomerFacingService,ResourceFacingService,ResourceItem]
173	Bad request, missing or wrong parameters.
174	Root product cannot be updated or removed using this operation, please use "updateSubscription" instead.
175	Only one type of element is allowed for update.
176	Invalid element to be inserted inside [%s]
177	Specification usage must be DEFINITION or TEMPLATE.
178	Characteristic or facet search not available at levels 2 and 3 in EXCLUSIVE mode.
179	Characteristic search is partial available at LEVEL2 and EXCLUSIVE index mode. Component name is required.
180	Characteristic search is partial available at LEVEL3 and EXCLUSIVE index mode. Component name and version is required.
181	Specification name and version are required on specification [%s].
182	Pagination values cannot be negative.
183	Pagination FROM value must be smaller than TO value.
184	Pagination range [%s - %s] cannot be higher than the MaxResults parameter [%s].
185	SQL Error counting the subscriptions.
186	Selected default date [%s] does not match the selected pattern [%s].
187	Selected default date [%s] does not match does not match the default SOLR pattern [%s]
188	Parsing error in default SOLR Config.
189	Error I/O in default SOLR Config.
190	SAXException in default SOLR Config.
191	SQL Error deleting subscription dependency by ProductCode
192	SQL Error deleting subscription dependency.
193	SQL Error inserting subscription dependency.
194	SQL Error querying subscription dependency.
195	Multiple association is not supported for dependency management.
196	This functionality is disabled.
197	ProductCode not found in the request.
198	Component with label [%s] or name and version [%s] - [%s] not found in subscription with productCode [%s]
199	The property [%s] does not exist in subscription with productCode [%s]
200	The property [%s] type can not be multivalued in subscription with productCode [%s]
201	The property [%s] type can not be multivalued in component [%s] - [%s] in subscription with productCode [%s]
202	Specification name and version are required in the request.
203	The property [%s] does not exist in specification with name and version [%s] - [%s]
204	The property [%s] type can not be multivalued in specification with name and version [%s] - [%s]
205	More than one object was found with the selected name and version [%s] - [%s]
206	SQL Error updating USED field in subscription
207	SQL Error updating USED field in dependency subscription

208	Component not found with label [%s] in subscription with productCode [%s]
209	SQL Component not found with label [%s] in subscription with productCode [%s]
210	SQL Dependency type [%s] does not allowed use functionality.
211	SQL Dependency productCodeA, productCodeZ and used fields are mandatory in the request.
212	SQL Dependency productCodeA, productCodeZ, component label and used fields are mandatory in the request.
213	productCode and componentLabel are required in dependency references.
214	USED dependencies are not allowed to be created for subscription with resource flag set to false.
215	Found USED dependencies in the system while creating a subscription with resource flag set to false.
216	Property resource in subscription is not allowed to be modified
217	SQL Error retrieving used and capacity from subscription with productCode [%s]
218	Node A list and node Z list are mandatory for for creating or updating dependencies
219	SQL The capacity for the resource [%s] has been exceeded. Actual capacity [%s] and pretended used [%s].
220	SQL Unique constraint violated in dependencies [nodeA, nodeZ, componentLabel, type]
221	Range does not allowed minimum or maximum value empty.
222	Characteristic value [%s] does not match the selected pattern [%s] for characteristic [%s].