

INTERCOPE



Version 3 Release 23 (V3R23)

A Comprehensive Guide to Box Web Services

Revision 1

Table of Contents

1	INTRODUCTION	1
1.1	DELIVERY	1
1.2	REQUIREMENTS	1
1.3	TEMINOLOGY	2
1.4	CONFUIRATION AND WAR FILE PREPARATION	2
1.4.1	<i>Configuration Files</i>	2
1.4.2	<i>Logging</i>	2
1.4.2.1	Log4j2.xml	2
1.4.2.2	Exemplary log4j2.xml Configuration	2
2	ARCHIVE RETRIEVAL WEB SERVICE	4
2.1	INTRODUCTION	4
2.2	CONFIGURATION	5
2.2.1	<i>BoxArchiveRetrievalService.properties</i>	5
2.2.1.1	Exemplary configuration	5
2.2.1.2	Parameters	5
2.2.2	<i>Configuration.properties</i>	6
2.2.2.1	Exemplary Configuration	6
2.2.2.2	Optional Parameters	6
2.2.3	<i>Log4j.xml</i>	7
2.2.4	<i>Initial Checks during the Start Up</i>	7
2.2.5	<i>Datasource</i>	7
2.2.6	<i>Log files</i>	7
2.3	OPERATIONS	8
2.3.1	<i>Login Operation</i>	8
2.3.1.1	Request	8
2.3.1.2	Response	8
2.3.2	<i>Simple SearchRequest Operation</i>	9
2.3.2.1	Request	9
2.3.2.2	Response (Excerpt)	9
2.3.3	<i>SwitchContextPath Operation</i>	9
2.3.3.1	Request	9
2.3.3.2	Response	10
2.3.4	<i>Logout Operation</i>	10
2.3.4.1	Request	10
2.3.4.2	Response	10
2.4	DOCUMENTATION	11
2.5	PROGRAMMATICAL STRUCTURE	14
2.5.1	<i>Outline</i>	14
2.5.2	<i>Generic Attributes</i>	15
2.5.3	<i>Login Operation</i>	16
2.5.3.1	login	16
2.5.3.2	LoginResponse	16
2.5.3.3	loginFault	16
2.5.4	<i>switchContextPath Operation</i>	16
2.5.4.1	switchContextPath	16
2.5.4.2	switchContextPathResponse	17
2.5.4.3	switchContextPathFault	17
2.5.5	<i>Search Operation</i>	17
2.5.5.1	searchRequest	17
2.5.5.2	SearchResponse	18
2.5.5.3	AbstractArchiveEntryType	18
2.5.5.4	TooManyResultsFaultType	18
2.5.5.5	ArchiveRetrievalFault	19
2.5.6	<i>Retrieve Operation</i>	19
2.5.6.1	retrieveRequest	19
2.5.6.2	RetrieveResponse	19

2.5.6.3	AbstractArchiveEntryType	20
2.5.6.4	EntryNotFoundRetrieveFaultType	20
2.5.6.5	ArchiveRetrievalFault.....	20
2.5.7	<i>Logout Operation</i>	20
2.5.7.1	logout.....	20
2.5.7.2	logoutResponse.....	21
2.5.7.3	FINArchiveEntryType.....	22
2.6	APPENDIX	23
2.6.1	<i>Box Archive Retrieval wds1</i>	23
3	BIC LOOKUP WEB SERVICE	40
3.1	INTRODUCTION	40
3.2	CONFIGURATION.....	41
3.2.1	<i>biclookup.properties</i>	41
3.2.1.1	Exemplary Configuration.....	41
3.2.2	<i>Log4j.xml</i>	41
3.3	OPERATIONS.....	42
3.3.1	<i>Looking Up if a BIC Exists</i>	42
3.3.1.1	Request	42
3.3.1.2	Response.....	42
3.3.1.3	Fault.....	42
3.3.2	<i>Looking Up the BIC Information</i>	43
3.3.2.1	Request	43
3.3.2.2	Response.....	43
3.4	DOCUMENTATION	44
3.5	PROGRAMMATICAL STRUCTURE	45
3.5.1	<i>Outline</i>	45
3.5.2	<i>Generic Attributes</i>	46
3.5.3	<i>existsBic Operation</i>	46
3.5.3.1	Request	46
3.5.3.2	Response.....	46
3.5.3.3	Fault.....	46
3.5.4	<i>lookupRecord Operation</i>	47
3.5.4.1	Request	47
3.5.4.2	Response.....	48
3.5.4.3	Fault.....	48
3.6	APPENDIX	49
3.6.1	<i>BicLookup wsdl</i>	49
4	BOX MESSAGING WEB SERVICE	54
4.1	INTRODUCTION.....	54
4.2	CONFIGURATION.....	55
4.2.1	<i>BoxMessagingService.properties</i>	55
4.2.1.1	Exemplary Configuration.....	56
4.2.2	<i>Log4j.xml</i>	57
4.3	OPERATIONS.....	57
4.3.1	<i>Login Operation</i>	57
4.3.1.1	Request	57
4.3.1.2	Response.....	57
4.3.1.3	Fault.....	57
4.3.2	<i>switchContextPath Operation</i>	58
4.3.2.1	Request	58
4.3.2.2	Response.....	58
4.3.3	<i>routeMessageToAbRecipient Operation</i>	58
4.3.3.1	Request:	58
4.3.3.2	Response.....	59
4.3.3.2.1	Fault SOAP	59
4.3.3.2.2	Fault Java Stack.....	60
4.3.4	<i>logout Operation</i>	60
4.3.4.1	Request	60
4.3.4.2	Response:.....	60
4.4	DOCUMENTATION	62

4.5	PROGRAMMATICAL STRUCTURE	64
4.5.1	Outline.....	64
4.5.2	Generic Attributes	65
4.5.3	Login Operation	66
4.5.3.1	Login.....	66
4.5.3.2	LoginResponse.....	66
4.5.3.3	loginFault.....	66
4.5.4	switchContextPath Operation	66
4.5.4.1	switchContextPath.....	66
4.5.4.2	switchContextPathResponse.....	67
4.5.4.3	switchContextPathFault.....	67
4.5.5	routeMessageToAbRecipient Operation	67
4.5.5.1	Request.....	67
4.5.5.2	Response.....	68
4.5.5.3	Fault.....	68
4.6	APPENDIX	68
4.6.1	Box Message Service wds/.....	68
5	FIN MESSAGING WEB SERVICE (RJE FORMATTED FILES).....	75
5.1	INTRODUCTION.....	75
5.2	CONFIGURATION.....	75
5.2.1	FINMessageService Configuration.....	75
5.2.1.1	FINMessageService.properties.....	75
5.2.1.1.1	Exemplary configuration.....	75
5.2.2	Log4j.xml.....	76
5.3	OPERATIONS.....	76
5.3.1	validateRJE Operation.....	76
5.3.1.1	Request.....	76
5.3.1.2	Response (all messages in RJE file correct)	77
5.3.2	sendRJE Operation	77
5.3.2.1	Positive Response.....	77
5.3.2.2	Response Status OK.....	77
5.3.2.3	MQ Message Processing.....	77
5.3.2.4	Negative Response.....	78
5.4	DOCUMENTATION	79
5.5	PROGRAMMATICAL STRUCTURE	80
5.5.1	Outline.....	80
5.5.2	Generic Attributes	81
5.5.3	ValidateRJE Operation	81
5.5.3.1	Request.....	81
5.5.3.2	Response.....	81
5.5.3.3	Fault.....	82
5.5.4	sendRJE	82
5.5.4.1	Request.....	82
5.5.4.2	Response.....	82
5.5.4.3	Fault.....	82
5.6	APPENDIX	83
5.6.1	Parameters	83
5.6.2	FIN Message wds/.....	84
6	FIN MESSAGE VALIDATION WEB SERVICE.....	88
6.1	INTRODUCTION.....	88
6.2	CONFIGURATION.....	89
6.2.1	validation.properties.....	89
6.2.1.1	Exemplary Configuration.....	90
6.2.2	Log4j.xml.....	90
6.3	OPERATION.....	90
6.3.1	FINMessageValidation Operation.....	90
6.3.1.1	Request.....	90
6.3.1.2	Negative Response.....	91
6.3.1.3	Positive Response.....	91
6.4	DOCUMENTATION	92

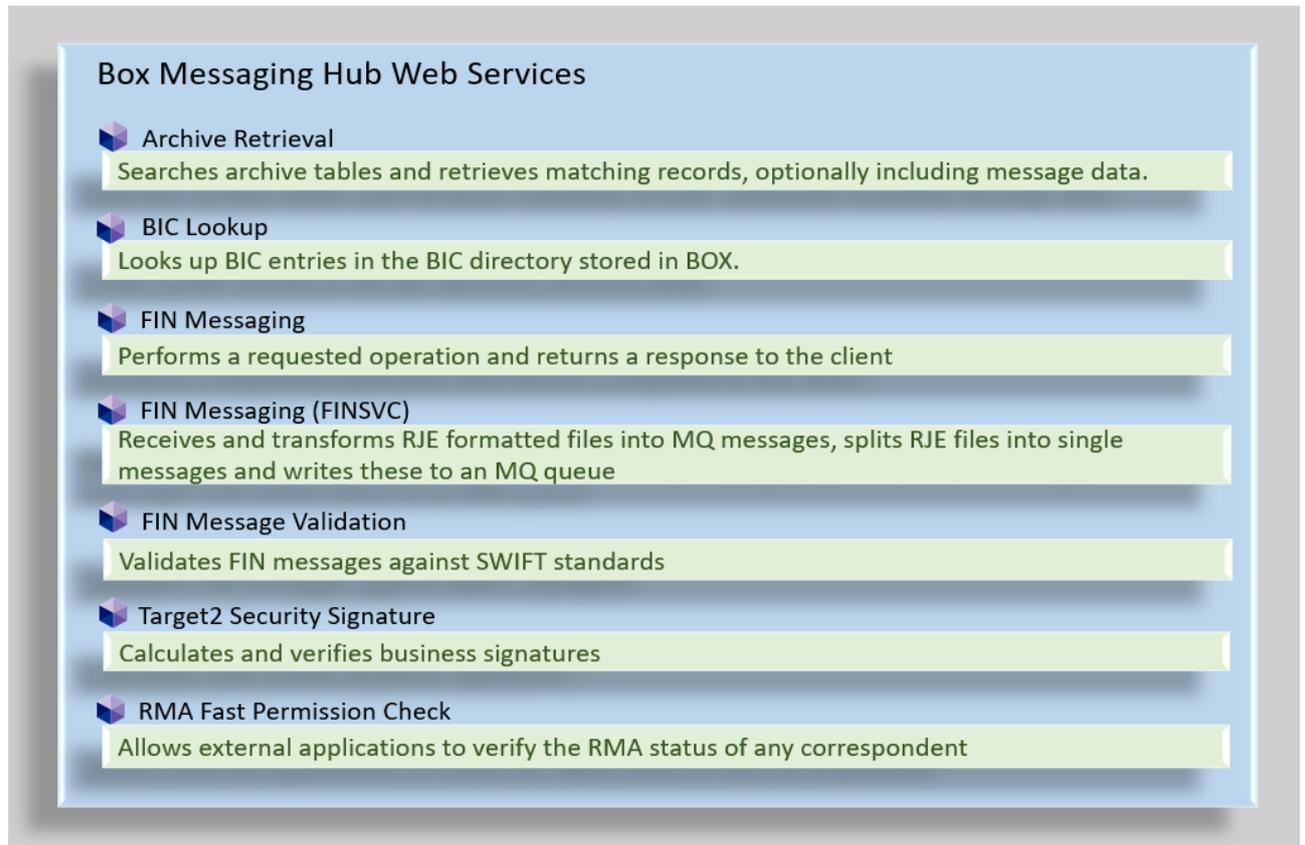
6.5	PROGRAMMATICAL STRUCTURE	93
6.5.1	Outline.....	93
6.5.2	Generic Attributes	93
6.5.2.1	Validation	93
6.5.2.1.1	ValidateRequest.....	93
6.5.2.1.2	ValidateResponse	93
6.5.2.1.3	Validation Fault.....	95
6.6	APPENDIX	95
6.6.1	FIN Message Validation wdsI	95
7	RMA FAST PERMISSION CHECK WEB SERVICE	97
7.1	INTRODUCTION.....	97
7.2	CONFIGURATION.....	97
7.2.1	<i>databaseloader.properties</i>	98
7.2.2	<i>Connection Pool</i>	98
7.2.3	<i>RMA Framework</i>	98
7.2.3.1	Configuration of the RMA Web Application.....	99
7.2.3.1.1	<i>rmaserviceconfig.properties</i>	99
7.2.3.1.2	<i>databaseloader.properties</i>	99
7.2.3.1.3	<i>Log4j.xml</i>	100
7.2.3.1.4	<i>Interface Format</i>	100
7.3	OPERATIONS.....	101
7.3.1	<i>checkNonFINRMAPermissionRequest Operation</i>	101
7.3.1.1	Request	101
7.3.1.2	Response.....	101
7.3.1.3	Fault.....	101
7.3.2	<i>checkRMAPermissionRequest Operation</i>	102
7.3.2.1	Request	102
7.3.2.2	Response.....	102
7.3.2.3	Fault.....	102
7.4	DOCUMENTATION	103
7.5	PROGRAMMATICAL STRUCTURE	104
7.5.1	Outline.....	104
7.5.2	Generic Attributes	105
7.5.2.1	<i>checkRMAPermission</i>	105
7.5.2.1.1	Request.....	105
7.5.2.1.2	Response	106
7.5.2.1.3	Fault	106
7.5.2.2	<i>checkNonFINRMAPermission</i>	106
7.5.2.2.1	Request.....	106
7.5.2.2.2	Response	106
7.5.2.2.3	Fault	106
7.6	APPENDIX	107
7.6.1	<i>RMA Fast Permission Check Web Service wdsI</i>	107
8	TARGET2 SECURITIES SIGNATURE VALIDATION WEB SERVICE	110
8.1	INTRODUCTION.....	110
8.2	CONFIGURATION.....	110
8.3	OPERATIONS.....	111
8.4	DOCUMENTATION	111
8.5	PROGRAMMATICAL STRUCTURE	113
8.5.1	Outline.....	113
8.5.2	Generic Attributes	114
8.5.3	<i>Login Operation</i>	115
8.5.3.1	login.....	115
8.5.3.2	LoginResponse.....	115
8.5.3.3	loginFault	115
8.5.3.4	Logout Operation	115
8.5.3.5	logoutResponse.....	116
8.5.4	<i>computeSignature</i>	116
8.5.5	<i>validateSignature</i>	116
8.6	APPENDIX	117

8.6.1	<i>Target2 Securities Signature Validation wdsI</i>	117
9	DISCLAIMER	123

1 Introduction

Intercope provides a number of Web Services supported by Apache Tomcat®, IBM WebSphere Application Server (WAS), JBOSS EAP, Oracle WebLogic Server and related application servers.

The following graph shows the available web services and their purpose:



The documentation describes all available web services in detail and gives an overview of functionality and configuration specialties, if applicable. A general introduction describes the purpose of the respective service.

1.1 Delivery

All Web Services are delivered in a zip file containing, depending on the Web Application Server (WAS) either a zipped war or ear file. The deployment follows the standard rules of the respective WAS. Please refer to the applicable documentation and specifications.

1.2 Requirements

Java 8 is required to run Web Services delivered with Version3 Release 23

BOX web services are compatible with:

- Apache Tomcat
- IBM WebSphere Application Server
- JBOSS
- Oracle WebLogic

The only WAS with MQ capabilities by default is the IBM WAS. If you plan to run the web service

using another WAS you need to include the following MQ jar files to the class path:

- connector.jar
- com.ibm.mq.jar
- com.ibm.mq.headers.jar
- com.ibm.mq.pcf.jar
- com.ibm.mq.jmqi.jar
- com.ibm.mq.commonservices.jar

1.3 Terminology

In this document the abbreviation BOX for **BOX Messaging Hub** and MQ for WebSphere MQ Series are used.

1.4 Configuration and War File Preparation

1.4.1 Configuration Files

Each web service comes with different number of configuration files specific to its purpose. A database connection is configured either by file (configuration.properties) or JNDI connection pool. Parameters and an exemplary configuration are described within the respective chapter.

Important

Changed configuration files need to be signed and updated in the security zip file delivered as part of the web service. Please also refer to the Security Implementation Guide (Security_Implementation_Guide.pdf) for further information.

1.4.2 Logging

1.4.2.1 Log4j2.xml

The log file name is by default set to '<web service>.log', the name, as well as further properties, can be changed according to respective requirements. The following sections refer to the configuration of the file's properties.

The log4j2.xml file is designed to configure the level of information written to log files, the location of these files and the properties such as size and amount of back-up filenames.

Example FIN Message Validation Web Service

'logDirectory' and 'baseName'

```
<Properties>
  <Property name="logDirectory">logs</Property>
  <Property name="baseName">finmsgval-service</Property>
</Properties>
```

The baseName, configured as 'finmsgval-service' is by default the respective log file name, and 'logs' the local directory, to which these files are written.

1.4.2.2 Exemplary log4j2.xml Configuration

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration shutdownHook="disable">
  <Properties>
    <Property name="logDirectory">logs</Property>
    <Property name="baseName">finmsgval-service</Property>
  </Properties>
```

```
<Appenders>
  <RollingFile name="mainAppender"
fileName="${logDirectory}${baseName}.log"
filePattern="${logDirectory}${baseName}-%i.log" append="false">
  <PatternLayout pattern="%d{ISO8601} ## %-20.30t ## %.30x ## %-20.50l ##
%-6p: %m%n" />
  <Policies>
    <OnStartupTriggeringPolicy />
    <SizeBasedTriggeringPolicy size="50 MB" />
  </Policies>
  <DefaultRolloverStrategy max="3" />
</RollingFile>
</Appenders>
<Loggers>
  <Logger name="com.intercope.box.validation" level="INFO"
additivity="false">
  <AppenderRef ref="mainAppender" />
</Logger>
  <Logger name="com.intercope.mpo.rma.ws.validation" level="INFO"
additivity="false">
  <AppenderRef ref="mainAppender" />
</Logger>
  <Root level="error">
  <AppenderRef ref="mainAppender"/>
</Root>
</Loggers>
</Configuration>
```

Important

The usage of tokens in the log4j2.xml is not possible!

2 Archive Retrieval Web Service

2.1 Introduction

This new web service allows to search archive tables and retrieve matching records, optionally including message data. The number of returned records is limited (default 100) and the search operation will return an error in case the limit is exceeded.

The archive data type mapping can be specified, hence keys are mapped correctly according to the search criteria and returned entries are formatted. This web service is currently implemented for FIN service only, as a generic archive search for FIN Wire search. Any column of the FIN warehouse can be used as a filter attribute. For other message types, the generic Attribute Set is to be used.

Whereby chapter 2 and 3 line out the implementation, configuration and a brief excursion into the request/response structure of exemplary operations, chapter 4 comprises detailed information regarding the programmatical structure of the web service to enable the developer to generate a suitable application connecting to the web service and making use of its operations.

Requirements

- ◆ FIN Archive as search repository
- ◆ FIN Wire must be stored in the archive

The Archive Retrieval Service is delivered in a zip file containing the `BoxArchiveRetrievalService.war` file.

2.2 Configuration

The configuration for the **BoxArchiveRetrievalService** is performed in two files, which are included in the WAR file "**BoxArchiveRetrievalService.war**".

- `BoxArchiveRetrievalService.properties`
- `Configuration.properties`
- `log4j2.xml`

Within `../WEB-INF/classes` the files `BoxArchiveSearchService.properties` and `configuration.properties` need to be adapted to suit the respective environment. Please also refer to the configuration guide `box_cfg_<Release>.pdf` for further information on the parameters of the file `configuration.properties`.

2.2.1 BoxArchiveRetrievalService.properties

2.2.1.1 Exemplary configuration

```
bar.api.configuration=configuration.properties
bar.replacement-file.resolve-via=path
bar.replacement-file.name=replace.rpl
bar.session.maxIdleTimeMillis=600000
bar.api.security-zip=security.zip
```

2.2.1.2 Parameters

- **bar.api.configuration**
Definition of the Box Messaging Service configuration file.
 - Type: String
 - Required: No**Example:**

```
bar.api.configuration=configuration.properties
```
- **bar.replacement-file.resolve-via**
Definition of how the replacement file is resolved. If this parameter is not set, replacements will not be used.
 - Type: String
 - Required: No**Possible values:**
 - **classpath** - use classpath to find replacement file
 - **path** - use filename directly**Example:**

```
bar.replacement-file.resolve-via=classpath
```
- **bar.replacement-file.name**
Definition of the replacement file name.
 - Type: String
 - Required: No**Example:**

```
bar.replacement-file.name=replace.rpl
```
- **bar.session.maxIdleTimeMillis**
Definition of the maximum Idle Time in milliseconds.
 - Type: Numeric

- Required: No

Example:

```
bar.session.maxIdleTimeMillis=600000
```

➤ **bar.api.security-zip**

Definition of the path and filename of the 'security.zip', if omitted '/WEB-INF/security.zip' is used.

- Type: String
- Required: Yes

Example:

```
bar.api.security-zip=/path/and/filename/of/security.zip
```

➤ **bar.api.mek**

Definition of the path and filename of the 'mek.enc', if omitted the file in security.zip is used.

- Type: String
- Required: Yes

Example:

```
bar.api.mek=samplemek.enc
```

2.2.2 Configuration.properties

Within this customized web application configuration file, the database connection properties are configured. It reflects the default web service configuration file. This file is provided and preconfigured and may be changed to use replacement tokens. We strongly recommend using this sample configuration.properties file, since the unique ID, with which the service registers within the database is set here, in this case System.VMID=BOXBARVC-01.

2.2.2.1 Exemplary Configuration

Basic Configuration DB

```
DB.JDBCdriverName= $$R$DB_DATABASE_QUALIFIERS$
```

```
DB.username= $$R$DB_DATABASE_USERS$
```

```
DB.password= $$R$DB_DATABASE_ENC_PWD$
```

```
DB.DBSchema= $$R$DB_DATABASE_QUALIFIER$
```

```
DB.URL= $$R$DB_DATABASE_T4URL$
```

```
DB.JNDIConnectionPool=MPO_JDBC_POOL
```

```
DB.type= $$R$ DB_DATABASE_TYPE$
```

```
Security.ArchiveDataProtectionStartDate= $$R$ARCHIVEDATA_PROTECTION_START$
```

```
Security.NonStaticDataBootstrapEndDate= $$R$NONSTATICDATA_BOOTSTRAP_ENDDATE$
```

2.2.2.2 Optional Parameters

Bootstrap Phase

Two parameters are required, if a bootstrap phase should be set.

```
Security.ArchiveDataProtectionStartDate
```

```
Security.NonStaticDataBootstrapEndDate
```

2.2.3 Log4j.xml

Please refer to chapter 1.4.2 for further configuration information and configuration options.

2.2.4 Initial Checks during the Start Up

- Verifying integrity api configuration
- Verifying replacement properties
- looking up data source

Please note, the usage of the replacement file and the further configuration parameters, which are contained in the configuration.properties file are described in the BOX Administration Guide.

IMPORTANT

As of release V3R22.0 the web service requires a security.zip and signed configuration files. Please refer to the Security_Implementation_Guide_V3R23.pdf for further information on how to sign the configuration files.

2.2.5 Datasource

For the application to work, a jdbc datasource must be configured.

Exemplary configuration for Datasource name 'MPO_JDBC_POOL':

```
<library id="DB2JCCLib">
<fileset dir="/opt/ibm/db2/V10.5/java/" includes="db2jcc4.jar db2jcc.jar
db2jcc_license_cisuz.jar"/>
</library>

<dataSource id="MPO_JDBC_POOL" jndiName="MPO_JDBC_POOL">
  <jdbcDriver libraryRef="DB2JCCLib"/>
<properties.db2.jcc databaseName="OMEGA" serverName="192.168.123.456"
portNumber="50001" user="db2inst1" password="5EF993D9BAE860BBD361F2643A6A8C1A"
/>
  <jdbcDriver libraryRef="DB2JCCLib"/>
</dataSource>

<application id="BoxArchiveRetrievalService" name="BoxArchiveRetrievalService"
location="BoxArchiveRetrievalService.war">
</application>
```

2.2.6 Log files

The web service creates two log files, which can be initially used to check the deployment and correct security checks.

- bar.log
- bar-api.log

A correct start may be logged in bar.log as

```
# servlet init called
# initializing BOX Archive Retrieval Service ...
# BOX Archive Search Service configuration:
  max idle time of user session (bar.session.maxIdleTimeMillis): 600000
  replacement file resolve method (bar.replacement-file.resolve-via): path
  replacement file name (bar.replacement-file.resolve-via): replace.rpl
  security-zip file name (bar.api.security-zip): security.zip
# initialization completed.
```

The bar-api.log file keeps a recording of the called mpo-api<version>.jar with an excerpt of the start as

```
# verifying api configuration (data='file:configuration.properties',
sig='file:configuration.properties.sig.xml') ...
# integrity of api configuration data verified
# verifying replacement properties (data='file:replace.rpl',
sig='file:replace.rpl.sig.xml') ...
# integrity of replacement properties verified
# successfully initialized BSL with description 'xxxxxxxxxx'
# getConnection() called
# connection 139083423 received from pool
```

If the web service is correctly implemented, it can be called upon in a browser.

Example

<IP Address>:9080/BoxArchiveRetrievalService/BoxArchiveRetrieval

2.3 Operations

Prior to calling the search operations, a session token needs to be acquired through the 'login' operation, optionally followed by a call to 'switchContextPath'.

Once the login returned a session token, search and retrieve operations may be called upon. As a final step and to free up resources, a 'logout' should be requested to log the user out of the system.

In this chapter are some examples on how the request and response would look like. For details, please refer to chapter 2.4.

2.3.1 Login Operation

2.3.1.1 Request

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:box="http://ws.box.intercope.com/BoxArchiveRetrieval/"
  <soapenv:Header/>
  <soapenv:Body>
    <box:login>
      <clientPrefix>demofin</clientPrefix>
      <userName>demofin</userName>
      <secret>
        <plain></plain>
      </secret>
    </box:login>
  </soapenv:Body>
</soapenv:Envelope>
```

2.3.1.2 Response

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns0:loginResponse
xmlns:ns0="http://ws.box.intercope.com/BoxArchiveRetrieval/"
      <sessionId> b23091d59c1488d3cb10359005793b89</sessionId>
    </ns0:loginResponse>
  </S:Body>
</S:Envelope>
```

2.3.2 Simple SearchRequest Operation

2.3.2.1 Request

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:box="http://ws.box.intercope.com/BoxArchiveRetrieval/">
  <soapenv:Header/>
  <soapenv:Body>
    <box:searchRequest>
      <sessionId>70444b774a06820d522c0630d0071174</sessionId>
      <datasource>ARCHDEMO</datasource>
      <retrieveData>>false</retrieveData>
      <attributeTypeSet>FIN</attributeTypeSet>
      <limitResultSetSize>100</limitResultSetSize>
      <dateRange>
        <from>2010-01-01</from> <to>2019-01-01</to>
      </dateRange>
      <searchCondition>
        <box:stringPredicate key="mt" operator="=">
          <value>540</value>
        </box:stringPredicate>
      </searchCondition>
    </box:searchRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

2.3.2.2 Response (Excerpt)

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns0:searchResponse
xmlns:ns0="http://ws.box.intercope.com/BoxArchiveRetrieval/"
  <listResult count="5">
    <archiveEntry xsi:type="ns0:FINArchiveEntryType"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <entryId>447420</entryId>
      <mpsId>447420</mpsId>
      <creationDate>2016-11-25Z</creationDate>
      <mt>540</mt>
    </archiveEntry>
    [...]
    <receptionTime>2016-11-23T10:01:54.0Z</receptionTime>
    <receptionTimeOffset>60</receptionTimeOffset>
  </listResult>
</ns0:searchResponse>
  </S:Body>
</S:Envelope>
```

2.3.3 SwitchContextPath Operation

2.3.3.1 Request

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:box="http://ws.box.intercope.com/BoxArchiveRetrieval/">
  <soapenv:Header/>
  <soapenv:Body>
```

```
<box:switchContextPath>
  <sessionId>70444b774a06820d522c0630d0071174</sessionId>
  <contextName>centro/MessageEntry</contextName>
</box:switchContextPath>
</soapenv:Body>
</soapenv:Envelope>
```

2.3.3.2 Response

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:switchContextPathResponse
xmlns:ns2="http://ws.box.intercope.com/BoxArchiveRetrieval/">
      <sessionId>70444b774a06820d522c0630d0071174</sessionId>
    </ns2:switchContextPathResponse>
  </soap:Body>
</soap:Envelope>
```

2.3.4 Logout Operation

2.3.4.1 Request

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:box="http://ws.box.intercope.com/BoxArchiveRetrieval/">
  <soapenv:Header/>
  <soapenv:Body>
    <box:logout>
      <sessionId>b23091d59c1488d3cb10359005793b89</sessionId>
    </box:logout>
  </soapenv:Body>
</soapenv:Envelope>
```

2.3.4.2 Response

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns2:logoutResponse
xmlns:ns2="http://ws.box.intercope.com/BoxArchiveRetrieval/">
      <success>true</success>
    </ns2:logoutResponse>
  </soap:Body>
</soap:Envelope>
```

2.4 Documentation

Web Service: BoxArchiveRetrieval

Target: <http://ws.box.intercope.com/BoxArchiveRetrieval/>
Namespace:

*Port **BoxArchiveRetrievalSOAP** [Port type](#) [Source code](#)*

Location: <http://192.168.21.208:9080/BoxArchiveRetrievalService/BoxArchiveRetrieval>

Protocol: SOAP

Default style: document

Transport SOAP over HTTP

protocol:

Operations:

1. [login](#) [Detail](#) [Source code](#)
2. [logout](#) [Detail](#) [Source code](#)
3. [retrieve](#) [Detail](#) [Source code](#)
4. [search](#) [Detail](#) [Source code](#)
5. [switchContextPath](#) [Detail](#) [Source code](#)

1. login

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: <http://ws.box.intercope.com/BoxArchiveRetrieval/login>

Input: loginRequest (soap:body, use = literal) [Source code](#)

```
parameters type login
  ■ clientPrefix type string
  ■ userName type string
  ■ secret type SecretType
    ■ plain type string
      plain text password
```

Output: loginResponse (soap:body, use = literal) [Source code](#)

```
parameters type loginResponse
  ■ sessionId type SessionIdType - type string with restriction
```

Fault: loginFault (soap:fault, use = literal) [Source code](#)

```
parameters type loginFault
  ■ loginFault type string
```

2. logout

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: <http://ws.box.intercope.com/BoxArchiveRetrieval/logout>

Input: logoutRequest (soap:body, use = literal) [Source code](#)

```
parameters type logout
  ■ sessionId type SessionIdType - type string with restriction
```

Output: logoutResponse (soap:body, use = literal) [Source code](#)

```
parameters type logoutResponse
  ■ success type boolean
```

3. retrieve

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: <http://ws.box.intercope.com/BoxArchiveRetrieval/retrieve>

Input: retrieveRequest (soap:body, use = literal) [Source code](#)

```
request type retrieveRequest
  ■ sessionId type SessionIdType - type string with restriction
  ■ datasource type string
    shortname of journal data source (V_FINARCH_$CENTRO)
  ■ retrieveData type boolean
  ■ attributeTypeSet type ArchiveAttributeSetType - type string with restriction - enum { 'FIN' }
  ■ entryId type integer
```

Output: retrieveResponse (soap:body, use = literal) [Source code](#)

```
response type retrieveResponse
  ■ archiveEntry - optional; type AbstractArchiveEntryType
    ■ entryId type integer
    ■ mpsId type integer
    ■ msgType - optional; type int
    ■ queueEntryTime - optional; type dateTime
    ■ queueEntryTimeOffset - optional; type int
    ■ boxDisplayReference - optional; type string
    ■ comment - optional; type string
    ■ creationDate type date
    ■ data - optional; type ArchiveDataType
      ■ content type anyType
        ◦ encoding type DataEncodingType - type string with restriction - enum { 'plain-text', 'Base64', 'hex-ascii', 'cdata' }
```

Fault: retrieveFault (soap:fault, use = literal) [Source code](#)

```
notFound type EntryNotFoundRetrieveFaultType type string
```

Fault: retrieveFault1 (soap:fault, use = literal) [Source code](#)

```
error type ArchiveRetrievalFault type string
```

4. search

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: <http://ws.box.intercope.com/BoxArchiveRetrieval/search>

Input: searchRequest (soap:body, use = literal) [Source code](#)

```
request type searchRequest
  ■ sessionId type SessionIdType - type string with restriction
  ■ datasource type string
    shortname of journal data source (V_FINARCH_$CENTRO)
  ■ retrieveData type boolean
  ■ attributeTypeSet type ArchiveAttributeSetType - type string with restriction - enum { 'FIN' }
  ■ limitResultSetSize - optional; type integer
    if omitted a default of 100 is used
  ■ dateRange type DateRangeType
    ■ from type date
    ■ to - optional; type date
  ■ searchCondition type BlockPredicateType - extension of abstract type AbstractPredicateElementType
    ■ stringPredicate type stringPredicate
    ■ intPredicate type intPredicate
    ■ bigIntPredicate type bigIntPredicate
    ■ bigDecimalPredicate type bigDecimalPredicate
    ■ timestampPredicate type timestampPredicate
    ■ timePredicate type timePredicate
    ■ datePredicate type datePredicate
    ■ block type block
      ■ and type AndConditionType - extension of abstract type AbstractConditionType
      ■ or type OrConditionType - extension of abstract type AbstractConditionType
      ■ stringPredicate type stringPredicate
      ■ intPredicate type intPredicate
      ■ bigIntPredicate type bigIntPredicate
      ■ bigDecimalPredicate type bigDecimalPredicate
      ■ timestampPredicate type timestampPredicate
      ■ timePredicate type timePredicate
      ■ datePredicate type datePredicate
      ■ block type block
```

Output: searchResponse (soap:body, use = literal) [Source code](#)

```
response type searchResponse
  ■ listResult type ListResultType
    ■ archiveEntry - unbounded; type AbstractArchiveEntryType
      ■ entryId type integer
      ■ mpsId type integer
      ■ msgType - optional; type int
      ■ queueEntryTime - optional; type dateTime
      ■ queueEntryTimeOffset - optional; type int
      ■ boxDisplayReference - optional; type string
      ■ comment - optional; type string
      ■ creationDate type date
      ■ data - optional; type ArchiveDataType
        ■ contentType type anyType
        ○ encoding type DataEncodingType - type string with restriction - enum { 'plain-text', 'Base64', 'hex-ascii', 'cdata' }
    ○ count type integer
```

Fault: searchFault (soap:fault, use = literal) [Source code](#)

```
tooMany type TooManyResultsFaultType type string
```

Fault: searchFault1 (soap:fault, use = literal) [Source code](#)

```
error type ArchiveRetrievalFault type string
```

5. switchContextPath

[Source code](#)

Operation type: Request-response. The endpoint receives a message, and sends a correlated message.

SOAP action: http://ws.box.intercope.com/BoxArchiveRetrieval/switchContextPath

Input: switchContextPathRequest (soap:body, use = literal) [Source code](#)

```
parameters type switchContextPath
  ■ sessionId type SessionIdType - type string with restriction
  ■ contextName type string
```

Output: switchContextPathResponse (soap:body, use = literal) [Source code](#)

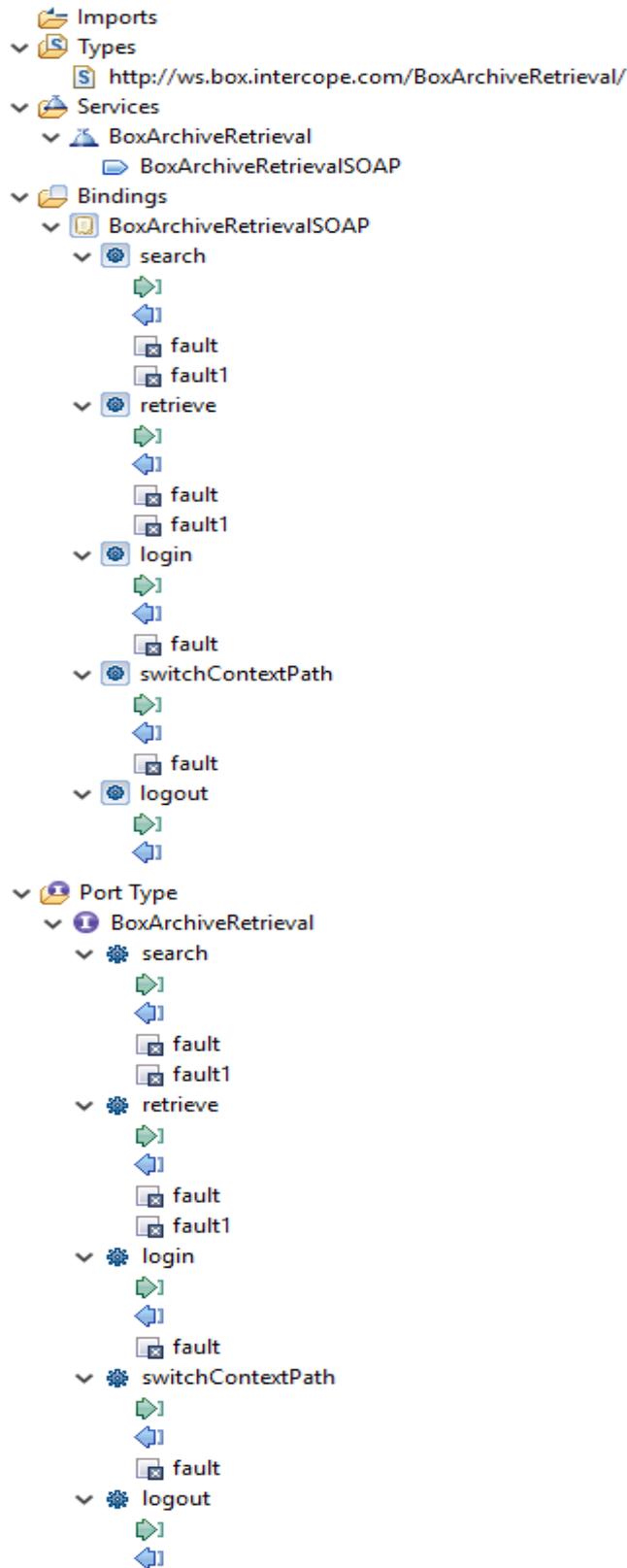
```
parameters type switchContextPathResponse
  ■ sessionId type SessionIdType - type string with restriction
```

Fault: switchContextPathFault (soap:fault, use = literal) [Source code](#)

```
parameters type switchContextPathFault
  ■ switchContextPathFault type string
```

2.5 Programmatical Structure

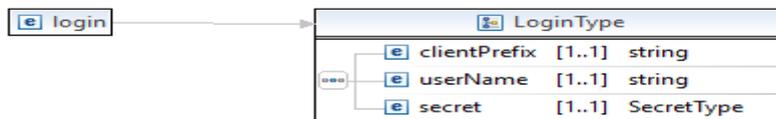
2.5.1 Outline



2.5.3 Login Operation

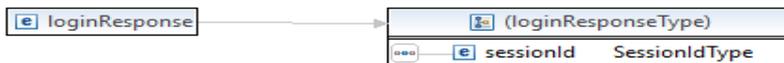
In order to perform “search/retrieve” from the archive, the application must first log in to the system.

2.5.3.1 login



clientPrefix	Client Prefix name of the institute.
userName	UserName of the account, which is used for the retrieval.
secret	Password for the account, it can be plain text or encrypted.

2.5.3.2 LoginResponse



sessionId	During the log-in, an actual user object is created. The session Id is the pointer to this object. if 'true' login operation will create a JSESSIONID which can be used by load balancers to correctly maintain session stickiness
-----------	--

2.5.3.3 loginFault

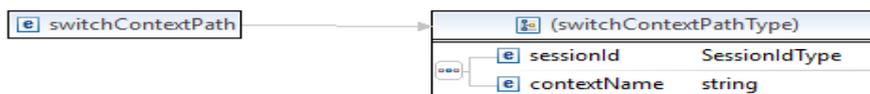
If an error occurs during log-in, the error text is returned.



2.5.4 switchContextPath Operation

Optional method, if multiple context paths for user object are defined. Use this method to select the user context.

2.5.4.1 switchContextPath



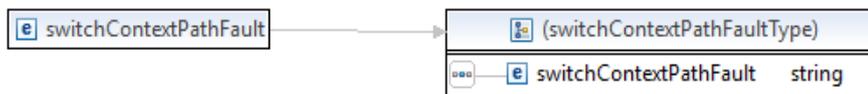
sessionId	During the log-in, an actual user object is created. The session Id is the pointer to this object. if 'true' login operation will create a JSESSIONID which can be used by load balancers to correctly maintain session stickiness
contextName	Name of the User Context Path.

2.5.4.2 switchContextPathResponse



sessionId	During the log-in, an actual user object is created. The session Id is the pointer to this object. If 'true' login operation will create a JSESSIONID which can be used by load balancers to correctly maintain session stickiness
-----------	--

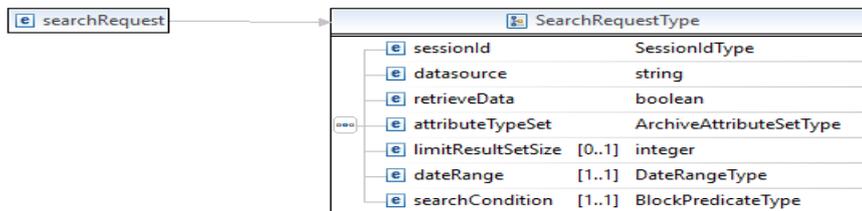
2.5.4.3 switchContextPathFault



switchContextPathFault	Error description of the switchContextPath failure.
------------------------	---

2.5.5 Search Operation

2.5.5.1 searchRequest



sessionId	During the log-in, an actual user object is created. The session Id is the pointer to this object. If 'true' login operation will create a JSESSIONID which can be used by load balancers to correctly maintain session stickiness
datasource	<u>Shortname</u> of journal data source(V xxxARCH \$cltprefix).
retrieveData	type="xsd:boolean" true/false.
attributeTypeSet	Type of attribute set: FIN or GENERIC.
limitResultSetSize	Number of entries to be returned with this query. If omitted, the default value "100" is used.
dataRange	Date filter for this query. Use "from and to" to specify the range: <pre><dateRange> <from>2017-09-01</from> <to>2017-10-31</to> </dateRange></pre>
searchCondition	Additional filter, which can be used to select the messages from the archive. All names belonging to the FINAttributeSet can be used, e.g.: <pre><searchCondition></pre>

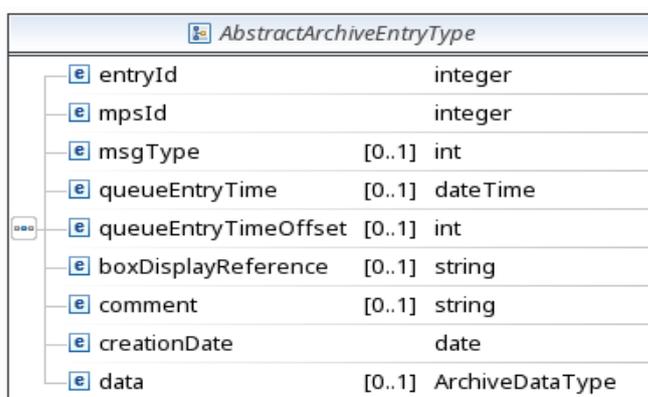
```
<box:stringPredicate key="mt" operator="=">
  <value>103</value>
</box:stringPredicate>
</searchCondition>
```

2.5.5.2 SearchResponse



listResultType	Sequence of AbstractArchiveEntryType. ArchiveAttributeSetType FIN will return FINArchiveEntryType (which extends the AbstractArchiveEntryType). See Chapter 4.
----------------	--

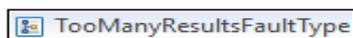
2.5.5.3 AbstractArchiveEntryType



entryId	Id of the archive entry. This Id is the primary key of the archive table. Normally returned by the search operation.
mpsId	Message Processing Id.
msgType	Type of message.
queueEntryTime	Time which the message enters the queue at.
queueEntryTimeOffset	Timezone offset of local time to UTC.
boxDisplayReference	Info generated by the application (MT type and TRN for FIN).
comment	Operator comment (optional textual comment).
creationDate	Message creation date.
data	ArchiveDataType: the message from the archive.

2.5.5.4 TooManyResultsFaultType

The service returns this failure notification, if the number of matching entries is higher than the number specified in limitResultSetSize.



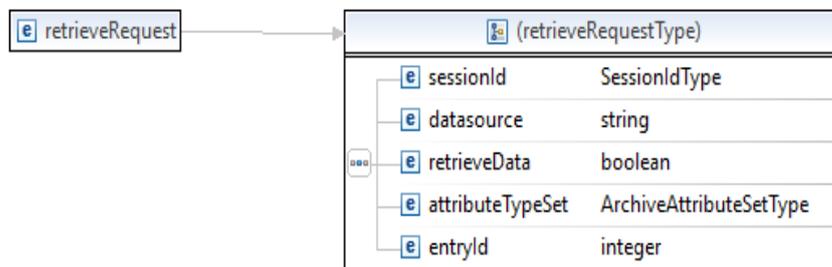
2.5.5.5 ArchiveRetrievalFault

The service returns this failure for all other errors.



2.5.6 Retrieve Operation

2.5.6.1 retrieveRequest



sessionId	During the log-in, an actual user object is created. The session ID is the pointer to this object.
datasource	<u>Shortname of journal data source(V_xxxARCH_ \$cltprefix).</u>
retrieveData	type="xsd:boolean" true/false.
attributeTypeSet	Type of attribute set: FIN or Generic.
entryId	Id of the archive entry. This Id is the primary key of the archive table. Normally returned by the search operation.

2.5.6.2 RetrieveResponse

ArchiveAttributeSetType FIN will return FINArchiveEntryType (which extends the AbstractArchiveEntryType). See Chapter 4.



2.5.6.3 AbstractArchiveEntryType

AbstractArchiveEntryType	
entryId	integer
mpsId	integer
msgType	[0..1] int
queueEntryTime	[0..1] dateTime
queueEntryTimeOffset	[0..1] int
boxDisplayReference	[0..1] string
comment	[0..1] string
creationDate	date
data	[0..1] ArchiveDataType

entryId	Id of the archive entry. This Id is the primary key of the archive table. Normally returned by the search operation
mpsId	Message Processing Id.
msgType	Type of message.
queueEntryTime	Time which the message enters the queue at.
queueEntryTimeOffset	Timezone offset of local time to UTC.
boxDisplayReference	Info generated by the application (MT type and TRN for FIN).
comment	Operator comment (optional textual comment).
creationDate	Message creation date.
data	ArchiveDataType: the message from the archive.

2.5.6.4 EntryNotFoundRetrieveFaultType

The service returns this failure, if the requested record doesn't exist in the database.

EntryNotFoundRetrieveFaultType

2.5.6.5 ArchiveRetrievalFault

The service returns this failure for all other errors.

ArchiveRetrievalFault

2.5.7 Logout Operation

2.5.7.1 logout

After completing the search / retrieval methods the application must logout. This will free all allocated resources for this user object.



sessionId	During the log-in, an actual user object is created. The session Id is the pointer to this object. If 'true' login operation will create a JSESSIONID which can be used by load balancers to correctly maintain session stickiness
-----------	--

2.5.7.2 logoutResponse



success	type="xsd:boolean" true/false.
---------	--------------------------------

2.5.7.3 FINArchiveEntryType

FINArchiveEntryType			
e	mt	[0..1]	string
e	mur	[0..1]	string
e	trn	[0..1]	string
e	currency	[0..1]	string
e	finCopyServiceIdentifier	[0..1]	string
e	bankingPriority	[0..1]	string
e	lt	[0..1]	string
e	correspondent	[0..1]	string
e	externalStatusDescription	[0..1]	string
e	originatorMessageId	[0..1]	string
e	receiverMessageId	[0..1]	string
e	messageReferenceId	[0..1]	string
e	displayReference	[0..1]	string
e	destinationDisplayReference	[0..1]	string
e	dllDisplayReference	[0..1]	string
e	sddOriginatorMessageId	[0..1]	string
e	sddReceiverMessageId	[0..1]	string
e	messageReferenceS	[0..1]	string
e	recipientOriginalDisplayAddress	[0..1]	string
e	recipientDisplayAddress	[0..1]	string
e	cbtAckNakCode	[0..1]	string
e	relatedReferenceNumber	[0..1]	string
e	securityCodeISIN	[0..1]	string
e	accountNumber	[0..1]	string
e	securitiesDepositAccount	[0..1]	string
e	orderingBank	[0..1]	string
e	beneficiaryBank	[0..1]	string
e	pdeFlag	[0..1]	string
e	bankOpCode	[0..1]	string
e	pdmFlag	[0..1]	string
e	lastActingUser	[0..1]	string
e	lastAckDelNotType	[0..1]	string
e	validationFlagFromUserHeader	[0..1]	string
e	rxChannelDisplayName	[0..1]	string
e	txChannelDisplayName	[0..1]	string
e	msgvalErrorCode	[0..1]	string
e	broadcastText	[0..1]	string
e	sourceApplicationName	[0..1]	string
e	targetApplicationName	[0..1]	string
e	appData1	[0..1]	string
e	appData2	[0..1]	string
e	appData3	[0..1]	string
e	appData4	[0..1]	string
e	userData1	[0..1]	string
e	userData2	[0..1]	string
e	userData3	[0..1]	string
e	appTextMini1	[0..1]	string
e	appTextMini2	[0..1]	string
e	appTextShort1	[0..1]	string
e	appTextShort2	[0..1]	string
e	appTextShort3	[0..1]	string
e	appTextShort4	[0..1]	string
e	appTextLarge1	[0..1]	string
e	appTextLarge2	[0..1]	string
e	serviceTypeUETR	[0..1]	string
e	applicationQueue	[0..1]	int
e	applicationStatusValue	[0..1]	int
e	mpsStatusValue	[0..1]	int
e	osnSession	[0..1]	int
e	osnSequence	[0..1]	int
e	msgvalStatus	[0..1]	int
e	embcheckStatus	[0..1]	int
e	reportingStatusValue	[0..1]	int
e	originationModeCV	[0..1]	int
e	boxMessageType	[0..1]	int
e	boxChannelType	[0..1]	int
e	applicationDefinedType	[0..1]	int
e	wireLength	[0..1]	int
e	isnSession	[0..1]	int
e	isnSequence	[0..1]	int
e	deliveryTryCount	[0..1]	int
e	deliveryStatus	[0..1]	int
e	multicastMessageFlag	[0..1]	int
e	gpaMessageFlag	[0..1]	int
e	securityFailureCode	[0..1]	int
e	appNum1	[0..1]	int
e	appNum2	[0..1]	int
e	amount	[0..1]	decimal
e	amountNormalized	[0..1]	decimal
e	appDecimal1	[0..1]	decimal
e	appDecimal2	[0..1]	decimal
e	deliveryStartTime	[0..1]	dateTime
e	deliveryStartTimeOffset	[0..1]	int
e	deliveryEndTime	[0..1]	dateTime
e	deliveryEndTimeOffset	[0..1]	int
e	receptionTime	[0..1]	dateTime
e	receptionTimeOffset	[0..1]	int
e	appTimestamp1	[0..1]	dateTime
e	appTimestamp1Offset	[0..1]	int
e	appTimestamp2	[0..1]	dateTime
e	appTimestamp2Offset	[0..1]	int
e	valueDate	[0..1]	date
e	appDate1	[0..1]	date
e	appDate2	[0..1]	date

2.6 Appendix

2.6.1 Box Archive Retrieval wdsI

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions xmlns:ar="http://ws.box.intercope.com/BoxArchiveRetrieval/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="BoxArchiveRetrieval"
targetNamespace="http://ws.box.intercope.com/BoxArchiveRetrieval/">
<wsdl:types>
<xsd:schema targetNamespace="http://ws.box.intercope.com/BoxArchiveRetrieval/"
xmlns:jxb="http://java.sun.com/xml/ns/jaxb" jxb:version="2.0">
<xsd:element name="retrieveRequest">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="sessionId" type="ar:SessionIdType" />
<xsd:element name="datasource"
type="xsd:string"><xsd:annotation><xsd:documentation>shortname of journal data
source (V_XXXARCH_<cltprefix)</xsd:documentation></xsd:annotation></xsd:element>
<xsd:element name="retrieveData" type="xsd:boolean" />
<xsd:element name="attributeTypeSet" type="ar:ArchiveAttributeSetType"/>
<xsd:element name="entryId" type="xsd:integer" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="searchResponse">
<xsd:complexType>
<xsd:sequence>
<xsd:choice>
<xsd:element name="listResult" type="ar:ListResultType" />
</xsd:choice>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="searchRequest" type="ar:SearchRequestType" />
<xsd:simpleType name="Max63TextType">
<xsd:restriction base="xsd:string">
<xsd:maxLength value="63" />
</xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="SearchRequestType">
<xsd:sequence>
<xsd:element name="sessionId" type="ar:SessionIdType" />
<xsd:element name="datasource"
type="xsd:string"><xsd:annotation><xsd:documentation>shortname of journal data
source (V_XXXARCH_<cltprefix)</xsd:documentation></xsd:annotation></xsd:element>
<xsd:element name="retrieveData" type="xsd:boolean" />
<xsd:element name="attributeTypeSet" type="ar:ArchiveAttributeSetType"/>
<xsd:element name="limitResultSetSize" type="xsd:integer" maxOccurs="1"
minOccurs="0" default="100">
<xsd:annotation>
<xsd:documentation>if omitted a default of 100 is
used</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="dateRange" type="ar:DateRangeType" maxOccurs="1"
minOccurs="1" />
<xsd:element name="searchCondition" type="ar:BlockPredicateType"
minOccurs="1" maxOccurs="1" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="AbstractPredicateType" abstract="true">
</xsd:complexType>
<xsd:complexType name="AbstractValuePredicateType" abstract="true">
<xsd:attribute name="key" type="ar:PredicateKeyList" use="required" />
<xsd:attribute name="operator" type="ar:OperatorType" use="required" />
</xsd:complexType>

```

```
<xsd:simpleType name="ArchiveAttributeSetType">
<xsd:restriction base="xsd:string">
  <xsd:enumeration value="FIN" />
<!--
<xsd:enumeration value="MX"/>
<xsd:enumeration value="RMA"/>
<xsd:enumeration value="FACT"/>
<xsd:enumeration value="EBICS_IBFT"/>
<xsd:enumeration value="SIA_FTS"/>
<xsd:enumeration value="SIA_FLS"/>
<xsd:enumeration value="RNI"/>
<xsd:enumeration value="T2SF"/>
<xsd:enumeration value="T2SM"/>
-->
<!--
<xsd:enumeration value="APPLDEF"/>
-->
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="OperatorType">
<xsd:restriction base="xsd:string">
  <xsd:enumeration value="" />
  <!--
  <xsd:enumeration value="eq" />
  -->
  <xsd:enumeration value="&lt;" />
  <!--
  <xsd:enumeration value="lt" />
  -->
  <xsd:enumeration value="&gt;" />
  <!--
  <xsd:enumeration value="gt" />
  -->
  <xsd:enumeration value="&lt;=" />
  <!--
  <xsd:enumeration value="le" />
  -->
  <xsd:enumeration value="&gt;=" />
  <!--
  <xsd:enumeration value="ge" />
  -->
  <xsd:enumeration value="&lt;&gt;" />
  <!--
  <xsd:enumeration value="ne" />
  -->
  <xsd:enumeration value="IS NULL" />
  <xsd:enumeration value="LIKE" />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="GenericPredicateKeyList">
<xsd:restriction base="xsd:string">
  <xsd:enumeration value="ENTRYID" />
  <xsd:enumeration value="CREATIONDATE" />
  <xsd:enumeration value="QENTRYTIME" />
  <xsd:enumeration value="QENTRYTIMEOFF" />
  <xsd:enumeration value="MPSID" />
  <xsd:enumeration value="DISPREF" />
  <xsd:enumeration value="ICCOMMENT" />
  <xsd:enumeration value="TEXT_1" />
  <xsd:enumeration value="TEXT_2" />
  <xsd:enumeration value="TEXT_3" />
  <xsd:enumeration value="TEXT_4" />
  <xsd:enumeration value="TEXT_5" />
  <xsd:enumeration value="TEXT_6" />
  <xsd:enumeration value="TEXT_7" />
  <xsd:enumeration value="TEXT_8" />
  <xsd:enumeration value="TEXT_9" />
  <xsd:enumeration value="TEXT_10" />
  <xsd:enumeration value="TEXT_11" />
</xsd:restriction>
</xsd:simpleType>
```

```
<xsd:enumeration value="TEXT_12" />
<xsd:enumeration value="TEXT_13" />
<xsd:enumeration value="TEXT_14" />
<xsd:enumeration value="TEXT_15" />
<xsd:enumeration value="TEXT_16" />
<xsd:enumeration value="TEXT_17" />
<xsd:enumeration value="TEXT_18" />
<xsd:enumeration value="TEXT_19" />
<xsd:enumeration value="TEXT_20" />
<xsd:enumeration value="TEXT_21" />
<xsd:enumeration value="TEXT_22" />
<xsd:enumeration value="TEXT_23" />
<xsd:enumeration value="TEXT_24" />
<xsd:enumeration value="TEXT_25" />
<xsd:enumeration value="TEXT_26" />
<xsd:enumeration value="TEXT_27" />
<xsd:enumeration value="TEXT_28" />
<xsd:enumeration value="TEXT_29" />
<xsd:enumeration value="TEXT_30" />
<xsd:enumeration value="TEXT_31" />
<xsd:enumeration value="TEXT_32" />
<xsd:enumeration value="TEXT_33" />
<xsd:enumeration value="TEXT_34" />
<xsd:enumeration value="TEXT_35" />
<xsd:enumeration value="TEXT_36" />
<xsd:enumeration value="TEXT_37" />
<xsd:enumeration value="TEXT_38" />
<xsd:enumeration value="TEXT_39" />
<xsd:enumeration value="TEXT_40" />
<xsd:enumeration value="TEXT_41" />
<xsd:enumeration value="TEXT_42" />
<xsd:enumeration value="TEXT_43" />
<xsd:enumeration value="TEXT_44" />
<xsd:enumeration value="TEXT_45" />
<xsd:enumeration value="TEXT_46" />
<xsd:enumeration value="TEXT_47" />
<xsd:enumeration value="TEXT_48" />
<xsd:enumeration value="TEXT_49" />
<xsd:enumeration value="TEXT_50" />
<xsd:enumeration value="TEXT_51" />
<xsd:enumeration value="TEXT_52" />
<xsd:enumeration value="TEXT_53" />
<xsd:enumeration value="TEXT_54" />
<xsd:enumeration value="TEXT_55" />
<xsd:enumeration value="TEXT_56" />
<xsd:enumeration value="TEXT_57" />
<xsd:enumeration value="TEXT_58" />
<xsd:enumeration value="TEXT_59" />
<xsd:enumeration value="TEXT_60" />
<xsd:enumeration value="TEXT_61" />
<xsd:enumeration value="TEXT_62" />
<xsd:enumeration value="TEXT_63" />
<xsd:enumeration value="TEXT_64" />
<xsd:enumeration value="TEXT_65" />
<xsd:enumeration value="NUM_1" />
<xsd:enumeration value="NUM_2" />
<xsd:enumeration value="NUM_3" />
<xsd:enumeration value="NUM_4" />
<xsd:enumeration value="NUM_5" />
<xsd:enumeration value="NUM_6" />
<xsd:enumeration value="NUM_7" />
<xsd:enumeration value="NUM_8" />
<xsd:enumeration value="NUM_9" />
<xsd:enumeration value="NUM_10" />
<xsd:enumeration value="NUM_11" />
<xsd:enumeration value="NUM_12" />
<xsd:enumeration value="NUM_13" />
<xsd:enumeration value="NUM_14" />
<xsd:enumeration value="NUM_15" />
```

```
<xsd:enumeration value="NUM_16" />
<xsd:enumeration value="NUM_17" />
<xsd:enumeration value="NUM_18" />
<xsd:enumeration value="NUM_19" />
<xsd:enumeration value="NUM_20" />
<xsd:enumeration value="NUM_21" />
<xsd:enumeration value="NUM_22" />
<xsd:enumeration value="NUM_23" />
<xsd:enumeration value="NUM_24" />
<xsd:enumeration value="NUM_25" />
<xsd:enumeration value="NUM_26" />
<xsd:enumeration value="NUM_27" />
<xsd:enumeration value="NUM_28" />
<xsd:enumeration value="NUM_29" />
<xsd:enumeration value="NUM_30" />
<xsd:enumeration value="NUM_31" />
<xsd:enumeration value="NUM_32" />
<xsd:enumeration value="NUM_33" />
<xsd:enumeration value="NUM_34" />
<xsd:enumeration value="NUM_35" />
<xsd:enumeration value="NUM_36" />
<xsd:enumeration value="NUM_37" />
<xsd:enumeration value="NUM_38" />
<xsd:enumeration value="NUM_39" />
<xsd:enumeration value="NUM_40" />
<xsd:enumeration value="NUM_41" />
<xsd:enumeration value="NUM_42" />
<xsd:enumeration value="NUM_43" />
<xsd:enumeration value="NUM_44" />
<xsd:enumeration value="NUM_45" />
<xsd:enumeration value="NUM_46" />
<xsd:enumeration value="NUM_47" />
<xsd:enumeration value="NUM_48" />
<xsd:enumeration value="NUM_49" />
<xsd:enumeration value="NUM_50" />
<xsd:enumeration value="NUM64_1" />
<xsd:enumeration value="DECIMAL_1" />
<xsd:enumeration value="DECIMAL_2" />
<xsd:enumeration value="DECIMAL_3" />
<xsd:enumeration value="DECIMAL_4" />
<xsd:enumeration value="DECIMAL_5" />
<xsd:enumeration value="TS_1" />
<xsd:enumeration value="TS_1_OFF" />
<xsd:enumeration value="TS_2" />
<xsd:enumeration value="TS_2_OFF" />
<xsd:enumeration value="TS_3" />
<xsd:enumeration value="TS_3_OFF" />
<xsd:enumeration value="TS_4" />
<xsd:enumeration value="TS_4_OFF" />
<xsd:enumeration value="TS_5" />
<xsd:enumeration value="TS_5_OFF" />
<xsd:enumeration value="TS_6" />
<xsd:enumeration value="TS_6_OFF" />
<xsd:enumeration value="TS_7" />
<xsd:enumeration value="TS_7_OFF" />
<xsd:enumeration value="TIME_1" />
<xsd:enumeration value="TIME_2" />
<xsd:enumeration value="DATE_1" />
<xsd:enumeration value="DATE_2" />
<xsd:enumeration value="DATE_3" />
<xsd:enumeration value="DATE_4" />
<xsd:enumeration value="DATE_5" />
<xsd:enumeration value="LASTUPDATE" />
<xsd:enumeration value="LASTUPDATEOFF" />
<xsd:enumeration value="CDATA" />
<xsd:enumeration value="MSGTYPE" />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="FINPredicateKeyList">
```

```
<xsd:restriction base="xsd:string">
  <xsd:enumeration value="mt" />
  <xsd:enumeration value="mur" />
  <xsd:enumeration value="trn" />
  <xsd:enumeration value="currency" />
  <xsd:enumeration value="finCopyServiceIdentifier" />
  <xsd:enumeration value="bankingPriority" />
  <xsd:enumeration value="lt" />
  <xsd:enumeration value="correspondent" />
  <xsd:enumeration value="externalStatusDescription" />
  <xsd:enumeration value="originatorMessageId" />
  <xsd:enumeration value="receiverMessageId" />
  <xsd:enumeration value="messageReferenceI" />
  <xsd:enumeration value="displayReference" />
  <xsd:enumeration value="destinationDisplayReference" />
  <xsd:enumeration value="dliDisplayReference" />
  <xsd:enumeration value="sddOriginatorMessageId" />
  <xsd:enumeration value="sddReceiverMessageId" />
  <xsd:enumeration value="messageReferenceS" />
  <xsd:enumeration value="recipientOriginalDisplayAddress" />
  <xsd:enumeration value="recipientDisplayAddress" />
  <xsd:enumeration value="cbtAckNakCode" />
  <xsd:enumeration value="relatedReferenceNumber" />
  <xsd:enumeration value="F21" />
  <xsd:enumeration value="securityCodeISIN" />
  <xsd:enumeration value="accountNumber" />
  <xsd:enumeration value="F25" />
  <xsd:enumeration value="securitiesDepositAccount" />
  <xsd:enumeration value="F97" />
  <xsd:enumeration value="orderingBank" />
  <xsd:enumeration value="beneficiaryBank" />
  <xsd:enumeration value="pdeFlag" />
  <xsd:enumeration value="bankOpCode" />
  <xsd:enumeration value="pdmFlag" />
  <xsd:enumeration value="lastActingUser" />
  <xsd:enumeration value="lastAckDelNotType" />
  <xsd:enumeration value="validationFlagFromUserHeader" />
  <xsd:enumeration value="rxChannelDisplayName" />
  <xsd:enumeration value="txChannelDisplayName" />
  <xsd:enumeration value="msgvalErrorCode" />
  <xsd:enumeration value="specProtMessageOD" />
  <xsd:enumeration value="specProtCodeOD" />
  <xsd:enumeration value="specDevCodeOD" />
  <xsd:enumeration value="specDevCode" />
  <xsd:enumeration value="broadcastText" />
  <xsd:enumeration value="sourceApplicationName" />
  <xsd:enumeration value="targetApplicationName" />
  <xsd:enumeration value="appData1" />
  <xsd:enumeration value="appData2" />
  <xsd:enumeration value="appData3" />
  <xsd:enumeration value="appData4" />
  <xsd:enumeration value="userData1" />
  <xsd:enumeration value="userData2" />
  <xsd:enumeration value="userData3" />
  <xsd:enumeration value="appTextMini1" />
  <xsd:enumeration value="appTextMini2" />
  <xsd:enumeration value="appTextShort1" />
  <xsd:enumeration value="appTextShort2" />
  <xsd:enumeration value="appTextShort3" />
  <xsd:enumeration value="appTextShort4" />
  <xsd:enumeration value="appTextLarge1" />
  <xsd:enumeration value="appTextLarge2" />
  <xsd:enumeration value="serviceTypeUETR" />
  <!-- numerical -->
  <xsd:enumeration value="applicationQueue" />
  <xsd:enumeration value="applicationStatusValue" />
  <xsd:enumeration value="mpsStatusValue" />
  <xsd:enumeration value="osnSession" />
  <xsd:enumeration value="osnSequence" />

```

```
<xsd:enumeration value="msgvalStatus" />
<xsd:enumeration value="embcheckStatus" />
<xsd:enumeration value="reportingStatusValue" />
<xsd:enumeration value="originationModeCV" />
<xsd:enumeration value="boxMessageType" />
<xsd:enumeration value="boxChannelType" />
<xsd:enumeration value="applicationDefinedType" />
<xsd:enumeration value="IN_OUT" />
<xsd:enumeration value="wireLength" />
<!--
<xsd:enumeration value="addressType" />
<xsd:enumeration value="addressExecutionNumber" />
-->
<xsd:enumeration value="isnSession" />
<xsd:enumeration value="isnSequence" />
<!--
<xsd:enumeration value="DELIVSTAT"/>
<xsd:enumeration value="SYS_RESCODE"/>
-->
<xsd:enumeration value="SDD_RESCODE" />
<xsd:enumeration value="SDD_ERRCODE" />
<xsd:enumeration value="deliveryTryCount" />
<xsd:enumeration value="deliveryStatus" />
<xsd:enumeration value="multicastMessageFlag" />
<xsd:enumeration value="gpaMessageFlag" />
<xsd:enumeration value="securityFailureCode" />
<xsd:enumeration value="appNum1" />
<xsd:enumeration value="appNum2" />
<xsd:enumeration value="amount" />
<xsd:enumeration value="amountNormalized" />
<xsd:enumeration value="appDecimal1" />
<xsd:enumeration value="appDecimal2" />
<!-- timestamp -->
<xsd:enumeration value="deliveryStartTime" />
<xsd:enumeration value="deliveryEndTime" />
<xsd:enumeration value="receptionTime" />
<xsd:enumeration value="appTimestamp1" />
<xsd:enumeration value="appTimestamp2" />
<!-- date -->
<xsd:enumeration value="valueDate" />
<xsd:enumeration value="appDate1" />
<xsd:enumeration value="appDate2" />
<xsd:enumeration value="wireText" />
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="PredicateKeyList">
<xsd:union memberTypes="ar:GenericPredicateKeyList ar:FINPredicateKeyList">
</xsd:union>
</xsd:simpleType>
<xsd:complexType name="StringPredicateType">
<xsd:complexContent>
<xsd:extension base="ar:AbstractValuePredicateType">
<xsd:choice minOccurs="0" maxOccurs="1">
<xsd:element name="value" type="xsd:string"/></xsd:choice>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="IntPredicateType">
<xsd:complexContent>
<xsd:extension base="ar:AbstractValuePredicateType">
<xsd:choice minOccurs="0" maxOccurs="1">
<xsd:element name="value" type="xsd:int"/></xsd:choice>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="BigIntPredicateType">
<xsd:complexContent>
```

```
<xsd:extension base="ar:AbstractValuePredicateType">
  <xsd:choice minOccurs="0" maxOccurs="1">
    <xsd:element name="value" type="xsd:integer"></xsd:element>
  </xsd:choice>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="BigDecimalPredicateType">
<xsd:complexContent>
  <xsd:extension base="ar:AbstractValuePredicateType">
    <xsd:choice minOccurs="0" maxOccurs="1">
      <xsd:element name="value" type="xsd:decimal"></xsd:element>
    </xsd:choice>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="TimestampPredicateType">
<xsd:complexContent>
  <xsd:extension base="ar:AbstractValuePredicateType">
    <xsd:choice minOccurs="0" maxOccurs="1">
      <xsd:element name="value" type="xsd:dateTime"></xsd:element>
    </xsd:choice>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="TimePredicateType">
<xsd:complexContent>
  <xsd:extension base="ar:AbstractValuePredicateType">
    <xsd:choice minOccurs="0" maxOccurs="1">
      <xsd:element name="value" type="xsd:time"></xsd:element>
    </xsd:choice>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="DatePredicateType">
<xsd:complexContent>
  <xsd:extension base="ar:AbstractValuePredicateType">
    <xsd:choice minOccurs="0" maxOccurs="1">
      <xsd:element name="value" type="xsd:date"></xsd:element>
    </xsd:choice>
  </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="AbstractConditionType" abstract="true">
</xsd:complexType>
<xsd:complexType name="AbstractPredicateElementType" abstract="true">
</xsd:complexType>
<xsd:complexType name="AndConditionType">
<xsd:complexContent>
  <xsd:extension base="ar:AbstractConditionType"></xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="OrConditionType">
<xsd:complexContent>
  <xsd:extension base="ar:AbstractConditionType"></xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:element name="block" type="ar:BlockPredicateType" />
<xsd:element name="stringPredicate" type="ar:StringPredicateType" />
<xsd:element name="intPredicate" type="ar:IntPredicateType" />
<xsd:element name="bigIntPredicate" type="ar:BigIntPredicateType" />
<xsd:element name="bigDecimalPredicate" type="ar:BigDecimalPredicateType" />
<xsd:element name="timestampPredicate" type="ar:TimestampPredicateType" />
<xsd:element name="timePredicate" type="ar:TimePredicateType" />
<xsd:element name="datePredicate" type="ar:DatePredicateType" />
<xsd:complexType name="BlockPredicateType">
<xsd:complexContent>
  <xsd:extension base="ar:AbstractPredicateElementType">
    <xsd:sequence>
```

```
<xsd:annotation>
  <xsd:appinfo>
    <jxb:property name="elements"></jxb:property>
  </xsd:appinfo>
</xsd:annotation>
<xsd:choice minOccurs="1" maxOccurs="1">
  <xsd:element ref="ar:stringPredicate" />
  <xsd:element ref="ar:intPredicate" />
  <xsd:element ref="ar:bigIntPredicate" />
  <xsd:element ref="ar:bigDecimalPredicate" />
  <xsd:element ref="ar:timestampPredicate" />
  <xsd:element ref="ar:timePredicate" />
  <xsd:element ref="ar:datePredicate" />
  <xsd:element ref="ar:block" />
</xsd:choice>
<xsd:sequence minOccurs="0" maxOccurs="unbounded">
  <xsd:choice minOccurs="1" maxOccurs="1">
    <xsd:element name="and" type="ar:AndConditionType" />
    <xsd:element name="or" type="ar:OrConditionType" />
  </xsd:choice>
  <xsd:choice minOccurs="1" maxOccurs="1">
    <xsd:element ref="ar:stringPredicate" />
    <xsd:element ref="ar:intPredicate" />
    <xsd:element ref="ar:bigIntPredicate" />
    <xsd:element ref="ar:bigDecimalPredicate" />
    <xsd:element ref="ar:timestampPredicate" />
    <xsd:element ref="ar:timePredicate" />
    <xsd:element ref="ar:datePredicate" />
    <xsd:element ref="ar:block" />
  </xsd:choice>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="DateRangeType">
<xsd:sequence>
  <xsd:element name="from" type="xsd:date" maxOccurs="1" minOccurs="1" />
  <xsd:element name="to" type="xsd:date" maxOccurs="1" minOccurs="0" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="AmountRangeType">
<xsd:sequence>
  <xsd:element name="from" type="xsd:decimal" maxOccurs="1" minOccurs="0" />
  <xsd:element name="to" type="xsd:decimal" maxOccurs="1" minOccurs="0" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ListResultType">
<xsd:sequence>
  <xsd:element name="archiveEntry" type="ar:AbstractArchiveEntryType"
maxOccurs="unbounded" minOccurs="1" />
</xsd:sequence>
<xsd:attribute name="count" type="xsd:integer"></xsd:attribute>
</xsd:complexType>
<xsd:complexType name="ArchiveDataType">
<xsd:sequence>
  <xsd:element name="content" type="xsd:anyType"/>
</xsd:sequence>
<xsd:attribute name="encoding" type="ar:DataEncodingType"></xsd:attribute>
</xsd:complexType>
<xsd:simpleType name="DataEncodingType">
<xsd:restriction base="xsd:string">
  <xsd:enumeration value="plain-text"/>
  <xsd:enumeration value="Base64"/>
  <xsd:enumeration value="hex-ascii"/>
  <xsd:enumeration value="cdata"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="AbstractArchiveEntryType" abstract="true">
```

```

<xsd:sequence>
  <!-- fixed (unmapped) attributes -->
  <xsd:element name="entryId" type="xsd:integer" />
  <xsd:element name="mpsId" type="xsd:integer" />
  <xsd:element name="msgType" type="xsd:int" minOccurs="0"/>
  <xsd:element name="queueEntryTime" type="xsd:dateTime" minOccurs="0"/>
  <xsd:element name="queueEntryTimeOffset" type="xsd:int" minOccurs="0"/>
  <xsd:element name="boxDisplayReference" type="xsd:string" minOccurs="0"/>
  <xsd:element name="comment" type="xsd:string" minOccurs="0"/>
  <xsd:element name="creationDate" type="xsd:date" />
  <!-- generic attributes -->
  <!-- ... -->
  <!-- optional data element -->
  <xsd:element name="data" type="ar:ArchiveDataType" minOccurs="0"/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="FINArchiveEntryType">
  <xsd:complexContent>
    <xsd:extension base="ar:AbstractArchiveEntryType">
      <xsd:sequence>
        <xsd:element name="mt" type="xsd:string" minOccurs="0"/>
        <xsd:element name="mur" type="xsd:string" minOccurs="0"/>
        <xsd:element name="trn" type="xsd:string" minOccurs="0"/>
        <xsd:element name="currency" type="xsd:string" minOccurs="0"/>
        <xsd:element name="finCopyServiceIdentifier" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="bankingPriority" type="xsd:string" minOccurs="0"/>
        <xsd:element name="lt" type="xsd:string" minOccurs="0"/>
        <xsd:element name="correspondent" type="xsd:string" minOccurs="0"/>
        <xsd:element name="externalStatusDescription" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="originatorMessageId" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="receiverMessageId" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="messageReferenceI" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="displayReference" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="destinationDisplayReference" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="dliDisplayReference" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="sddOriginatorMessageId" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="sddReceiverMessageId" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="messageReferences" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="recipientOriginalDisplayAddress" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="recipientDisplayAddress" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="cbtAckNakCode" type="xsd:string" minOccurs="0"/>
        <xsd:element name="relatedReferenceNumber" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="securityCodeISIN" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="accountNumber" type="xsd:string" minOccurs="0"/>
        <xsd:element name="securitiesDepositAccount" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="orderingBank" type="xsd:string" minOccurs="0"/>
        <xsd:element name="beneficiaryBank" type="xsd:string" minOccurs="0"/>
        <xsd:element name="pdeFlag" type="xsd:string" minOccurs="0"/>
        <xsd:element name="bankOpCode" type="xsd:string" minOccurs="0"/>
        <xsd:element name="pdmFlag" type="xsd:string" minOccurs="0"/>
        <xsd:element name="lastActingUser" type="xsd:string" minOccurs="0"/>
        <xsd:element name="lastAckDelNotType" type="xsd:string"
minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```

        <xsd:element name="validationFlagFromUserHeader" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="rxChannelDisplayName" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="txChannelDisplayName" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="msgvalErrorCode" type="xsd:string" minOccurs="0"/>
        <!--
        <xsd:element name="specProtMessageOD" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="specProtCodeOD" type="xsd:string" minOccurs="0"/>
        <xsd:element name="specDevCodeOD" type="xsd:string" minOccurs="0"/>
        <xsd:element name="specDevCode" type="xsd:string" minOccurs="0"/>
        -->
        <xsd:element name="broadcastText" type="xsd:string" minOccurs="0"/>
        <xsd:element name="sourceApplicationName" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="targetApplicationName" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="appData1" type="xsd:string" minOccurs="0"/>
        <xsd:element name="appData2" type="xsd:string" minOccurs="0"/>
        <xsd:element name="appData3" type="xsd:string" minOccurs="0"/>
        <xsd:element name="appData4" type="xsd:string" minOccurs="0"/>
        <xsd:element name="userData1" type="xsd:string" minOccurs="0"/>
        <xsd:element name="userData2" type="xsd:string" minOccurs="0"/>
        <xsd:element name="userData3" type="xsd:string" minOccurs="0"/>
        <xsd:element name="appTextMini1" type="xsd:string" minOccurs="0"/>
        <xsd:element name="appTextMini2" type="xsd:string" minOccurs="0"/>
        <xsd:element name="appTextShort1" type="xsd:string" minOccurs="0"/>
        <xsd:element name="appTextShort2" type="xsd:string" minOccurs="0"/>
        <xsd:element name="appTextShort3" type="xsd:string" minOccurs="0"/>
        <xsd:element name="appTextShort4" type="xsd:string" minOccurs="0"/>
        <xsd:element name="appTextLarge1" type="xsd:string" minOccurs="0"/>
        <xsd:element name="appTextLarge2" type="xsd:string" minOccurs="0"/>
        <xsd:element name="serviceTypeUETR" type="xsd:string" minOccurs="0"/>
        <!-- numerical -->
        <xsd:element name="applicationQueue" type="xsd:int" minOccurs="0"/>
        <xsd:element name="applicationStatusValue" type="xsd:int"
minOccurs="0"/>
        <xsd:element name="mpsStatusValue" type="xsd:int" minOccurs="0"/>
        <xsd:element name="osnSession" type="xsd:int" minOccurs="0"/>
        <xsd:element name="osnSequence" type="xsd:int" minOccurs="0"/>
        <xsd:element name="msgvalStatus" type="xsd:int" minOccurs="0"/>
        <xsd:element name="embcheckStatus" type="xsd:int" minOccurs="0"/>
        <xsd:element name="reportingStatusValue" type="xsd:int"
minOccurs="0"/>
        <xsd:element name="originationModeCV" type="xsd:int" minOccurs="0"/>
        <xsd:element name="boxMessageType" type="xsd:int" minOccurs="0"/>
        <xsd:element name="boxChannelType" type="xsd:int" minOccurs="0"/>
        <xsd:element name="applicationDefinedType" type="xsd:int"
minOccurs="0"/>
        <xsd:element name="wireLength" type="xsd:int" minOccurs="0"/>
        <!--
        <xsd:element name="addressType" type="xsd:int" minOccurs="0"/>
        <xsd:element name="addressExecutionNumber" type="xsd:int"
minOccurs="0"/>
        -->
        <xsd:element name="isnSession" type="xsd:int" minOccurs="0"/>
        <xsd:element name="isnSequence" type="xsd:int" minOccurs="0"/>
        <xsd:element name="deliveryTryCount" type="xsd:int" minOccurs="0"/>
        <xsd:element name="deliveryStatus" type="xsd:int" minOccurs="0"/>
        <xsd:element name="multicastMessageFlag" type="xsd:int"
minOccurs="0"/>
        <xsd:element name="gpaMessageFlag" type="xsd:int" minOccurs="0"/>
        <xsd:element name="securityFailureCode" type="xsd:int"
minOccurs="0"/>
        <xsd:element name="appNum1" type="xsd:int" minOccurs="0"/>
        <xsd:element name="appNum2" type="xsd:int" minOccurs="0"/>
        <xsd:element name="amount" type="xsd:decimal" minOccurs="0"/>

```

```
        <xsd:element name="amountNormalized" type="xsd:decimal"
minOccurs="0"/>
        <xsd:element name="appDecimal1" type="xsd:decimal" minOccurs="0"/>
        <xsd:element name="appDecimal2" type="xsd:decimal" minOccurs="0"/>
        <!-- timestamp -->
        <xsd:element name="deliveryStartTime" type="xsd:dateTime"
minOccurs="0"/>
        <xsd:element name="deliveryStartTimeOffset" type="xsd:int"
minOccurs="0"/>
        <xsd:element name="deliveryEndTime" type="xsd:dateTime"
minOccurs="0"/>
        <xsd:element name="deliveryEndTimeOffset" type="xsd:int"
minOccurs="0"/>
        <xsd:element name="receptionTime" type="xsd:dateTime" minOccurs="0"/>
        <xsd:element name="receptionTimeOffset" type="xsd:int"
minOccurs="0"/>
        <xsd:element name="appTimestamp1" type="xsd:dateTime" minOccurs="0"/>
        <xsd:element name="appTimestamp1Offset" type="xsd:int"
minOccurs="0"/>
        <xsd:element name="appTimestamp2" type="xsd:dateTime" minOccurs="0"/>
        <xsd:element name="appTimestamp2Offset" type="xsd:int"
minOccurs="0"/>
        <!-- date -->
        <xsd:element name="valueDate" type="xsd:date" minOccurs="0"/>
        <!-- 2, 3 empty -->
        <xsd:element name="appDate1" type="xsd:date" minOccurs="0"/>
        <xsd:element name="appDate2" type="xsd:date" minOccurs="0"/>
    </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="GenericArchiveEntryType">
<xsd:complexContent>
    <xsd:extension base="ar:AbstractArchiveEntryType">
        <xsd:sequence>
            <xsd:element name="text1" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text2" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text3" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text4" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text5" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text6" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text7" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text8" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text9" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text10" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text11" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text12" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text13" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text14" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text15" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text16" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text17" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text18" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text19" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text20" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text21" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text22" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text23" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text24" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text25" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text26" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text27" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text28" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text29" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text30" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text31" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text32" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text33" type="xsd:string" minOccurs="0"/>
            <xsd:element name="text34" type="xsd:string" minOccurs="0"/>
        </xsd:sequence>
    </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
</xsd:complexType>
```



```

<xsd:element name="num39" type="xsd:integer" minOccurs="0"/>
<xsd:element name="num40" type="xsd:integer" minOccurs="0"/>
<xsd:element name="num41" type="xsd:integer" minOccurs="0"/>
<xsd:element name="num42" type="xsd:integer" minOccurs="0"/>
<xsd:element name="num43" type="xsd:integer" minOccurs="0"/>
<xsd:element name="num44" type="xsd:integer" minOccurs="0"/>
<xsd:element name="num45" type="xsd:integer" minOccurs="0"/>
<xsd:element name="num46" type="xsd:integer" minOccurs="0"/>
<xsd:element name="num47" type="xsd:integer" minOccurs="0"/>
<xsd:element name="num48" type="xsd:integer" minOccurs="0"/>
<xsd:element name="num49" type="xsd:integer" minOccurs="0"/>
<xsd:element name="num50" type="xsd:integer" minOccurs="0"/>
<xsd:element name="num64_1" type="xsd:integer" minOccurs="0"/>
<xsd:element name="decimal1" type="xsd:decimal" minOccurs="0"/>
<xsd:element name="decimal2" type="xsd:decimal" minOccurs="0"/>
<xsd:element name="decimal3" type="xsd:decimal" minOccurs="0"/>
<xsd:element name="decimal4" type="xsd:decimal" minOccurs="0"/>
<xsd:element name="decimal5" type="xsd:decimal" minOccurs="0"/>
<xsd:element name="timestamp1" type="xsd:dateTime" minOccurs="0"/>
<xsd:element name="timestamp1offset" type="xsd:integer"
minOccurs="0"/>
<xsd:element name="timestamp2" type="xsd:dateTime" minOccurs="0"/>
<xsd:element name="timestamp2offset" type="xsd:integer"
minOccurs="0"/>
<xsd:element name="timestamp3" type="xsd:dateTime" minOccurs="0"/>
<xsd:element name="timestamp3offset" type="xsd:integer"
minOccurs="0"/>
<xsd:element name="timestamp4" type="xsd:dateTime" minOccurs="0"/>
<xsd:element name="timestamp4offset" type="xsd:integer"
minOccurs="0"/>
<xsd:element name="timestamp5" type="xsd:dateTime" minOccurs="0"/>
<xsd:element name="timestamp5offset" type="xsd:integer"
minOccurs="0"/>
<xsd:element name="timestamp6" type="xsd:dateTime" minOccurs="0"/>
<xsd:element name="timestamp6offset" type="xsd:integer"
minOccurs="0"/>
<xsd:element name="timestamp7" type="xsd:dateTime" minOccurs="0"/>
<xsd:element name="timestamp7offset" type="xsd:integer"
minOccurs="0"/>
<xsd:element name="time1" type="xsd:time" minOccurs="0"/>
<xsd:element name="time2" type="xsd:time" minOccurs="0"/>
<xsd:element name="date1" type="xsd:date" minOccurs="0"/>
<xsd:element name="date2" type="xsd:date" minOccurs="0"/>
<xsd:element name="date3" type="xsd:date" minOccurs="0"/>
<xsd:element name="date4" type="xsd:date" minOccurs="0"/>
<xsd:element name="date5" type="xsd:date" minOccurs="0"/>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>

<xsd:element name="searchFault">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="searchFault" type="xsd:string" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="TooManyResultsFaultType" type="xsd:string"/>
<xsd:element name="retrieveResponse">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="archiveEntry" type="ar:AbstractArchiveEntryType"
maxOccurs="1" minOccurs="0" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="retrieveFault">
<xsd:complexType>

```

```
<xsd:sequence>
  <xsd:element name="retrieveFault" type="xsd:string" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="EntryNotFoundRetrieveFaultType" type="xsd:string" />
<xsd:element name="retrieveFault1">
<xsd:complexType>
  <xsd:sequence>
    <xsd:element name="retrieveFault1" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="ArchiveRetrievalFault" type="xsd:string" />
<xsd:element name="searchFault1">
<xsd:complexType>
  <xsd:sequence>
    <xsd:element name="searchFault1" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:complexType name="SecretType">
<xsd:choice>
  <xsd:element name="plain" type="xsd:string">
    <xsd:annotation>
      <xsd:documentation>plain text password</xsd:documentation>
    </xsd:annotation>
  </xsd:element>
</xsd:choice>
</xsd:complexType>
<xsd:simpleType name="SessionIdType">
<xsd:restriction base="xsd:string" />
</xsd:simpleType>
<xsd:complexType name="LoginType">
<xsd:sequence>
  <xsd:element maxOccurs="1" minOccurs="1" name="clientPrefix"
type="xsd:string">
  <xsd:annotation>
    <xsd:documentation />
  </xsd:annotation>
</xsd:element>
  <xsd:element maxOccurs="1" minOccurs="1" name="userName" type="xsd:string">
  <xsd:annotation>
    <xsd:documentation />
  </xsd:annotation>
</xsd:element>
  <xsd:element maxOccurs="1" minOccurs="1" name="secret" type="ar:SecretType">
  <xsd:annotation>
    <xsd:documentation />
  </xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:element name="switchContextPath">
<xsd:complexType>
  <xsd:sequence>
    <xsd:element name="sessionId" type="ar:SessionIdType">
    </xsd:element>
    <xsd:element name="contextName" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="switchContextPathResponse">
<xsd:complexType>
  <xsd:sequence>
    <xsd:element name="sessionId" type="ar:SessionIdType" />
  </xsd:sequence>
</xsd:complexType>
```

```
</xsd:element>
<xsd:element name="login" type="ar:LoginType">
</xsd:element>
<xsd:element name="loginResponse">
<xsd:complexType>
  <xsd:sequence>
    <xsd:element name="sessionId" type="ar:SessionIdType" />
  </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="loginFault">
<xsd:complexType>
  <xsd:sequence>
    <xsd:element name="loginFault" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="logout">
<xsd:complexType>
  <xsd:sequence>
    <xsd:element name="sessionId" type="ar:SessionIdType" />
  </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="logoutResponse">
<xsd:complexType>
  <xsd:sequence>
    <xsd:element name="success" type="xsd:boolean" />
  </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="switchContextPathFault">
<xsd:complexType>
  <xsd:sequence>
    <xsd:element name="switchContextPathFault" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
</wsdl:types>
<wsdl:message name="searchRequest">
<wsdl:part element="ar:searchRequest" name="request" />
</wsdl:message>
<wsdl:message name="searchResponse">
<wsdl:part element="ar:searchResponse" name="response" />
</wsdl:message>
<wsdl:message name="searchFault">
<wsdl:part name="tooMany" element="ar:TooManyResultsFaultType" />
</wsdl:message>
<wsdl:message name="retrieveRequest">
<wsdl:part name="request" element="ar:retrieveRequest" />
</wsdl:message>
<wsdl:message name="retrieveResponse">
<wsdl:part name="response" element="ar:retrieveResponse" />
</wsdl:message>
<wsdl:message name="retrieveFault">
<wsdl:part name="notFound" element="ar:EntryNotFoundRetrieveFaultType" />
</wsdl:message>
<wsdl:message name="retrieveFault1">
<wsdl:part name="error" element="ar:ArchiveRetrievalFault" />
</wsdl:message>
<wsdl:message name="searchFault1">
<wsdl:part name="error" element="ar:ArchiveRetrievalFault" />
</wsdl:message>
<wsdl:message name="loginRequest">
<wsdl:part name="parameters" element="ar:login" />
</wsdl:message>
<wsdl:message name="loginResponse">
```

```
<wsdl:part name="parameters" element="ar:loginResponse" />
</wsdl:message>
<wsdl:message name="loginFault">
<wsdl:part name="parameters" element="ar:loginFault" />
</wsdl:message>
<wsdl:message name="switchContextPathRequest">
<wsdl:part name="parameters" element="ar:switchContextPath" />
</wsdl:message>
<wsdl:message name="switchContextPathResponse">
<wsdl:part name="parameters" element="ar:switchContextPathResponse" />
</wsdl:message>
<wsdl:message name="switchContextPathFault">
<wsdl:part name="parameters" element="ar:switchContextPathFault" />
</wsdl:message>
<wsdl:message name="logoutRequest">
<wsdl:part name="parameters" element="ar:logout" />
</wsdl:message>
<wsdl:message name="logoutResponse">
<wsdl:part name="parameters" element="ar:logoutResponse" />
</wsdl:message>
<wsdl:portType name="BoxArchiveRetrieval">
<wsdl:operation name="search">
<wsdl:input message="ar:searchRequest" />
<wsdl:output message="ar:searchResponse" />
<wsdl:fault name="fault" message="ar:searchFault" />
<wsdl:fault name="fault1" message="ar:searchFault1" />
</wsdl:operation>
<wsdl:operation name="retrieve">
<wsdl:input message="ar:retrieveRequest" />
<wsdl:output message="ar:retrieveResponse" />
<wsdl:fault name="fault" message="ar:retrieveFault" />
<wsdl:fault name="fault1" message="ar:retrieveFault1" />
</wsdl:operation>
<wsdl:operation name="login">
<wsdl:input message="ar:loginRequest" />
<wsdl:output message="ar:loginResponse" />
<wsdl:fault name="fault" message="ar:loginFault" />
</wsdl:operation>
<wsdl:operation name="switchContextPath">
<wsdl:input message="ar:switchContextPathRequest" />
<wsdl:output message="ar:switchContextPathResponse" />
<wsdl:fault name="fault" message="ar:switchContextPathFault" />
</wsdl:operation>
<wsdl:operation name="logout">
<wsdl:input message="ar:logoutRequest" />
<wsdl:output message="ar:logoutResponse" />
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="BoxArchiveRetrievalSOAP" type="ar:BoxArchiveRetrieval">
<soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
</>
<wsdl:operation name="search">
<soap:operation
soapAction="http://ws.box.intercope.com/BoxArchiveRetrieval/search" />
<wsdl:input>
<soap:body use="literal" />
</wsdl:input>
<wsdl:output>
<soap:body use="literal" />
</wsdl:output>
<wsdl:fault name="fault">
<soap:fault use="literal" name="fault" />
</wsdl:fault>
<wsdl:fault name="fault1">
<soap:fault use="literal" name="fault1" />
</wsdl:fault>
</wsdl:operation>
<wsdl:operation name="retrieve">
<soap:operation
```

```
soapAction="http://ws.box.intercope.com/BoxArchiveRetrieval/retrieve" />
<wsdl:input>
<soap:body use="literal" />
</wsdl:input>
<wsdl:output>
<soap:body use="literal" />
</wsdl:output>
<wsdl:fault name="fault">
<soap:fault use="literal" name="fault" />
</wsdl:fault>
<wsdl:fault name="fault1">
<soap:fault use="literal" name="fault1" />
</wsdl:fault>
</wsdl:operation>
<wsdl:operation name="login">
<soap:operation
soapAction="http://ws.box.intercope.com/BoxArchiveRetrieval/login" />
<wsdl:input>
<soap:body use="literal" />
</wsdl:input>
<wsdl:output>
<soap:body use="literal" />
</wsdl:output>
<wsdl:fault name="fault">
<soap:fault use="literal" name="fault" />
</wsdl:fault>
</wsdl:operation>
<wsdl:operation name="switchContextPath">
<soap:operation
soapAction="http://ws.box.intercope.com/BoxArchiveRetrieval/switchContextPath"
/>
<wsdl:input>
<soap:body use="literal" />
</wsdl:input>
<wsdl:output>
<soap:body use="literal" />
</wsdl:output>
<wsdl:fault name="fault">
<soap:fault use="literal" name="fault" />
</wsdl:fault>
</wsdl:operation>
<wsdl:operation name="logout">
<soap:operation
soapAction="http://ws.box.intercope.com/BoxArchiveRetrieval/logout" />
<wsdl:input>
<soap:body use="literal" />
</wsdl:input>
<wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="BoxArchiveRetrieval">
<wsdl:port binding="ar:BoxArchiveRetrievalSOAP" name="BoxArchiveRetrievalSOAP">
<soap:address location="http://localhost/" />
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

3 BIC Lookup Web Service

3.1 Introduction

The BIC Lookup Web Service is designed to enable the user to enquire BIC entries stored in the database of BOX.

The service provides two operations:

- Look up if a BIC exists and return the result
- Look up a BIC and return the details of the entry

Please note, if the BOX installation has an IBAN directory, the service will currently only return the information values included in a BIC directory entry.

3.2 Configuration

The Web Service is delivered as a separate package in an ear/war file with the name BicLookupServiceEAR.ear/war.

Before deploying the Web Service, the following configuration tasks need to be performed:

- biclookup.properties
- log4j2.xml

3.2.1 biclookup.properties

The source for the BIC directory is the BOX database.

To connect to the BOX database you have to use a JNDI Connection Pool (configured in the respective WAS).

The following parameters must be configured:

DB.JNDIConnectionPool

DB.JNDIConnectionPool=<JNDI name>

DB.qualifier

DB.qualifier=<the schema/owner/qualifier of the BOX tables>

table.bicdir

table.bicdir=<table name of the BIC directory>

3.2.1.1 Exemplary Configuration

DB.JNDIConnectionPool=MPO_JDBC_POOL

DB.qualifier=box

table.bicdir=BANKPLUSDIR

3.2.2 Log4j.xml

Please refer to chapter 1.4.2 for further configuration information and options.

3.3 Operations

In the request, two operations:

- Looking up if a BIC exists
- Looking up the BIC information

The possible entry formats for both operations are:

- BIC 8 (the branch code will be filled with XXX)
- BIC 8 + 3-digit branch code
- BIC 8 + LT + 3-digit branch code (the LT will be dropped for the operation)

3.3.1 Looking Up if a BIC Exists

The request element is called "existsBicRequest". In the response, the result of the inquiry is a Boolean value (true or false).

3.3.1.1 Request

```
<?xml version="1.0" encoding="UTF-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header/>
  <soap:Body>
    <ns1:existsBicRequest
xmlns:ns1="http://biclookup.ws.intercope.com/BicLookupService/">
      <bic>PTSADESA</bic>
      <exactMatch>true</exactMatch>
    </ns1:existsBicRequest>
  </soap:Body>
</soap:Envelope>
```

3.3.1.2 Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns0:existsBicResponse
xmlns:ns0="http://biclookup.ws.intercope.com/BicLookupService/">true</ns
0:existsBicResponse>
  </S:Body>
</S:Envelope>
```

3.3.1.3 Fault

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns0:Fault xmlns:ns0="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://www.w3.org/2003/05/soap-envelope">
      <faultcode>ns0:Server</faultcode>
      <faultstring>Error querying service.; nested exception is:
java.lang.Exception: Error checking BIC'PTSADESBA'.</faultstring>
      <detail>
        <ns0:existsBicFault
xmlns:ns0="http://biclookup.ws.intercope.com/BicLookupService/">
          <existsBicFault>record lookup failed</existsBicFault>
        </ns0:existsBicFault>
      </detail>
```

```
    </ns0:Fault>
  </S:Body>
</S:Envelope>
```

3.3.2 Looking Up the BIC Information

The request element is called "bicLookupRequest". In the response, the result of the inquiry is a structure containing the BIC entry values.

3.3.2.1 Request

```
<?xml version="1.0" encoding="UTF-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header/>
  <soap:Body>
    <ns1:bicLookupRequest
xmlns:ns1="http://biclookup.ws.intercoope.com/BicLookupService/">
      <bic>PTSADESAX</bic>
      <branch>XXX</branch>
      <institutionName>Intercoope</institutionName>
      <branchInformation>StringValue</branchInformation>
      <cityHeading>Hamburg</cityHeading>
      <subtypeIndicator>StringValue</subtypeIndicator>
      <valueAddedService>StringValue</valueAddedService>
      <extraInformation>StringValue</extraInformation>
      <physicalAddress1>StringValue</physicalAddress1>
      <physicalAddress2>StringValue</physicalAddress2>
      <physicalAddress3>StringValue</physicalAddress3>
      <physicalAddress4>StringValue</physicalAddress4>
      <location>StringValue</location>
      <countryName>StringValue</countryName>
      <pobNumber>StringValue</pobNumber>
      <pobLocation>StringValue</pobLocation>
      <pobCountry>StringValue</pobCountry>

      <externalUserInformation1>StringValue</externalUserInformation1>
      <externalUserInformation2>StringValue</externalUserInformation2>
      <externalUserInformation3>StringValue</externalUserInformation3>
    </ns1:bicLookupRequest>
  </soap:Body>
</soap:Envelope>
```

3.3.2.2 Response

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns0:bicLookupResponse
xmlns:ns0="http://biclookup.ws.intercoope.com/BicLookupService/">
  </S:Body>
</S:Envelope>
```

3.4 Documentation

Port type **BicLookupService** [Source code](#)

1. existsBic

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: <http://biclookup.ws.intercope.com/BicLookupService/existsBic>

Input: existsBicRequest (soap:body, use = literal) [Source code](#)

existsBicRequest type *existsBicRequest*

- **bic** type *string*
The searched BIC. Supported are BIC8 (branch is automatically set to 'XXX'), BIC11 (BIC+branch) and desination string (LT at position 9 is ignored).
- **exactMatch** - optional; type *boolean*
If exactMatch is false (default if omitted) the return value is also 'true' if the searched BIC is a BIC11 (BIC+branch) or a destination string (BIC+LT+branch) and a BIC8 with branch 'XXX' exists.

Output: existsBicResponse (soap:body, use = literal) [Source code](#)

existsBicResponse type *existsBicResponse* type *boolean*

Fault: existsBicFault [Source code](#)

parameters type *existsBicFault*

- existsBicFault type *string*

2. lookupRecord

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: <http://biclookup.ws.intercope.com/BicLookupService/lookupRecord>

Input: bicLookupRequest (soap:body, use = literal) [Source code](#)

bicLookupRequest type *bicLookupRequest*

- **bic** - optional; type *string*
the BIC8
- **branch** - optional; type *string*
three character branch, 'XXX' if not specified
- **institutionName** - optional; type *string*
- **branchInformation** - optional; type *string*
- **cityHeading** - optional; type *string*
- **subtypeIndicator** - optional; type *string*
see the SWIFT document 'BIC Directory Business Code' for a complete list of subtype indicators and their definitions
- **valueAddedService** - optional; type *string*
- **extralInformation** - optional; type *string*
- **physicalAddress1** - optional; type *string*
- **physicalAddress2** - optional; type *string*
- **physicalAddress3** - optional; type *string*
- **physicalAddress4** - optional; type *string*
- **location** - optional; type *string*
- **countryName** - optional; type *string*
- **pobNumber** - optional; type *string*
- **pobLocation** - optional; type *string*
- **pobCountry** - optional; type *string*
- **externalUserInformation1** - optional; type *string*
- **externalUserInformation2** - optional; type *string*
- **externalUserInformation3** - optional; type *string*

Output: bicLookupResponse (soap:body, use = literal) [Source code](#)

bicLookupResponse type *bicLookupResponse*

- **bicRecord** - optional, unbounded; type *bicRecord*

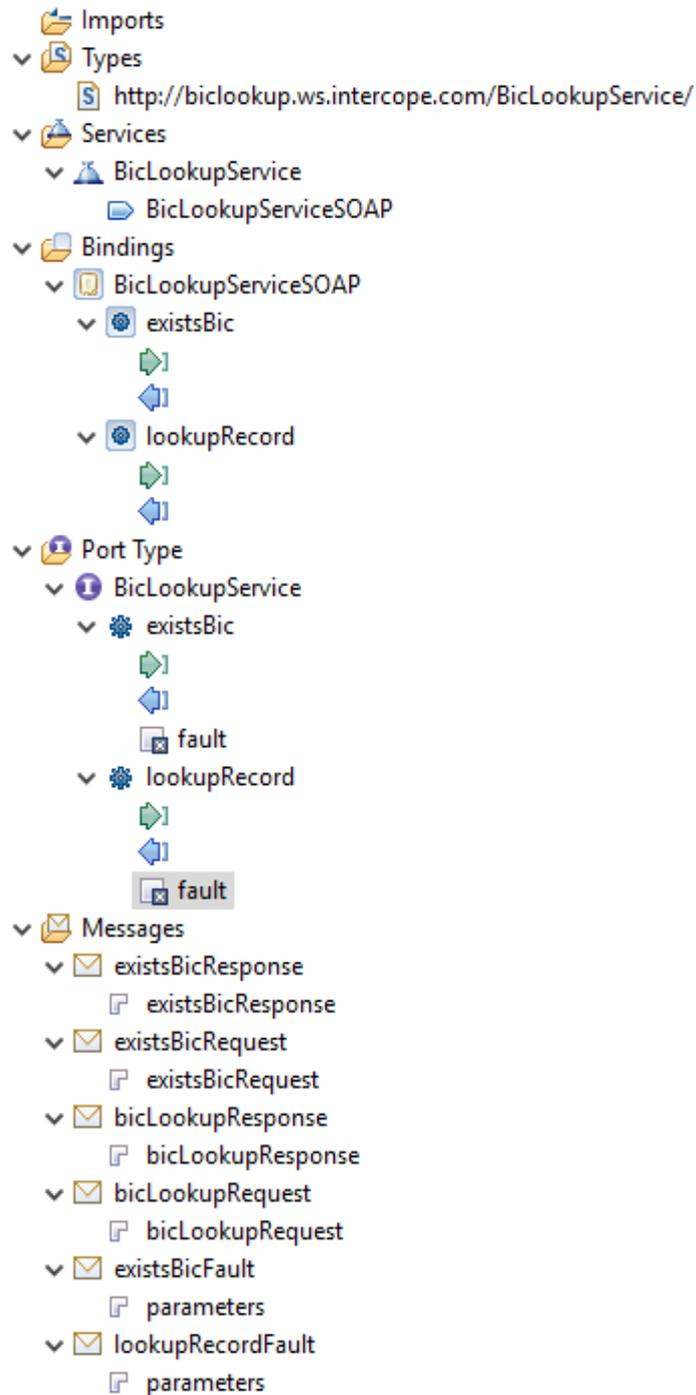
Fault: lookupRecordFault [Source code](#)

parameters type *lookupRecordFault*

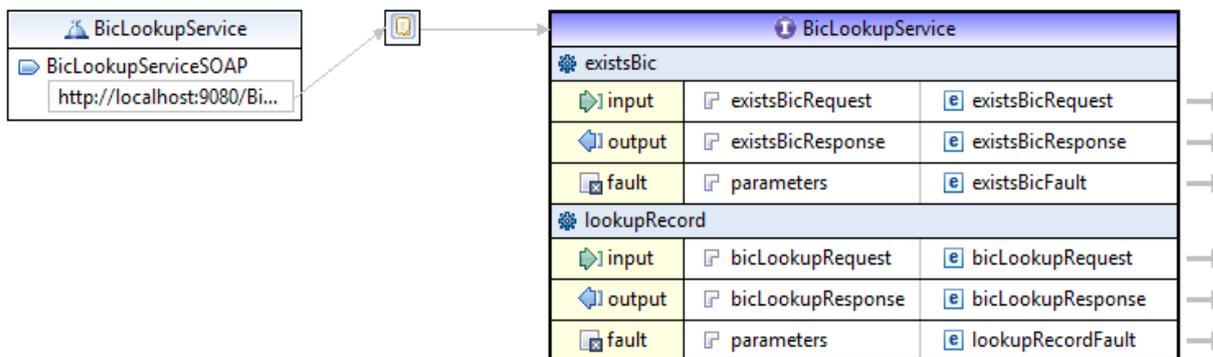
- lookupRecordFault type *string*

3.5 Programmatical Structure

3.5.1 Outline

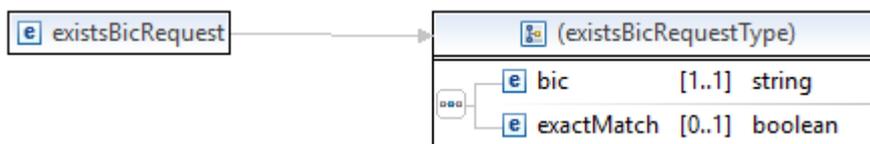


3.5.2 Generic Attributes



3.5.3 existsBic Operation

3.5.3.1 Request

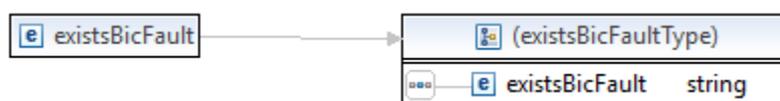


bic	The searched BIC. Supported are BIC8 (branch is automatically set to 'XXX'), BIC11 (BIC+branch) and destination string (LT at position 9 is ignored)
exactMatch	True, False If exactMatch is false (default if omitted) the return value is also 'true' if the searched BIC is a BIC11 (BIC+branch) or a destination string (BIC+LT+branch) and a BIC8 with branch 'XXX' exists

3.5.3.2 Response



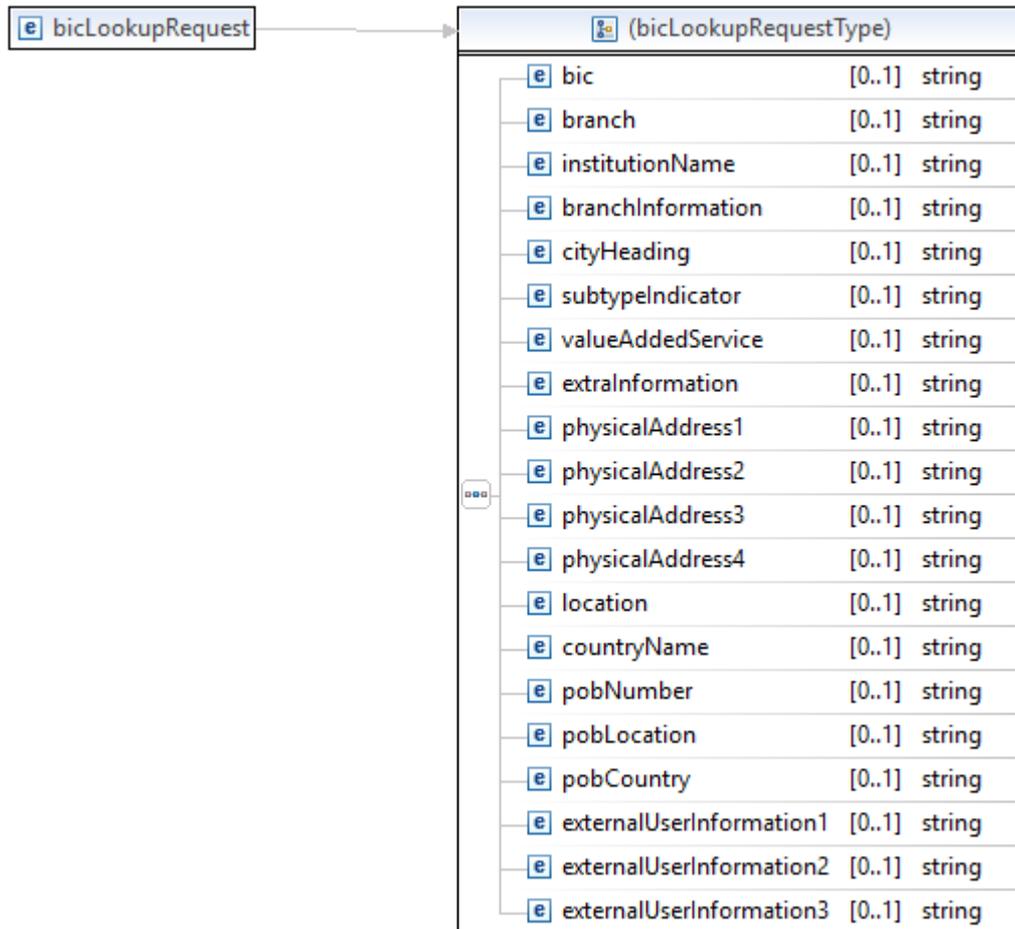
3.5.3.3 Fault



existsBicFault	String 'record lookup failed' <code><faultstring>Error querying service.; nested exception is: java.lang.Exception: Error checking BIC'PTSADESBA'.</faultstring></code>
----------------	--

3.5.4 lookupRecord Operation

3.5.4.1 Request



<code>bic</code>	The searched BIC. Supported are BIC8 (branch is automatically set to 'XXX'), BIC11 (BIC+branch) and destination string (LT at position 9 is ignored)
<code>branch</code>	type="xsd:string"
<code>institutionName</code>	type="xsd:string"
<code>branchInformation</code>	type="xsd:string"
<code>cityHeading</code>	type="xsd:string"
<code>subtypeIndicator</code>	see the SWIFT document 'BIC Directory Business Code' for a complete list of subtype indicators and their definitions
<code>valueAddedService</code>	type="xsd:string"
<code>extraInformation</code>	type="xsd:string"
<code>physicalAddress1</code>	type="xsd:string", Street Address1
<code>physicalAddress2</code>	type="xsd:string", Street Address2
<code>physicalAddress3</code>	type="xsd:string", Street Address3
<code>physicalAddress4</code>	type="xsd:string", Street Address4
<code>location</code>	type="xsd:string",

INTERCOPE

countryName	type="xsd:string", Country
pobNumber	type="xsd:string",
pobLocation	type="xsd:string",
pobCountry	type="xsd:string",
externalUserInformation1	type="xsd:string",
externalUserInformation2	type="xsd:string",
externalUserInformation3	type="xsd:string",

3.5.4.2 Response



BicRecord	minOccurs="0" maxOccurs="unbounded"
-----------	-------------------------------------

3.5.4.3 Fault



lookupRecordFault	message="bic:lookupRecordFault"
-------------------	---------------------------------

3.6 Appendix

3.6.1 BicLookup wsdl

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
xmlns:bic="http://biclookup.ws.intercope.com/BicLookupService/"
xmlns:wss="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="BicLookupService"
targetNamespace="http://biclookup.ws.intercope.com/BicLookupService/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
<wsdl:types>
<xsd:schema
targetNamespace="http://biclookup.ws.intercope.com/BicLookupService/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:element name="existsBicResponse" type="xsd:boolean" />
<xsd:element name="existsBicRequest">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="bic"
type="xsd:string" minOccurs="1" maxOccurs="1">
<xsd:annotation>
<xsd:documentation>The searched BIC. Supported are BIC8 (branch
is automatically set to 'XXX'), BIC11 (BIC+branch) and destination string (LT at
position 9 is ignored).</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="exactMatch"
type="xsd:boolean" minOccurs="0" maxOccurs="1">
<xsd:annotation>
<xsd:documentation>If exactMatch is false (default if omitted)
the return value is also 'true' if the searched BIC is a BIC11 (BIC+branch) or a
destination string (BIC+LT+branch) and a BIC8 with branch 'XXX'
exists.</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="bicRecord">
<xsd:complexType>
<xsd:sequence>
<xsd:element maxOccurs="1" minOccurs="1"
name="bic" type="xsd:string">
<xsd:annotation>
<xsd:documentation>
the BIC8
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element maxOccurs="1" minOccurs="1"
name="branch" type="xsd:string">
<xsd:annotation>
<xsd:documentation>
three character branch, 'XXX' if not
specified
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element maxOccurs="1" minOccurs="1"
name="creation" type="xsd:dateTime">
<xsd:annotation>
<xsd:documentation>
record creation timestamp
</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</wsdl:types>
</wsdl:definitions>
```

```
<xsd:element maxOccurs="1" minOccurs="1"
  name="lastUpdate" type="xsd:dateTime">
  <xsd:annotation>
    <xsd:documentation>
      last update timestamp
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element maxOccurs="1" minOccurs="1"
  name="institutionName" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="1"
  name="branchInformation" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="1"
  name="cityHeading" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="1"
  name="subtypeIndicator" type="xsd:string">
  <xsd:annotation>
    <xsd:documentation>
      see the SWIFT document 'BIC
      Directory Business Code' for a
      complete list of subtype indicators
      and their definitions
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element maxOccurs="1" minOccurs="1"
  name="valueAddedService" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="1"
  name="extraInformation" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="1"
  name="physicalAddress1" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="1"
  name="physicalAddress2" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="1"
  name="physicalAddress3" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="1"
  name="physicalAddress4" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="1"
  name="location" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="1"
  name="countryName" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="1"
  name="pobNumber" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="1"
  name="pobLocation" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="1"
  name="pobCountry" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="0"
  name="externalUserInformation1" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="0"
  name="externalUserInformation2" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="0"
  name="externalUserInformation3" type="xsd:string" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="bicLookupRequest">
<xsd:complexType>
<xsd:sequence>
  <xsd:element maxOccurs="1" minOccurs="0"
    name="bic" type="xsd:string">
    <xsd:annotation>
      <xsd:documentation>
        the BIC8
      </xsd:documentation>
    </xsd:annotation>
  </xsd:element>
  <xsd:element maxOccurs="1" minOccurs="0"
    name="branch" type="xsd:string">
```

```
<xsd:annotation>
  <xsd:documentation>
    three character branch, 'XXX' if not
    specified
  </xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element maxOccurs="1" minOccurs="0"
  name="institutionName" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="0"
  name="branchInformation" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="0"
  name="cityHeading" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="0"
  name="subtypeIndicator" type="xsd:string">
  <xsd:annotation>
    <xsd:documentation>
      see the SWIFT document 'BIC
      Directory Business Code' for a
      complete list of subtype indicators
      and their definitions
    </xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element maxOccurs="1" minOccurs="0"
  name="valueAddedService" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="0"
  name="extraInformation" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="0"
  name="physicalAddress1" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="0"
  name="physicalAddress2" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="0"
  name="physicalAddress3" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="0"
  name="physicalAddress4" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="0"
  name="location" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="0"
  name="countryName" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="0"
  name="pobNumber" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="0"
  name="pobLocation" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="0"
  name="pobCountry" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="0"
  name="externalUserInformation1" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="0"
  name="externalUserInformation2" type="xsd:string" />
<xsd:element maxOccurs="1" minOccurs="0"
  name="externalUserInformation3" type="xsd:string" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="bicLookupResponse">
<xsd:complexType>
<xsd:sequence>
  <xsd:element ref="bic:bicRecord" minOccurs="0" maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="existsBicFault">
<xsd:complexType>
<xsd:sequence>
  <xsd:element name="existsBicFault"
    type="xsd:string">
  </xsd:element>
</xsd:sequence>
</xsd:element>
</xsd:sequence>
```

```
</xsd:complexType>
</xsd:element>
<xsd:element name="lookupRecordFault">
<xsd:complexType>
<xsd:sequence>

    <xsd:element name="lookupRecordFault"
        type="xsd:string">
    </xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
</wsdl:types>
<wsdl:message name="existsBicResponse">
<wsdl:part element="bic:existsBicResponse"
name="existsBicResponse" />
</wsdl:message>
<wsdl:message name="existsBicRequest">
<wsdl:part element="bic:existsBicRequest"
name="existsBicRequest" />
</wsdl:message>
<wsdl:message name="bicLookupResponse">
<wsdl:part name="bicLookupResponse" element="bic:bicLookupResponse"></wsdl:part>
</wsdl:message>
<wsdl:message name="bicLookupRequest">
<wsdl:part name="bicLookupRequest"
element="bic:bicLookupRequest">
</wsdl:part>
</wsdl:message>
<wsdl:message name="existsBicFault">
<wsdl:part name="parameters" element="bic:existsBicFault"></wsdl:part>
</wsdl:message>
<wsdl:message name="lookupRecordFault">
<wsdl:part name="parameters" element="bic:lookupRecordFault"></wsdl:part>
</wsdl:message>
<wsdl:portType name="BicLookupService">
<wsdl:operation name="existsBic">
<wsdl:input message="bic:existsBicRequest" />
<wsdl:output message="bic:existsBicResponse" />
<wsdl:fault name="fault" message="bic:existsBicFault"></wsdl:fault>
</wsdl:operation>
<wsdl:operation name="lookupRecord">
<wsdl:input message="bic:bicLookupRequest" >
<wsdl:documentation>The input structure is used to filter on the specified
values.
String values are automatically enclosed by wildcards.</wsdl:documentation>
</wsdl:input>
<wsdl:output message="bic:bicLookupResponse" />
<wsdl:fault name="fault" message="bic:lookupRecordFault"></wsdl:fault>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="BicLookupServiceSOAP"
type="bic:BicLookupService">
<soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http" />
<wsdl:operation name="existsBic">
<soap:operation
soapAction="http://biclookup.ws.intercope.com/BicLookupService/existsBic" />
<wsdl:input>
<soap:body use="literal" />
</wsdl:input>
<wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
```

```
<wsdl:operation name="lookupRecord">
  <soap:operation
    soapAction="http://biclookup.ws.intercope.com/BicLookupService/lookupRecord" />
  <wsdl:input >
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="BicLookupService">
  <wsdl:port binding="bic:BicLookupServiceSOAP"
    name="BicLookupServiceSOAP">
    <soap:address
      location="http://localhost:9080/BicLookupService/services/BicLookupServiceSOAP"
    />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

4 BOX Messaging Web Service

4.1 Introduction

The **Box Messaging Web Service** works as a request-response operation, where an operation request is sent from a client application to the Web Application Server, the service performs the requested operation and returns a response to the Client.

The **Box Messaging Web Service** offers the following operations:

- **Login**
With this operation, a login to the Box Server can be established. The provided user will be used for the `routeMessageToAbRecipient` operation and must therefore have the appropriate UPM access rights.
- **switchContextPath**
With this operation, the user context path of a logged in user can be switched. This may become necessary if the home context path does not have sufficient UPM access rights that are required for the `routeMessageToAbRecipient` operation.
- **routeMessageToAbRecipient**
With this operation, a message can be routed to an address book recipient. This operation also includes the option to perform a login, or to use a current login session.
- **logout**

4.2 Configuration

The configuration for the Box Messaging is performed in two files, located in `..\WEB-INF\classes`:

• `BoxMessagingService.properties`

• `log4j.properties`

4.2.1 BoxMessagingService.properties

The following parameters are available for `BoxMessagingService.properties`:

bms.api.configuration:

Definition of the Box Messaging Service configuration file.

- Type: String
- Required: No

Example:

```
bms.api.configuration=configuration.properties
```

bms.replacement-file-resolve-via:

Definition of how the replacement file is resolved. If this parameter is not set, replacements will not be used.

- Type: String
- Required: No

Possible values:

- **classpath** - use classpath to find replacement file
- **path** - use filename directly

Example:

```
bms.replacement-file-resolve-via=classpath
```

bms.replacement-file.name:

Definition of the replacement file name.

- Type: String
- Required: No

Example:

```
bms.replacement-file.name=replace.rpl
```

bms.session.maxIdleTimeMillis:

Definition of the maximum Idle Time in milli seconds.

- Type: Numeric
- Required: No

Example:

```
bms.session.maxIdleTimeMillis=600000
```

bms.api.security-zip:

Definition of the the path and filename of the 'security.zip', if omitted '/WEB-INF/security.zip' is used

- Type: String
- Required: Yes

Example:

```
bms.api.security-zip=/path/and/filename/of/security.zip
```

bms.api.mek:

Definition of the path and filename of the 'mek.enc', if omitted the file in security.zip is used

- Type: String
- Required: Yes

Example:

```
bms.api.mek=samplemek.enc
```

4.2.1.1 Exemplary Configuration

```
# BOX api configuration file, default value is
'configuration.properties'
# Note: file must be accessible via classpath.
bms.api.configuration=configuration.properties

# specifies how to resolve the replacement file
# valid values:
# - classpath = use classpath to find replacement file
# - path      = use filename directly
# Note: if left commented out, replacement is not used
bms.replacement-file.resolve-via=path

# name of the replacement file, default is 'replace.rpl'
bms.replacement-file.name=/opt/box/server/config/replace.rpl

# idle time (in milliseconds) after which a user session is closed
automatically
# default: 600000 (10 minutes)
bms.session.maxIdleTimeMillis=600000

# specifies the path and filename of the 'security.zip', if omitted
'/WEB-INF/security.zip' is used
#bms.api.security-zip=/path/and/filename/of/security.zip

# specifies the path and filename of the 'mek.enc', if omitted the file
in security.zip is used
#bms.api.mek
```

Please note, the usage of the replacement file and the further configuration parameters, which are part of the configuration.properties file are to be found in chapter 1.4 .

4.2.2 Log4j.xml

Please refer to chapter 1.4.2 for further configuration information and options.

4.3 Operations

4.3.1 Login Operation

4.3.1.1 Request

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:box="http://box.intercope.com/boxws/">
  <soapenv:Header/>
  <soapenv:Body>
    <box:login>
      <clientPrefix>demofin</clientPrefix>
      <userName>bank1</userName>
      <secret>
        <plain>admin</plain>
      </secret>
    </box:login>
  </soapenv:Body>
</soapenv:Envelope>
```

4.3.1.2 Response

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns0:loginResponse xmlns:ns0="http://box.intercope.com/boxws/">
      <sessionId>f9c37415072b3399abc87bb3ddc6347d</sessionId>
    </ns0:loginResponse>
  </S:Body>
</S:Envelope>
```

4.3.1.3 Fault

```
<?xml version="1.0" encoding="UTF-8"?><S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns0:Fault
xmlns:ns0="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="http://www.w3.org/2003/05/soap-envelope">
      <faultcode>ns0:Server</faultcode>
      <faultstring>login failed</faultstring>
      <detail>
        <ns0:loginFault
xmlns:ns0="http://box.intercope.com/boxws/">
          <loginFault>Wrong password.</loginFault>
        </ns0:loginFault>
      </detail>
    </ns0:Fault>
  </S:Body>
</S:Envelope>
```

Please note, if the login failed, the response will have an element loginFault which provides a short text.

4.3.2 switchContextPath Operation

4.3.2.1 Request

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:box="http://box.intercope.com/boxws/">
  <soapenv:Header/>
  <soapenv:Body>
    <box:switchContextPath>
      <sessionId>?</sessionId>
      <contextName>?</contextName>
    </box:switchContextPath>
  </soapenv:Body>
</soapenv:Envelope>
```

4.3.2.2 Response:

```
<S:Body><S:Envelope
xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <ns0:loginResponse xmlns:ns0="http://box.intercope.com/boxws/">
    <sessionId>f9c37415072b3399abc87bb3ddc6347d</sessionId>
  </ns0:loginResponse>
</S:Body>
</S:Envelope>
```

Please note, if the switch context path operation fails, the response will have an element switchContextPath_fault which provides a short text (e.g. "session not found").

4.3.3 routeMessageToAbRecipient Operation

4.3.3.1 Request:

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:box="http://box.intercope.com/boxws/">
  <soapenv:Header/>
  <soapenv:Body>
    <box:routeMessageToAbRecipient>
      <!--You have a CHOICE of the next 2 items at this level-->
      <box:login>
        <clientPrefix?</clientPrefix>
        <userName?</userName>
        <secret>
          <plain?</plain>
        </secret>
      </box:login>
      <sessionId?</sessionId>
      <messageId?</messageId>
      <workflowEntryPoint>
        <patternLabel?</patternLabel>
        <!--Optional:-->
        <cvAuthListMode?</cvAuthListMode>
      </workflowEntryPoint>
      <recipientList>
        <!--1 or more repetitions:-->
        <singleRecipientReference>
          <uniqueAddressbookName?</uniqueAddressbookName>
          <uniqueRecipientName?</uniqueRecipientName>
          <!--Optional:-->
          <addressReference>
            <addressDisplayName?</addressDisplayName>
            <!--Optional:-->
            <addressType?</addressType>
          </addressReference>
        </singleRecipientReference>
      </recipientList>
      <!--Optional:-->
      <operatorComment?</operatorComment>
    </box:routeMessageToAbRecipient>
  </soapenv:Body>
</soapenv:Envelope>

```

4.3.3.2 Response

Please note, if the route operation fails, the response will have an element `RouteMessageToAbRecipient_fault` which provides a short text. If the error occurred due to a login failure, the response will be similar to the response of the login operation.

4.3.3.2.1 Fault SOAP

```

<?xml version="1.0" encoding="UTF-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header/>
  <soap:Body>
    <ns1:routeMessageToAbRecipient
xmlns:ns1="http://box.intercope.com/boxws/">
      <sessionId>92dbfb712212cd993a0abbd356f98d9</sessionId>
      <messageId>20</messageId>
      <workflowEntryPoint>
        <patternLabel>FIN-Routing-Output-Msgs</patternLabel>
        <cvAuthListMode>NO_UPDATE</cvAuthListMode>
      </workflowEntryPoint>
      <recipientList>
        <singleRecipientReference>

```

```

    <uniqueAddressbookName>FINAddressBook</uniqueAddressbookName>
ame>
    <uniqueRecipientName>TO.BE2</uniqueRecipientName>
    <addressReference>
      <addressDisplayName>TO.BE2</addressDisplayName>
      <addressType>UNDEFINED</addressType>
    </addressReference>
  </singleRecipientReference>
  <singleRecipientReference>
    <uniqueAddressbookName>StringValue</uniqueAddressbookName>
ame>
    <uniqueRecipientName>StringValue</uniqueRecipientName>
e>
    <addressReference>
      <addressDisplayName>StringValue</addressDisplayName>
ame>
      <addressType>FAXPSTN</addressType>
    </addressReference>
  </singleRecipientReference>
</recipientList>
  <operatorComment>Test</operatorComment>
</nsl:routeMessageToAbRecipient>
</soap:Body>
</soap:Envelope>

```

4.3.3.2 Fault Java Stack

```

java.lang.RuntimeException:
com.sun.xml.ws.fault.ServerSOAPFaultException: Client received SOAP
Fault from server: error loading MPS Please see the server log to find
more detail regarding exact cause of the failure. at
com.oracle.webservices.testclient.core.ws.cdf.SOAPBindingEngine.call(SOAP
BindingEngine.java:87) at
com.oracle.webservices.testclient.wstest.InvocationDelegate.invoke(Invoc
ationDelegate.java:46)

```

4.3.4 Logout Operation

4.3.4.1 Request

```

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:box="http://box.intercope.com/boxws/">
  <soapenv:Header/>
  <soapenv:Body>
    <box:logout>
      <sessionId>ead56b9cc885edb4e1c21f29a90b724e</sessionId>
    </box:logout>
  </soapenv:Body>
</soapenv:Envelope>

```

4.3.4.2 Response:

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns0:logoutResponse xmlns:ns0="http://box.intercope.com/boxws/">
      <success>true</success>
    </ns0:logoutResponse>
  </S:Body>
</S:Envelope>

```

```
</ns0:logoutResponse>  
</S:Body>  
</S:Envelope>
```

4.4 Documentation

Operations

Port type **boxws** [Source code](#)

1. login

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: <http://box.intercope.com/boxws/login>

Input: loginRequest (soap:body, use = literal) [Source code](#)

```
parameters type login
  ▪ clientPrefix type string
  ▪ userName type string
  ▪ secret type SecretType
    ▪ plain type string
      plain text password
```

Output: loginResponse (soap:body, use = literal) [Source code](#)

```
parameters type loginResponse
  ▪ sessionId type SessionIdType - type string with restriction
```

Fault: loginFault (soap:fault, use = literal) [Source code](#)

```
parameters type loginFault
  ▪ loginFault type string
```

2. logout

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: <http://box.intercope.com/boxws/logout>

Input: logoutRequest (soap:body, use = literal) [Source code](#)

```
parameters type logout
  ▪ sessionId type SessionIdType - type string with restriction
```

Output: logoutResponse (soap:body, use = literal) [Source code](#)

```
parameters type logoutResponse
  ▪ success type boolean
```

3. routeMessageToAbRecipient

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: <http://box.intercope.com/boxws/routeMessageToAbRecipient>

Input: routeMessageToAbRecipientRequest (soap:body, use = literal) [Source code](#)

```

parameters type routeMessageToAbRecipient
  login type login
  sessionId type SessionIdType - type string with restriction
  messageId type MessageIdType - type int with restriction minInclusive(1)
  workflowEntryPoint type WorkflowEntryPointType
  This is the pattern label of the IPS used as entry point for the added SBI.
  patternLabel type ShortNameType - type string with restriction minLength(1) maxLength(127)
  cvAuthListMode - optional; type MP_CVAUTHLIST_MODE - type string with restriction - enum {
    'NO_UPDATE', 'RESET_ADD', 'ADD', 'RESET_ONLY' }
  recipientList type RecipientListType
  singleRecipientReference - unbounded; type SingleRecipientReferenceType
  uniqueAddressbookName type ShortNameType - type string with restriction minLength(1)
    maxLength(127)
  The name of the addressbook attached to the company node of the acting user.
  uniqueRecipientName type ShortNameType - type string with restriction minLength(1) maxLength(127)
  The unique name of the recipient within the addressbook.
  addressReference - optional; type AddressReferenceType
  addressDisplayName type DisplayNameType - type string with restriction minLength(1)
    maxLength(255)
  Display name of the address.
  addressType - optional; type MP_ADR_TYPE - type string with restriction - enum { 'FAXPSTN',
    'SMS', 'DB', 'TELEX', 'SMTP', 'DOMINO', 'EXCHANGE', 'FILE_EXCH', 'PRINTER', 'IBM_CM_V1',
    'SWIFT', 'IACT', 'FACT', 'MQ', 'MEADOW_APPL', 'SAA_BA', 'RMA', 'MWH', 'FACT_BA',
    'WS_ADAPTER' }
  Type of the address. Can be used to distinguish between addresses in case their display names are
  not unique.
  Optional address selection data.
  operatorComment - optional; type string
  
```

Output: routeMessageToAbRecipientResponse (soap:body, use = literal) [Source code](#)

```

parameters type routeMessageToAbRecipientResponse
  sessionId type SessionIdType - type string with restriction
  
```

Fault: routeMessageToAbRecipient_faultMsg (soap:fault, use = literal) [Source code](#)

```

parameters type routeMessageToAbRecipient_fault
  routeMessageToAbRecipient_fault type string
  
```

Fault: routeMessageToAbRecipient_fault1Msg (soap:fault, use = literal) [Source code](#)

```

parameters type loginFault
  loginFault type string
  
```

4. switchContextPath

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: <http://box.intercope.com/boxws/switchContextPath>

Input: switchContextPathRequest (soap:body, use = literal) [Source code](#)

```

parameters type switchContextPath
  sessionId type SessionIdType - type string with restriction
  contextName type string
  
```

Output: switchContextPathResponse (soap:body, use = literal) [Source code](#)

```

parameters type switchContextPathResponse
  sessionId type SessionIdType - type string with restriction
  
```

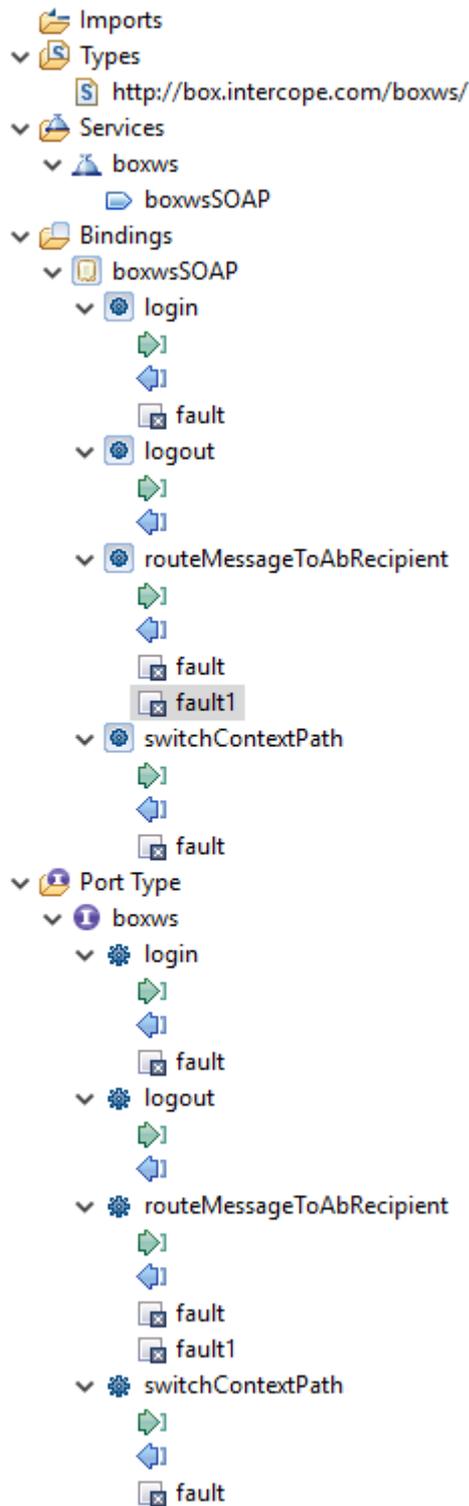
Fault: switchContextPath_faultMsg (soap:fault, use = literal) [Source code](#)

```

parameters type switchContextPath_fault
  switchContextPath_fault type string
  
```

4.5 Programmatical Structure

4.5.1 Outline



4.5.3 Login Operation

4.5.3.1 Login



clientPrefix	Client Prefix name of the institute.
userName	UserName of the account, which is used for the retrieval.
secret	Password for the account, it can be plain text or encrypted.

4.5.3.2 LoginResponse



sessionId	During the log-in, an actual user object is created. The session Id is the pointer to this object. If 'true' login operation will create a JSESSIONID which can be used by load balancers to correctly maintain session stickiness
-----------	--

4.5.3.3 loginFault

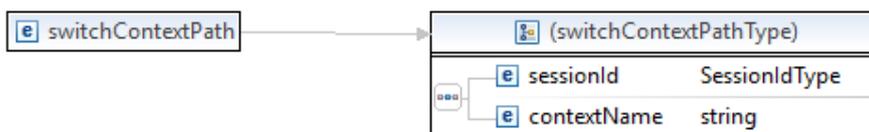
If an error occurs during log-in, the error text is returned.



4.5.4 switchContextPath Operation

Optional method, if multiple context paths for user object are defined. Use this method to select the user context.

4.5.4.1 switchContextPath



sessionId	During the log-in, an actual user object is created. The session Id is the pointer to this object. If 'true' login operation will create a JSESSIONID which can be used by load balancers to correctly maintain session stickiness
-----------	--

contextName	Name of the User Context Path.
-------------	--------------------------------

4.5.4.2 switchContextPathResponse



sessionId	During the log-in, an actual user object is created. The session Id is the pointer to this object. If 'true' login operation will create a JSESSIONID which can be used by load balancers to correctly maintain session stickiness
-----------	--

4.5.4.3 switchContextPathFault

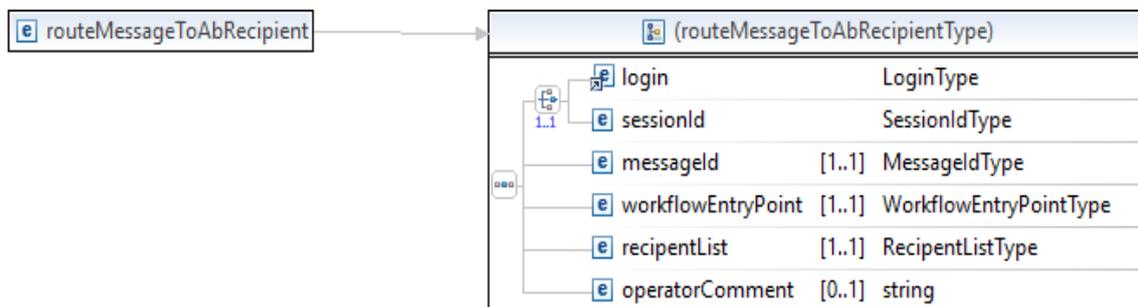


switchContextPathFault	Error description of the switchContextPath failure.
------------------------	---

4.5.5 routeMessageToAbRecipient Operation

Routing a message by message ID to an address book recipient.

4.5.5.1 Request



login	The session Id is the pointer to the user object.
sessionId	pointer to the user object. If 'true' login operation will create a JSESSIONID which can be used by load balancers to correctly maintain session stickiness
messageId	Message ID number
workflowEntryPoint	Point in the workflow, from which the message should be routed onwards

INTERCOPE

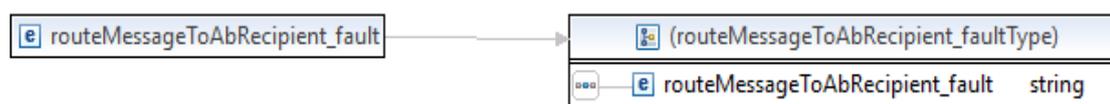
recipientList	List of address book recipients
operatorComment	Comment attached by the operator

4.5.5.2 Response



sessionId	During the log-in, an actual user object is created. The session Id is the pointer to this object. If 'true' login operation will create a JSESSIONID which can be used by load balancers to correctly maintain session stickiness
-----------	--

4.5.5.3 Fault



routeMessageToAbRecipient_fault	Error description of the routeMessageToAbRecipient failure. Example: <pre><faultcode>ns0:Server</faultcode> <faultstring>session not found</faultstring></pre>
---------------------------------	--

4.6 Appendix

4.6.1 Box Message Service wds1

```
<?xml version="1.0" encoding="UTF-8"?><wsdl:definitions name="boxws"
targetNamespace="http://box.intercope.com/boxws/"
xmlns:boxws="http://box.intercope.com/boxws/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <xsd:schema targetNamespace="http://box.intercope.com/boxws/">
      <xsd:element name="login" type="boxws:LoginType" />
      <xsd:element name="loginResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="sessionId" type="boxws:SessionIdType"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="loginFault">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="loginFault" type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="logout">
```

```

<xsd:complexType>
  <xsd:sequence>
    <xsd:element name="sessionId" type="boxws:SessionIdType"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="logoutResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="success" type="xsd:boolean"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:complexType name="SecretType">
  <xsd:choice>
    <xsd:element name="plain" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation>plain text password</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:choice>
</xsd:complexType>
<xsd:simpleType name="SessionIdType">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:element name="routeMessageToAbRecipient">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:choice maxOccurs="1" minOccurs="1">
        <xsd:element ref="boxws:login"/>
        <xsd:element name="sessionId" type="boxws:SessionIdType">
</xsd:element>
          <xsd:choice>
            <xsd:element maxOccurs="1" minOccurs="1" name="messageId"
type="boxws:MessageIdType"> </xsd:element>
            <xsd:element maxOccurs="1" minOccurs="1"
name="workflowEntryPoint" type="boxws:WorkflowEntryPointType">
              <xsd:annotation>
                <xsd:documentation> This is the pattern label of the IPS
used as entry point for the added SBI. </xsd:documentation>
              </xsd:annotation>
            </xsd:element>
            <xsd:element maxOccurs="1" minOccurs="1" name="recipientList"
type="boxws:RecipientListType"/>
            <xsd:element maxOccurs="1" minOccurs="0"
name="operatorComment" type="xsd:string"> </xsd:element>
          </xsd:choice>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="routeMessageToAbRecipientResponse">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="sessionId" type="boxws:SessionIdType"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:complexType name="LoginType">
      <xsd:sequence>
        <xsd:element maxOccurs="1" minOccurs="1" name="clientPrefix"
type="xsd:string">
          <xsd:annotation>
            <xsd:documentation/>
          </xsd:annotation>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>

```

```
        </xsd:annotation>
    </xsd:element>
    <xsd:element maxOccurs="1" minOccurs="1" name="userName"
type="xsd:string">
        <xsd:annotation>
            <xsd:documentation/>
        </xsd:annotation>
    </xsd:element>
    <xsd:element maxOccurs="1" minOccurs="1" name="secret"
type="boxws:SecretType">
        <xsd:annotation>
            <xsd:documentation/>
        </xsd:annotation>
    </xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="MessageIdType">
    <xsd:restriction base="xsd:int">
        <xsd:minInclusive value="1"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ShortNameType">
    <xsd:restriction base="xsd:string">
        <xsd:minLength value="1"/>
        <xsd:maxLength value="127"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="DisplayNameType">
    <xsd:restriction base="xsd:string">
        <xsd:minLength value="1"/>
        <xsd:maxLength value="255"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:element name="routeMessageToAbRecipient_fault">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="routeMessageToAbRecipient_fault"
type="xsd:string"> </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="routeMessageToAbRecipient_fault1">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="routeMessageToAbRecipient_fault1"
type="xsd:string"> </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="switchContextPath">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="sessionId" type="boxws:SessionIdType">
</xsd:element>
            <xsd:element name="contextName" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="switchContextPathResponse">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="sessionId" type="boxws:SessionIdType"/>

```

```

        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="switchContextPath_fault">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="switchContextPath_fault"
type="xsd:string"> </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:complexType name="AddressReferenceType">
    <xsd:sequence>
        <xsd:element maxOccurs="1" minOccurs="1"
name="addressDisplayName" type="boxws:DisplayNameType">
            <xsd:annotation>
                <xsd:documentation>Display name of the
address.</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element maxOccurs="1" minOccurs="0" name="addressType"
type="boxws:MP_ADR_TYPE">
            <xsd:annotation>
                <xsd:documentation>Type of the address. Can be used to
distinguish between addresses in case their display names are not
unique.</xsd:documentation>
            </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="WorkflowEntryPointType">
    <xsd:sequence>
        <xsd:element maxOccurs="1" minOccurs="1" name="patternLabel"
type="boxws:ShortNameType"> </xsd:element>
        <xsd:element maxOccurs="1" minOccurs="0" name="cvAuthListMode"
type="boxws:MP_CVAUTHLIST_MODE"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="MP_CVAUTHLIST_MODE">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="NO_UPDATE"/>
        <xsd:enumeration value="RESET_ADD"/>
        <xsd:enumeration value="ADD"/>
        <xsd:enumeration value="RESET_ONLY"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="MP_ADR_TYPE">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="FAXPSTN"/>
        <xsd:enumeration value="SMS"/>
        <xsd:enumeration value="DB"/>
        <xsd:enumeration value="TELEX"/>
        <xsd:enumeration value="SMTP"/>
        <xsd:enumeration value="DOMINO"/>
        <xsd:enumeration value="EXCHANGE"/>
        <xsd:enumeration value="FILE_EXCH"/>
        <xsd:enumeration value="PRINTER"/>
        <xsd:enumeration value="IBM_CM_V1"/>
        <xsd:enumeration value="SWIFT"/>
        <xsd:enumeration value="IACT"/>
        <xsd:enumeration value="FACT"/>
        <xsd:enumeration value="MQ"/>
    </xsd:restriction>
</xsd:simpleType>

```

```
<xsd:enumeration value="MEADOW_APPL"/>
<xsd:enumeration value="SAA_BA"/>
<xsd:enumeration value="RMA"/>
<xsd:enumeration value="MWH"/>
<xsd:enumeration value="FACT_BA"/>
<xsd:enumeration value="WS_ADAPTER"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="RecipientListType">
  <xsd:sequence>
    <xsd:element maxOccurs="unbounded" minOccurs="1"
name="singleRecipientReference"
type="boxws:SingleRecipientReferenceType"> </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="SingleRecipientReferenceType">
  <xsd:sequence>
    <xsd:element maxOccurs="1" minOccurs="1"
name="uniqueAddressbookName" type="boxws:ShortNameType">
      <xsd:annotation>
        <xsd:documentation>The name of the addressbook attached to
the company node of the acting user.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element maxOccurs="1" minOccurs="1"
name="uniqueRecipientName" type="boxws:ShortNameType">
      <xsd:annotation>
        <xsd:documentation>The unique name of the recipient within
the addressbook.</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element maxOccurs="1" minOccurs="0"
name="addressReference" type="boxws:AddressReferenceType">
      <xsd:annotation>
        <xsd:documentation>Optional address selection data.
</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>
</wsdl:types>
<wsdl:message name="loginRequest">
  <wsdl:part element="boxws:login" name="parameters"/>
</wsdl:message>
<wsdl:message name="loginResponse">
  <wsdl:part element="boxws:loginResponse" name="parameters"/>
</wsdl:message>
<wsdl:message name="loginFault">
  <wsdl:part element="boxws:loginFault" name="parameters"/>
</wsdl:message>
<wsdl:message name="logoutRequest">
  <wsdl:part element="boxws:logout" name="parameters"/>
</wsdl:message>
<wsdl:message name="logoutResponse">
  <wsdl:part element="boxws:logoutResponse" name="parameters"/>
</wsdl:message>
<wsdl:message name="routeMessageToAbRecipientRequest">
  <wsdl:part element="boxws:routeMessageToAbRecipient"
name="parameters"/>
</wsdl:message>
<wsdl:message name="routeMessageToAbRecipientResponse">
```

```

        <wsdl:part element="boxws:routeMessageToAbRecipientResponse"
name="parameters"/>
    </wsdl:message>
    <wsdl:message name="routeMessageToAbRecipient_faultMsg">
        <wsdl:part element="boxws:routeMessageToAbRecipient_fault"
name="parameters"/>
    </wsdl:message>
    <wsdl:message name="routeMessageToAbRecipient_fault1Msg">
        <wsdl:part element="boxws:loginFault" name="parameters"/>
    </wsdl:message>
    <wsdl:message name="switchContextPathRequest">
        <wsdl:part element="boxws:switchContextPath" name="parameters"/>
    </wsdl:message>
    <wsdl:message name="switchContextPathResponse">
        <wsdl:part element="boxws:switchContextPathResponse"
name="parameters"/>
    </wsdl:message>
    <wsdl:message name="switchContextPath_faultMsg">
        <wsdl:part element="boxws:switchContextPath_fault"
name="parameters"/>
    </wsdl:message>
    <wsdl:portType name="boxws">
        <wsdl:operation name="login">
            <wsdl:input message="boxws:loginRequest"/>
            <wsdl:output message="boxws:loginResponse"/>
            <wsdl:fault message="boxws:loginFault" name="fault"/>
        </wsdl:operation>
        <wsdl:operation name="logout">
            <wsdl:input message="boxws:logoutRequest"/>
            <wsdl:output message="boxws:logoutResponse"/>
        </wsdl:operation>
        <wsdl:operation name="routeMessageToAbRecipient">
            <wsdl:input message="boxws:routeMessageToAbRecipientRequest"/>
            <wsdl:output message="boxws:routeMessageToAbRecipientResponse"/>
            <wsdl:fault message="boxws:routeMessageToAbRecipient_faultMsg"
name="fault"/>
            <wsdl:fault message="boxws:routeMessageToAbRecipient_fault1Msg"
name="fault1"/>
        </wsdl:operation>
        <wsdl:operation name="switchContextPath">
            <wsdl:input message="boxws:switchContextPathRequest"/>
            <wsdl:output message="boxws:switchContextPathResponse"/>
            <wsdl:fault message="boxws:switchContextPath_faultMsg"
name="fault"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="boxwsSOAP" type="boxws:boxws">
        <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="login">
            <soap:operation
soapAction="http://box.intercope.com/boxws/login"/>
            <wsdl:input>
                <soap:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="fault">
                <soap:fault name="fault" use="literal"/>
            </wsdl:fault>
        </wsdl:operation>

```

```
<wsdl:operation name="logout">
  <soap:operation
soapAction="http://box.intercope.com/boxws/logout"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
</wsdl:operation>
<wsdl:operation name="routeMessageToAbRecipient">
  <soap:operation
soapAction="http://box.intercope.com/boxws/routeMessageToAbRecipient"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
  <wsdl:fault name="fault">
    <soap:fault name="fault" use="literal"/>
  </wsdl:fault>
  <wsdl:fault name="fault1">
    <soap:fault name="fault1" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
<wsdl:operation name="switchContextPath">
  <soap:operation
soapAction="http://box.intercope.com/boxws/switchContextPath"/>
  <wsdl:input>
    <soap:body use="literal"/>
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal"/>
  </wsdl:output>
  <wsdl:fault name="fault">
    <soap:fault name="fault" use="literal"/>
  </wsdl:fault>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="boxws">
  <wsdl:port binding="boxws:boxwsSOAP" name="boxwsSOAP">
    <!-- WAS 6 URL: .../<context>/services/<port-name> -->
    <!-- WAS 7 URL: .../<context>/<service-name> -->
    <soap:address
location="http://localhost:9080/BoxMessagingService"/>
    <!--
    <soap:address
location="http://localhost:9080/BoxMessagingService/boxws"/>
    -->
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

5 FIN Messaging Web Service (RJE formatted files)

5.1 Introduction

The FIN Message Web Service is designed to receive and transform RJE formatted files into MQ messages and consists of two request types. The first type is used to validate a file in the RJE format against a SWIFT FIN message definition set. This web service is independent from the BOX Messaging Hub installation and can be installed on a wide variety of Web Application Servers. The second service type splits the RJE file into single messages and writes these messages to a MQ queue read by BOX to be processed further.

5.2 Configuration

5.2.1 FINMessageService Configuration

The configuration for the `FINMessageService` is performed in two files, which are included in the WAR file “`finsvc.war`”.

- ◆ `FINMessageService.properties`
- ◆ `log4j2.xml`

Within `../WEB-INF/classes` the files `FINMessageService.properties` and `log4j2.xml` need to be adapted to suit the respective environment. Please also refer to the configuration guide `box_cfg_<Release>.pdf` for further information on the parameters of the file `FINMessageService.properties`.

5.2.1.1 FINMessageService.properties

This file is used to configure the MQ connection parameters.

5.2.1.1.1 Exemplary configuration

```
finsvc.replacement-file.resolve-via=path
finsvc.replacement-file.name=/opt/box/server/config/replace.rpl
finsvc.validation.standardrelease=2018
finsvc.rje.send.reference.maxLength=256
finsvc.failOnEmptyFile=true
finsvc.rje.send.reference.characterSet=UTF-8
finsvc.mq.env.hostname=localhost
finsvc.mq.env.port=1414
finsvc.mq.env.channel=SYSTEM.DEF.SVRCONN
finsvc.mq.env.userID=mqm
finsvc.mq.env.password=mqmadmin
finsvc.mq.queueManager=QMTEST
finsvc.mq.inputQueueName=BOX_RJE_IN
!finsvc.mq.cipherSuite=TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
!finsvc.mq.ccdt.url=file:///home/admdata/ccdt1.tab
!finsvc.mq.ccdt.url=file:///Z:/admdata/ccdt1.tab
!finsvc.mq.ccdt.url=ftp://ftp.server/admdata/ccdt1.tab
!finsvc.mq.fipsRequired=false
!-Djavax.net.ssl.trustStore=<path filename trustStore.jks>
!-Djavax.net.ssl.keyStore=<path filename keyStore.jks>
!-Djavax.net.ssl.keyStorePassword=<storePass>
```

If a cipher suite is used, the trustStore and keyStore related system properties may have to be specified:

```
-Djavax.net.ssl.trustStore=<path filename trustStore.jks>  
-Djavax.net.ssl.keyStore=<path filename keyStore.jks>  
-Djavax.net.ssl.keyStorePassword=<storePass>
```

5.2.2 Log4j.xml

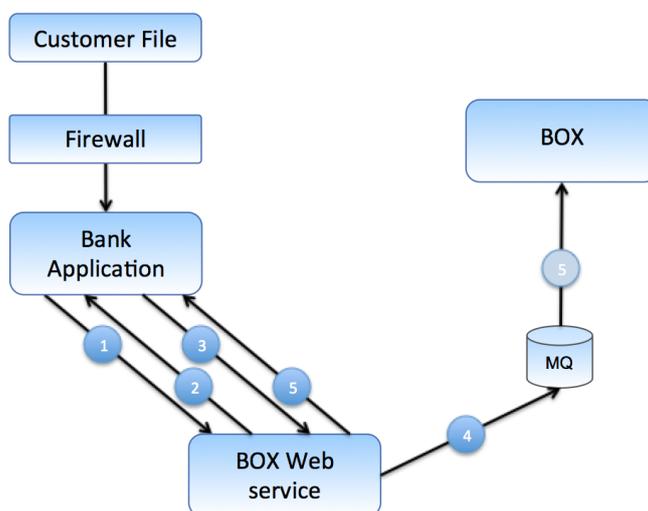
Please refer to chapter 1.4.2 for further configuration information and options.

5.3 Operations

After the successful deployment, the web service is available under the following context

➤ `http[s]://server:port/context_url/finsvc`

The following picture describes the workflow of an RJE file received and processed.



The customer is sending a file to a Bank Application. The Bank Application is converting the file to base64 and composing the requests.

5.3.1 validateRJE Operation

5.3.1.1 Request

Validate Request (rjeBase64)

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:fin="http://finsvc.ws.mpo.intercope.com">  
  <soapenv:Header/>  
  <soapenv:Body>  
    <fin:validateRJERequest>  
      <rjeBase64>ezE6RjAxQkNJVElUTU1YWfYM.....TovQw0KREVVVERFRkYNCi19</rjeBase64>  
    </fin:validateRJERequest>  
  </soapenv:Body>  
</soapenv:Envelope>
```

5.3.1.2 Response (all messages in RJE file correct)

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <a:validateRJEResponse
xmlns:a="http://finsvc.ws.mpo.intercope.com">
      <fileStatus>OK</fileStatus>
      <messageResults>
        <result msgNumber="1" status="OK"/>
        <result msgNumber="2" status="OK"/>
      </messageResults>
    </a:validateRJEResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

5.3.2 sendRJE Operation

5.3.2.1 Positive Response

If the response was positive (**<fileStatus>OK</fileStatus>**)
The second web service is invoked adding the reference:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:fi
  <soapenv:Header/>
  <soapenv:Body>
    <fin:sendRJERequest>
      <reference>DATEINAME_DER_ANGELIEFERTEN_DATEI</reference>
<rjeBase64>ezE6RjAxQkNJVElUTU1YWfhYMDAwMDAwMDAwMH17MjpJMjAyREVVVVERFRkZYWFhYTn17NDONC
MjE6SGVYyYmVydCBpc3QgaGllciB2aWVsIHp1IGxhbmcNCjozMkE6MTYwNjA2RVVSMsWNCjo1OEE6L0MNCkRF
  </fin:sendRJERequest>
  </soapenv:Body>
</soapenv:Envelope>
```

5.3.2.2 Response Status OK

All the messages are written to the MQ queue

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <a:sendRJEResponse xmlns:a="http://finsvc.ws.mpo.intercope.com">
      <status>OK</status>
    </a:sendRJEResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

5.3.2.3 MQ Message Processing

BOX is starting to read and process the messages.

5.3.2.4 Negative Response

Response (with invalid message in the RJE file):

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <a:validateRJEResponse
xmlns:a="http://finsvc.ws.mpo.intercope.com">
      <fileStatus>NOK</fileStatus>
      <messageResults>
        <result msgNumber="1" status="OK"/>
        <result msgNumber="2"
status="NOK">{1:F21BCITITMMXXX0000000000}{4:{177:1606161826}{451:1}{405
:T33002}}</result>
      </messageResults>
    </a:validateRJEResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

5.4 Documentation

Port type *FinMessageService* [Source code](#)

1. sendRJE

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: <http://finsvc.ws.mpo.intercope.com/sendRJE>

Input: sendRJERequest (soap:body, use = literal) [Source code](#)

```
sendRJERequest type sendRJERequest
  ■ reference type ReferenceType - type string with restriction maxLength(255)
    optional reference text
      RJE data either plain or base64 encoded
      ■ rjePlain type string
      ■ rjeBase64 type base64Binary
```

Output: sendRJEResponse (soap:body, use = literal) [Source code](#)

```
sendRJEResponse type sendRJEResponse
  ■ status type StatusType - type string with restriction - enum { 'OK', 'NOK' }
```

Fault: sendRJEFault (soap:fault, use = literal) [Source code](#)

```
sendRJEFault type sendRJEFault
  ■ sendRJEFault type string
```

2. validateRJE

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: <http://finsvc.ws.mpo.intercope.com/validateRJE>

Input: validateRJERequest (soap:body, use = literal) [Source code](#)

```
validateRJERequest type validateRJERequest
  ■ swiftStandardRelease - optional; type int
    4 digit year of SWIFT Standard Release, if omitted the configured year will be used
    RJE data either plain or base64 encoded
    ■ rjePlain type string
    ■ rjeBase64 type base64Binary
```

Output: validateRJEResponse (soap:body, use = literal) [Source code](#)

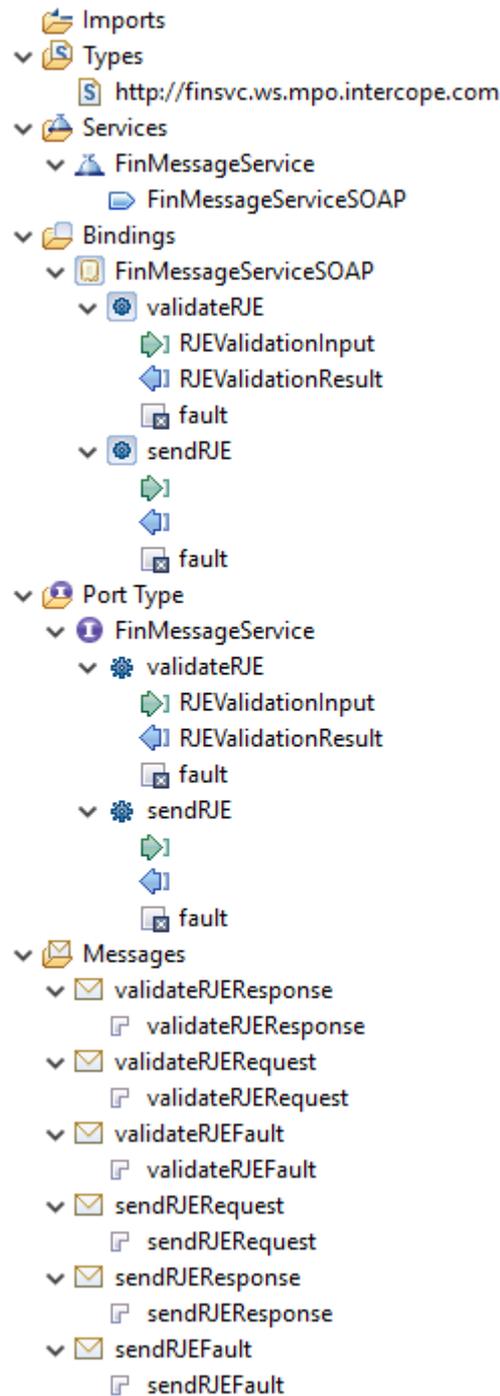
```
validateRJEResponse type validateRJEResponse
  ■ fileStatus type StatusType - type string with restriction - enum { 'OK', 'NOK' }
  ■ messageResults type RJEMessageResultType
    ■ result - unbounded; type ResultType
      o msgNumber type integer
      o status type StatusType - type string with restriction - enum { 'OK', 'NOK' }
```

Fault: validateRJEFault (soap:fault, use = literal) [Source code](#)

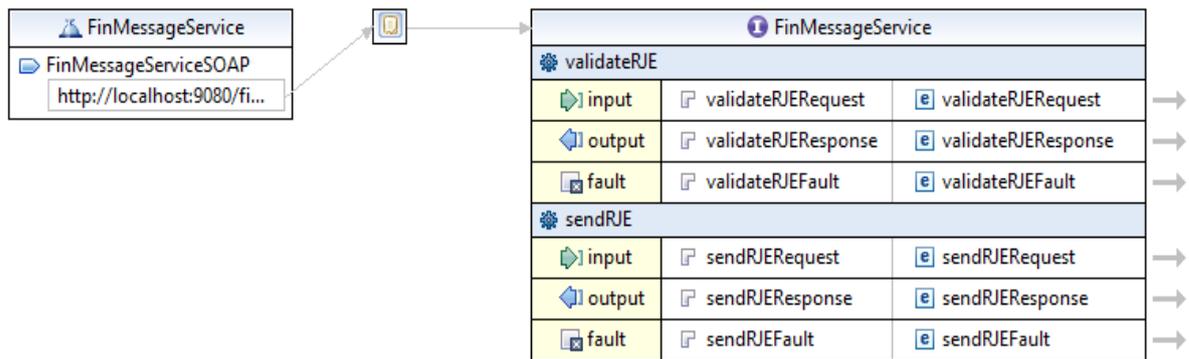
```
validateRJEFault type validateRJEFault
  ■ validateRJEFault type string
```

5.5 Programmatical Structure

5.5.1 Outline

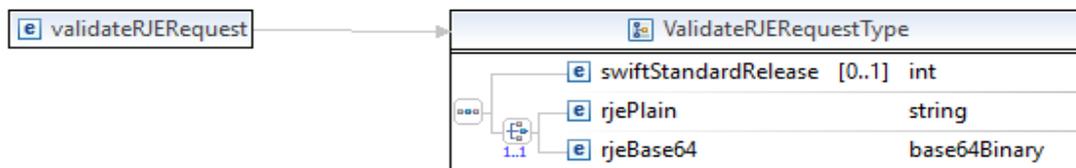


5.5.2 Generic Attributes



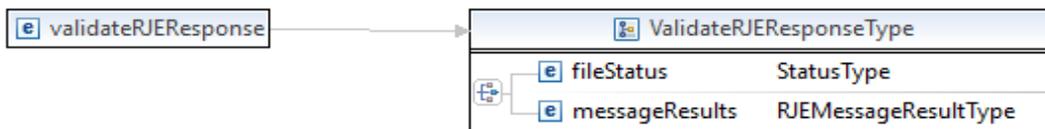
5.5.3 ValidateRJE Operation

5.5.3.1 Request



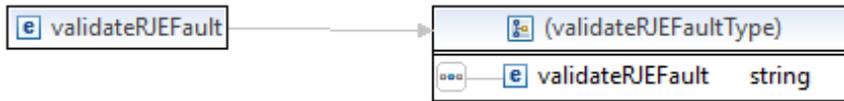
swiftStandardRelease	Integer, Current Swift Standard Release year
rjePlain	String, plain RJE string
rjeBase64	Base64Binary, Example ezE6RjAxQkNjVElUTU1YWfhYM.....TovQw0KREVVVERFRkYNCi19

5.5.3.2 Response



fileStatus	OK
messageResults	OK Example <messageResults> <result msgNumber="1" status="OK"/> <result msgNumber="2" status="OK"/> </messageResults>

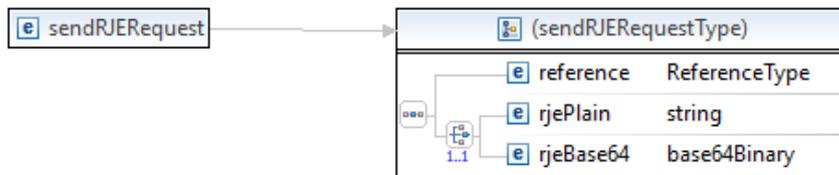
5.5.3.3 Fault



ValidateRJEFault	Error description string
------------------	--------------------------

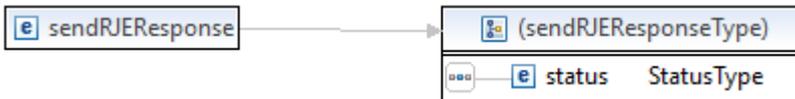
5.5.4 sendRJE

5.5.4.1 Request



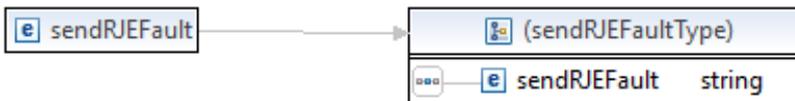
reference	Integer, Current Swift Standard Release year
rjePlain	String, plain RJE string
rjeBase64	Base64Binary, Example ezE6RjAxQkNJVElUTU1YWfhYM.....TovQw0KREVVVERFRkYNCi19

5.5.4.2 Response



status	type <i>string</i> with restriction - enum { 'OK', 'NOK' }
--------	--

5.5.4.3 Fault



sendRJEFault	Error description string
--------------	--------------------------

5.6 Appendix

5.6.1 Parameters

Parameter	Description												
<code>finsvc.replacement-file.resolve</code>	Specifies how to resolve the replacement file valid values: - <code>classpath</code> = use classpath to find replacement file - <code>path</code> = use path including filename If omitted no replacement file will be used												
<code>finsvc.replacement-file.name</code>	If the parameter <code>finsvc.replacement-file.resolve</code> is set to <code>path</code> , the path must be specified using this parameter.												
<code>finsvc.failOnEmptyFile=true</code>	This parameter controls the behavior of the processing of empty files. The default is true which results in NOK, if an empty file is validated. If set to false an empty file results in OK with no message result present.												
<code>finsvc.validation.standardrelease</code>	Specifies the default SWIFT message validation set to e.g. 2018												
<code>finsvc.rje.send.reference.maxLength</code>	This parameter specifies the length of the prefix added to single messages. It is used for transporting the original filename or another reference. Please note, this parameter must match the BOX definition!												
<code>finsvc.rje.send.reference.characterSet</code>	The character set used to convert input reference to MQ message bytes: <table border="0"> <tr> <td>US-ASCII</td> <td>Seven-bit ASCII, a.k.a. ISO646-US, a.k.a. the Basic Latin block of the Unicode character set</td> </tr> <tr> <td>ISO-8859-1</td> <td>ISO Latin Alphabet No. 1, a.k.a. ISO-LATIN-1</td> </tr> <tr> <td>UTF-8</td> <td>Eight-bit UCS Transformation Format</td> </tr> <tr> <td>UTF-16BE</td> <td>Sixteen-bit UCS Transformation Format, big-endian byte order</td> </tr> <tr> <td>UTF-16LE</td> <td>Sixteen-bit UCS Transformation Format, little-endian byte order</td> </tr> <tr> <td>UTF-16</td> <td>Sixteen-bit UCS Transformation Format, byte order identified by an optional byte-order mark</td> </tr> </table> Default is UTF-8	US-ASCII	Seven-bit ASCII, a.k.a. ISO646-US, a.k.a. the Basic Latin block of the Unicode character set	ISO-8859-1	ISO Latin Alphabet No. 1, a.k.a. ISO-LATIN-1	UTF-8	Eight-bit UCS Transformation Format	UTF-16BE	Sixteen-bit UCS Transformation Format, big-endian byte order	UTF-16LE	Sixteen-bit UCS Transformation Format, little-endian byte order	UTF-16	Sixteen-bit UCS Transformation Format, byte order identified by an optional byte-order mark
US-ASCII	Seven-bit ASCII, a.k.a. ISO646-US, a.k.a. the Basic Latin block of the Unicode character set												
ISO-8859-1	ISO Latin Alphabet No. 1, a.k.a. ISO-LATIN-1												
UTF-8	Eight-bit UCS Transformation Format												
UTF-16BE	Sixteen-bit UCS Transformation Format, big-endian byte order												
UTF-16LE	Sixteen-bit UCS Transformation Format, little-endian byte order												
UTF-16	Sixteen-bit UCS Transformation Format, byte order identified by an optional byte-order mark												
<code>finsvc.mq.env.userID</code>	If needed you can specify the MQ user ID												
<code>finsvc.mq.env.password</code>	If a user is specified, the password can be specified.												
<code>finsvc.mq.env.hostname</code>	IP address or DNS name of the MQ server												
<code>finsvc.mq.env.port</code>	Port number of the MQ server												
<code>finsvc.mq.env.channel</code>	Name of the channel used for the connection. Must be a channel of type SRVCONN												

Parameter	Description
<code>finsvc.mq.ccdt.url</code>	Location of the MQCHLTAB. If specified, the previous 3 parameters are ignored. Examples: <code>finsvc.mq.ccdt.url=file:///home/admdata/cc.tab</code> <code>finsvc.mq.ccdt.url=file:///Z:/admdata/cc.tab</code> <code>finsvc.mq.ccdt.url=ftp://ftp.server/admdata/cc.tab</code>
<code>finsvc.mq.connectqueuemanager</code>	If needed you can specify the connect queue manager
<code>finsvc.mq.queuemanager</code>	Name of the queue manager where the queue is defined
<code>finsvc.mq.inputqueuename</code>	Name of the queue where the messages are put
<code>finsvc.mq.cipherSuite</code>	Name of the cipher suite (e.g. TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA) Example (see https://www.ibm.com/support/knowledgecenter/SSFKSJ_8.0.0/com.ibm.mq.dev.doc/q113210_.htm) For this cipher suite you need to set system property: - <code>Dcom.ibm.mq.cfg.useIBMCipherMappings=false</code> and <code>fips</code> required to 'false'
<code>finsvc.mq.fipsRequired</code>	If the cipher suite requires fips you can set it to true

5.6.2 FIN Message wdsI

```

<wsdl:definitions name="FINMessageService"
targetNamespace="http://finsvc.ws.mpo.intercope.com"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://finsvc.ws.mpo.intercope.com"
xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <xsd:schema targetNamespace="http://finsvc.ws.mpo.intercope.com">
      <xsd:element name="validateRJEResponse"
type="tns:ValidateRJEResponseType"/>
      <xsd:element name="validateRJERequest"
type="tns:ValidateRJERequestType"/>
      <xsd:complexType name="ValidateRJERequestType">
        <xsd:sequence>
          <xsd:element name="swiftStandardRelease" type="xsd:int"
maxOccurs="1" minOccurs="0">
            <xsd:annotation>
              <xsd:documentation>4-digit year of SWIFT Standard Release,
if omitted the configured year will be used</xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:choice maxOccurs="1" minOccurs="1">
            <xsd:annotation>
              <xsd:documentation>RJE data either plain or base64
encoded</xsd:documentation>
            </xsd:annotation>
            <xsd:element name="rjePlain" type="xsd:string"/>
            <xsd:element name="rjeBase64" type="xsd:base64Binary"/>
          </xsd:choice>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:complexType name="ValidateRJEResponseType">
        <xsd:choice>
          <xsd:element name="fileStatus" type="tns:StatusType">
            <xsd:annotation>

```

```

        <xsd:documentation/>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="messageResults"
type="tns:RJEMessageResultType"/>
</xsd:choice>
</xsd:complexType>
<xsd:complexType name="RJEMessageResultType">
    <xsd:choice>
        <xsd:element name="result" type="tns:ResultType"
maxOccurs="unbounded" minOccurs="1"/>
    </xsd:choice>
</xsd:complexType>
<xsd:complexType name="ResultType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute name="msgNumber" type="xsd:integer"/>
            <xsd:attribute name="status" type="tns:StatusType"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
<xsd:simpleType name="StatusType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="OK"/>
        <xsd:enumeration value="NOK"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:element name="validateRJEFault">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="validateRJEFault" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="sendRJERequest">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="reference" type="tns:ReferenceType">
                <xsd:annotation>
                    <xsd:documentation>optional reference
text</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
            <xsd:choice maxOccurs="1" minOccurs="1">
                <xsd:annotation>
                    <xsd:documentation>RJE data either plain or base64
encoded</xsd:documentation>
                </xsd:annotation>
                <xsd:element name="rjePlain" type="xsd:string"/>
                <xsd:element name="rjeBase64" type="xsd:base64Binary"/>
            </xsd:choice>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:simpleType name="ReferenceType">
    <xsd:restriction base="xsd:string">
        <xsd:maxLength value="255"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:element name="sendRJEResponse">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="status" type="tns:StatusType"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="sendRJEFault">
    <xsd:complexType>
        <xsd:sequence>

```

```
        <xsd:element name="sendRJEFault" type="xsd:string"/>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
</wsdl:types>
<wsdl:message name="validateRJEResponse">
    <wsdl:part element="tns:validateRJEResponse" name="validateRJEResponse"/>
</wsdl:message>
<wsdl:message name="validateRJERequest">
    <wsdl:part element="tns:validateRJERequest" name="validateRJERequest"/>
</wsdl:message>
<wsdl:message name="validateRJEFault">
    <wsdl:part element="tns:validateRJEFault" name="validateRJEFault"/>
</wsdl:message>
<wsdl:message name="sendRJERequest">
    <wsdl:part name="sendRJERequest" element="tns:sendRJERequest"/>
</wsdl:message>
<wsdl:message name="sendRJEResponse">
    <wsdl:part name="sendRJEResponse" element="tns:sendRJEResponse"/>
</wsdl:message>
<wsdl:message name="sendRJEFault">
    <wsdl:part name="sendRJEFault" element="tns:sendRJEFault"/>
</wsdl:message>
<wsdl:portType name="FinMessageService">
    <wsdl:operation name="validateRJE">
        <wsdl:input message="tns:validateRJERequest"
name="RJEValidationInput"/>
        <wsdl:output message="tns:validateRJEResponse"
name="RJEValidationResult"/>
        <wsdl:fault name="fault" message="tns:validateRJEFault"/>
    </wsdl:operation>
    <wsdl:operation name="sendRJE">
        <wsdl:input message="tns:sendRJERequest"/>
        <wsdl:output message="tns:sendRJEResponse"/>
        <wsdl:fault name="fault" message="tns:sendRJEFault"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="FinMessageServiceSOAP" type="tns:FinMessageService">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="validateRJE">
        <soap:operation
soapAction="http://finsvc.ws.mpo.intercope.com/validateRJE"/>
        <wsdl:input name="RJEValidationInput">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="RJEValidationResult">
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="fault">
            <soap:fault use="literal" name="fault"/>
        </wsdl:fault>
    </wsdl:operation>
    <wsdl:operation name="sendRJE">
        <soap:operation
soapAction="http://finsvc.ws.mpo.intercope.com/sendRJE"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
        <wsdl:fault name="fault">
            <soap:fault use="literal" name="fault"/>
        </wsdl:fault>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="FinMessageService">
```

```
<wsdl:port binding="tns:FinMessageServiceSOAP"
name="FinMessageServiceSOAP">
  <soap:address location="http://192.168.50.73:8080/finsvc"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

6 FIN Message Validation Web Service

6.1 Introduction

BOX includes a module for a comprehensive validation of SWIFT FIN messages. This module is used internally by the application to validate messages which are received from external applications or generated by manual message entry.

In addition, the FIN message validation is available as a Web Service for any external application. This service can be used e.g. by financial applications which generate FIN messages to detect an invalid message structure or content at an early stage of message processing.

The FIN Message Validation Service is accessible as Web Service deployed in a Web Application Server.

Please note, this document assumes that you are familiar with the concepts of Web Services.

6.2 Configuration

The Web Service is delivered as a separate package in an ear file with the name `FINMessageValidation.war/ear`. This package contains the following main JAVA classes under `\WEB-INF\classes\com\intercope\mpo\rma\ws.:`

Before deploying the Service you have to perform the following configuration tasks:

- Customization of the WSDL-binding in `\wsdl\com\intercope\mpo\rma\ws\FinMessageValidation.wsdl`
- Configuration of the database connection in:

- `validation.properties`
- `log4j2.xml`

6.2.1 validation.properties

Customizing of the web service in `WEB-INF\classes\validation.properties`

The configuration file **validation.properties** is used to specify table names of the BIC directory, Currency directory and (optional) Country directory.

To connect to the BOX database you have the option of using either a JNDI Connection Pool (configured in your WAS) or a direct database connection, whereby we recommend to use a JNDI Connection Pool, which simplifies the configuration.

For the Connection Pool you just need to configure the JNDI Name:

```
DB.JNDIConnectionPool=<JNDI name>
```

For the direct database connection you need to configure the following parameters:

```
DB.qualifier=<the schema/owner/qualifier of the RMA tables>
```

As of Release 3.23.3 the following parameters have been removed:

```
DB.username=<username used to connect>
DB.password=<password used to connect>
DB.URL=<the JDBC URL, dependent on your RDBMS>
DB.driver=<the full qualified JDBC Driver class name>
```

Specify Standards MT Release:

```
year=<YYYY> e.g. 2019 for Standards 2019
```

Specify the table names of the BIC directory, Currency directory and Country directory:

- `table.bicdir=<table name of the BIC directory>` e.g. BICDOU1
- `table.curdir=<table name of the Currency directory>` e.g. CURDOU1
- `table.countrydir=<table name of the Country directory>` e.g. COUNTRYDIR

The tables must exist in the BOX Database. Usually they are configured and used for the validation in the GUI. (See **BOX Administration Guide, Appendix - BOX Data Maintenance** for details on SWIFT Reference Data)

The use of the Country directory is optional.

By setting **useBICDirectory=false** BIC lookup can be switched off, to avoid getting T28 results for BICs that are not contained in the BIC directory. In that case only the syntax check for the BIC is performed.

6.2.1.1 Exemplary Configuration

```
year=2018
DB.JNDIConnectionPool=MPO_JDBC_POOL
DB.qualifier=box
```

As of Release 3.23.3 the following parameters have been removed:

```
table.bicdir=bicdoul
table.curdir=curdoul
table.countrydir=countrydir
useBICDirectory=true
```

6.2.2 Log4j.xml

Please refer to chapter 1.4.2 for further configuration information and options.

6.3 Operation

The Web Service is accessible as “FINMessageValidation” under a URL which has to be configured before deployment.

6.3.1 FINMessageValidation Operation

6.3.1.1 Request

In the request, you have to specify:

```
<testmessage>
  \{1:F01PTSADES0XXXX0000000000\}{2:I103LRLRXXXX4A05N}{3:{108:MT103
015 OF 034}}{4:
:20:00071
:23B:SPAY
:32A:001118USD3,34
:33B:USD3,34
:50A:/34x
TESTUSVT
:52A:/C/34x
TESTUSVT
:53A:/D/34x
TESTUSVT
:54A:/D/34x
TESTUSVT
:55A:/D/34x
TESTUSVT
:56C://x
:57C:/34x
:59:/34x
4X35x
:70:/RFB/BET072///IPI/up to 20 chars
3X35x
:71A:BEN
:71F:USD1,34
:72:/RETN/12
/AC01/
/MREF/16x
/TEXT/29x
//33x
//33x
:77T:9000z
-}
</testmessage>
```

6.3.1.2 Negative Response

Result Number: 20038
Result Message: Semantic rule violated
UHB Code: E12
UHB Code description: MT 103 (not 103+).
Either field 70 or 77T, but not both, may be present.

If field 70 is present, then field 77T is not present.
If field 70 is not present, then field 77T is optional.

Character in Message: 293

Line (of Message): 20

Fin Validation Result:

Result Number: 20038

Result Message: Semantic rule violated

UHB Code: E12

UHB Code description: MT 103 (not 103+).

Either field 70 or 77T, but not both, may be present.

If field 70 is present, then field 77T is not present.

If field 70 is not present, then field 77T is optional.

Character in Message: 421

Line (of Message): 30

</result>

6.3.1.3 Positive Response

<result>

Fin Validation Result:

Result Number: 0

Result Message: Success.

</result>

The appendix lists the definition of the interface in the Web Service Description Language (WSDL).

6.4 Documentation

Web Service: FinMessageValidation

Target <http://validation.ws.rma.mpo.intercope.com>

Namespace:

*Port **FinMessageValidationSOAP** [Port type](#) [Source code](#)*

Location: <http://localhost:9080/FinMessageValidation/services/FinMessageValidationSOAP>

Protocol: SOAP

Default style: document

Transport SOAP over HTTP

protocol: **Operations:**

1. *validate* [Detail](#) [Source code](#)

Operations

*Port type **FinMessageValidation** [Source code](#)*

1. **validate**

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message

SOAP action: <http://validation.ws.rma.mpo.intercope.com/validate>

Input: validateRequest (soap:body, use = literal) [Source code](#)

validateRequest type *validateRequest* type *string*

Output: validateResponse (soap:body, use = literal) [Source code](#)

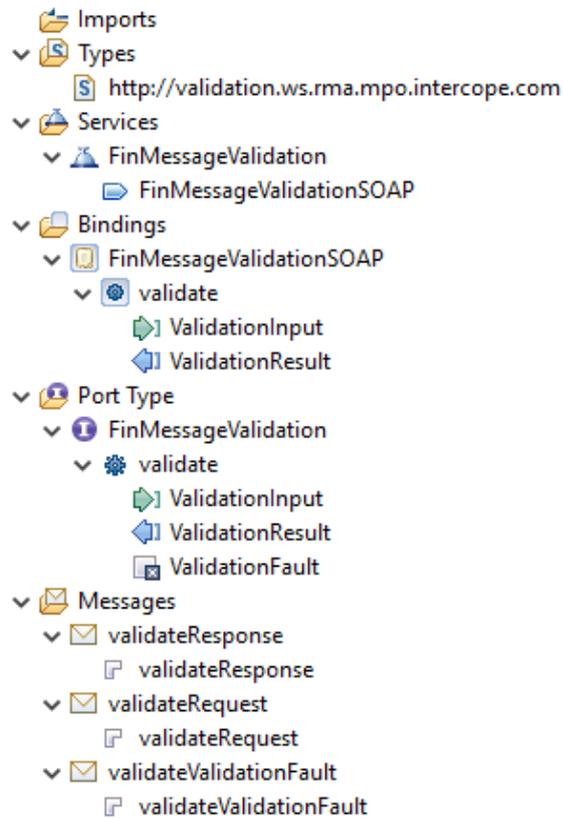
validateResponse type *validateResponse* type *string*

Fault: validateValidationFault [Source code](#)

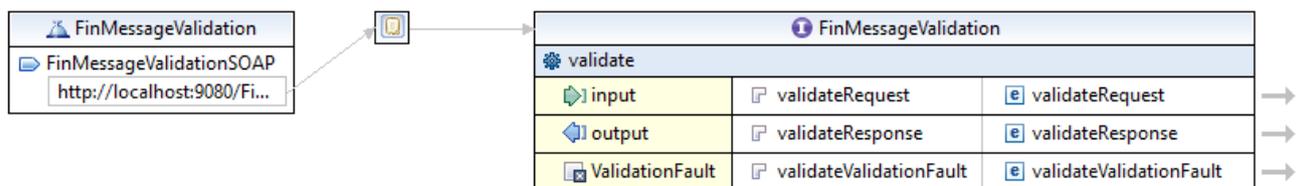
validateValidationFault type *validateValidationFault* type *string*

6.5 Programmatical Structure

6.5.1 Outline

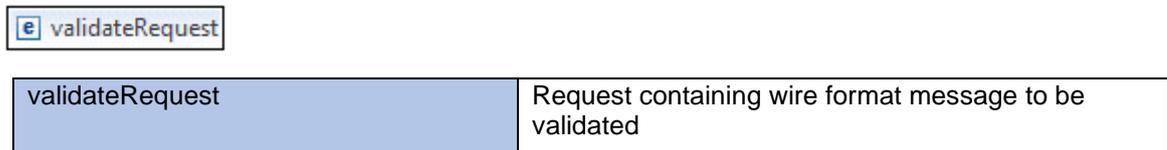


6.5.2 Generic Attributes



6.5.2.1 Validation

6.5.2.1.1 ValidateRequest



6.5.2.1.2 ValidateResponse

INTERCOPE

 validateResponse

validateResponse	Positive response : success Negative response : Semantic rule violated
------------------	--

6.5.2.1.3 Validation Fault

 validateValidationFault

validateValidationFault	Validation Fault Description
	<p>Example</p> <pre> Fin Validation Result:Result Number: 20032 Result Message: No end of text block indicator '&lt;CR>&lt;LF>-}' found, or non-trailer data was found after the text block 4. Block Number: 4 Line Number (Msg.): 0 Line Number (Block): 0 Position in Line: 1 Position in Block: 1 Position in Field: 99 UHB Code: V10 UHB Code description: ===== ===== End of block character '}'not found, or line, subfield or component greater than defined length ===== ===== </pre>

6.6 Appendix

6.6.1 FIN Message Validation wdsI

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://validation.ws.rma.mpo.intercope.com"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" name="FinMessageValidation"
targetNamespace="http://validation.ws.rma.mpo.intercope.com">
  <wsdl:types>
    <xsd:schema
targetNamespace="http://validation.ws.rma.mpo.intercope.com"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:element name="validateResponse" type="xsd:string"/>
      <xsd:element name="validateRequest" type="xsd:string"/>
      <xsd:element name="validateValidationFault"
type="xsd:string"></xsd:element>
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="validateResponse">
    <wsdl:part element="tns:validateResponse" name="validateResponse"/>
  </wsdl:message>
  <wsdl:message name="validateRequest">
    <wsdl:part element="tns:validateRequest" name="validateRequest"/>
  </wsdl:message>
  <wsdl:message name="validateValidationFault">
    <wsdl:part name="validateValidationFault"
element="tns:validateValidationFault">
  </wsdl:part>
  </wsdl:message>
  <wsdl:portType name="FinMessageValidation">
    <wsdl:operation name="validate">

```

INTERCOPE

```
        <wsdl:input message="tns:validateRequest" name="ValidationInput"/>
        <wsdl:output message="tns:validateResponse"
name="ValidationResult"/>
        <wsdl:fault name="ValidationFault"
            message="tns:validateValidationFault">
        </wsdl:fault>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="FinMessageValidationSOAP"
type="tns:FinMessageValidation">
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="validate">
        <soap:operation
soapAction="http://validation.ws.rma.mpo.intercope.com/validate"/>
        <wsdl:input name="ValidationInput">
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="ValidationResult">
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="FinMessageValidation">
    <wsdl:port binding="tns:FinMessageValidationSOAP"
name="FinMessageValidationSOAP">
        <soap:address
location="localhost:7001/FinMessageValidation/services/FinMessageValidat
ionSOAP"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

7 RMA Fast Permission Check Web Service

7.1 Introduction

Extending the scope of SWIFT specifications BOX-RMA provides a service, which allows external applications to verify the RMA status of any correspondent (RMA permission check service). Any application generating or handling FIN messages can check if a specific correspondent has authorized the own institution to send the message, which the application is generating. If the application detects that this authorization does not exist it can initiate an appropriate exception handling at an early processing stage. Otherwise the message would go down all the way to the CBT, the CBT would potentially reject to send the message and either send a corresponding notification back to the application or route the message to an exception queue within the CBT.

The RMA permission check service is accessible as Web Service deployed in a Web Application Server (the required files for this option can be found in the package under add-ons\web service and it is called RmaFastPermissionCheck - available separately for IBM WebSphere and BEA). However, the service can also be made available as application based on an IBM WebSphere MQ interface (the required files for this option can be found in the package under add-ons and it is called RMAService).

Please note, this document assumes that you are familiar with the basic concepts of a Relationship Management Application as defined by SWIFT. Please refer to the corresponding publications from SWIFT. In case the application runs on a z/OS (CICS or IMS) you can consider making use of the MERVA RMA Interface Format as described in the RMA admin guide, chapter 5.2.10.

7.2 Configuration

Important!

As of version 3 Release 23.2 the configuration file `databaseloader.properties` must be loaded via CLASSPATH, preferably in its full name (`databaseloader.properties`)! The configuration of `rmbserviceconfig.properties` must reflect this:

```
RecordProvider.implementation =
com.intercope.rma.provider.SimpleRecordProvider
RecordLoader.implementation = com.intercope.rma.loader.RmaDatabaseLoader
RecordLoader.config = databaseloader.properties

Service.count = 1
Service.1.implementation =
com.intercope.rma.service.retrieval.MervaMqPollingTask
Service.1.config = mqpollingtask.properties
```

The web service is delivered as a separate package in an ear file with the name "RmaFastPermissionCheck.ear/ RmaFastPermissionCheck.war". This package contains the following main JAVA classes under "WEB-INF\classes\com\intercope\mpo\rma\ws.":

- The Service Interface "IRmaFastPermissionCheck"
- The request class "CheckRMAPermissionRequest"
- The response class "CheckRMAPermissionResponse"

The following files are delivered and should be revisited:

- databaseloader.properties
- log4j2.xml
- rmaserviceconfig.properties

7.2.1 databaseloader.properties

In the file 'databaseloader.properties', the database connection has to be configured. This can be done by either setting a JNDI connection pool (configured in your WAS) or configuring a direct database connection.

Please note, Intercope recommends using a JNDI Connection Pool, which simplifies the configuration.

7.2.2 Connection Pool

For the Connection Pool you just need to configure the JNDI Name:

```
DatabaseLoader.JNDIConnectionPool=<JNDI name>
```

For the direct database connection you need to configure the following parameters:

```
DatabaseLoader.qualifier=<the schema/owner/qualifier of the RMA tables>
DatabaseLoader.username=<username used to connect>
DatabaseLoader.password=<password used to connect>
DatabaseLoader.url=<the JDBC URL, dependent on your RDBMS>
DatabaseLoader.jdbc.driver=<the full qualified JDBC Driver class name>
```

The parameter 'DatabaseLoader.JNDIConnectionPool' must be either commented out or removed from the configuration file.

7.2.3 RMA Framework

Customizing the RMA Service framework in "WEB-INF\classes\rmaserviceconfig.properties"

Parameters are set to

```
RecordProvider.implementation =
com.intercope.rma.provider.SimpleRecordProvider

RecordLoader.implementation = com.intercope.rma.loader.RmaDatabaseLoader
RecordLoader.config = databaseloader

Service.count = 0
```

Important

These parameters should not be changed

7.2.3.1 Configuration of the RMA Web Application

7.2.3.1.1 rmaserviceconfig.properties

`Logger.filename = rmaservice.%u.log`

This parameter specifies the name (and path) of the log file created for the service. The format is defined by the `java.util.logging.FileHandler` class.

`RecordLoader.config = databaseloader`

This parameter defines the configuration file used for the database connection.

`Service.1.config = mqpollingtask`

This parameter defines the configuration file used for the MQ connection (the description of this configuration file can be found in 5.1.3.3).

7.2.3.1.2 databaseloader.properties

The `DatabaseLoader` implementation can make use of an externally defined JNDI connection pool or of its own implementation.

Please note, INTERCOPE recommends the use of an externally provided JNDI connection pool.

`DatabaseLoader.qualifier=rmaowner`

This parameter is used in either configuration. Enter the qualifier of the RMA tables.

`DatabaseLoader.JNDIConnectionPool=RMA_CONN_POOL`

Enter the JNDI name of the connection pool. If this parameter is set the remaining parameters are ignored.

`DatabaseLoader.username=rmaowner`

Enter the name of the database owner; this user must have full access right to the database

`DatabaseLoader.password=password`

Enter the password for the user configured in the previous parameter

`DatabaseLoader.jdbc.driver=oracle.jdbc.driver.OracleDriver`

Enter the name of the JDBC driver to use.:

`oracle.jdbc.driver.OracleDriver` for Oracle

or

`com.ibm.db2.jcc.DB2Driver` for DB2 Universal Driver

`DatabaseLoader.url=jdbc:oracle:thin:@192.168.7.1:1521:rmadb`

Enter the URL to connect to the database:

ORACLE:

`jdbc:oracle:thin:@192.168.7.1:1521:rmadb`

DB2:

a) `jdbc:db2://<ip address>:<port>/<db/location name>`
for the Type 4 Driver

b) `jdbc:db2:<db/location name>`
for a Type 2 Driver

Please make sure the environment for the user that starts the application is configured correctly, e.g.: `LIBPATH`, `CLASSPATH` (`STEPLIB` on host).

7.2.3.1.3 Log4j.xml

Please refer to chapter 1.4.2 for further configuration information and options.

7.2.3.1.4 Interface Format

This section describes the interface format between the RMA Permission Check and RMA Permission Check Service in BOX.

Please note, this format can also be used from applications other than to obtain information from the RMA datastore.

The RMA Permission Check composes a query for a pair of BIC8s (own BIC and correspondent BIC).

The format of the query is:

```
"ICOPE RMA INQUIRY: PTSADESSABCDEFF"  
12345678901234567890123456789012345678 (character positions)
```

Description:

PTSADESS: own BIC8

ABCDEFF: correspondent BIC8

This record is written to a local MQ queue as a datagram. The BOX RMA Service reads this queue and returns a response message to another MQ queue in the following format.

Sample response:

```
ICOPE RMA INQUIRY: PTSADESSABCDEFF Issued Enabled Position  
1234567890123456789012345678901234567890123 1- 53  
  
PTSADESSABCDEFFswift.fin!p 2006-08-01 16:58:58  
4567890123456789012345678901234567890123456789 54-109  
  
2006-12-312007-12-31ALL;I200,I202;;X498;;;+ReceivedEnabled  
0123456789012345678901234567890123456789012345670 110-170  
  
PTSADESSABCDEFFswift.finÛp 2006-08-01 16:58:58  
12345678901234567890123456789012345678901234567 171-226  
  
2006-12-312007-12-31ALL;200,202;;X498;;;.  
7890123456789012345678901234567890123456789 227-269
```

The record is built similar to the format on the RMA datastore.

Explanation of the above example:

The length is variable. The Issued and Receives records are separated by "+" (plus sign).

The response consists out of the query itself: ICOPE RMA INQUIRY:, the Issued and the Received records, each containing the own BIC: PTSADESSA and the correspondent BIC: ABCADEFF, followed by the status: Enabled, the Business service: swift.fin!p, the Creation Date/Time: 2006-08-01 16:58:58, the Validity period: from 2006-12-12 to 2007-12-31, then entries for all categories 1 to 8, separated by ";" (semicolon) in the following format:

```
ALL;I200,I202;;X498;;;
```

Description:

A preceding "I" means included; "X" means excluded from permissions. In category 1 all MTs are permitted (keyword ALL), In category 2 only MTs 200 and 202 are permitted. No specification for

category 3 (=not permitted). In category 4 all MTs except MT 498 are permitted. Categories 5, 6, 7, and 8 are not specified (= not permitted).

In case the requested pair of BIC8s is not in the RMA datastore, an empty response (only "ICOPE RMA INQUIRY: PTSADNESSABCADEFF") is returned.

In case there is another problem on the RMA Server, the following error message is returned:

```
"ICOPE RMA INQUIRY: PTSADNESSABCADEFF ERROR
```

```
An Error occurred during RMA record retrieval".
```

7.3 Operations

The Web service is accessible as "RmaFastPermissionCheck" under a URL which has to be configured before deployment.

7.3.1 checkNonFINRMAPermissionRequest Operation

7.3.1.1 Request

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:rmaf="http://ws.rma.mpo.intercope.com/RmaFastPermissionCheck/">
  <soapenv:Header/>
  <soapenv:Body>
    <rmaf:checkNonFINRMAPermissionRequest>
      <ownBIC>PTSADESB</ownBIC>
      <correspondentBIC>PTSADESS</correspondentBIC>
      <rmaBusinessService>swift.fin!x</rmaBusinessService>
      <rmaRequestType>LIVE</rmaRequestType>
      <rmaDirection>ISSUED</rmaDirection>
    </rmaf:checkNonFINRMAPermissionRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

7.3.1.2 Response

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns0:checkNonFINRMAPermissionResponse
xmlns:ns0="http://ws.rma.mpo.intercope.com/RmaFastPermissionCheck/">
      <result>>true</result>
    </ns0:checkNonFINRMAPermissionResponse>
  </S:Body>
</S:Envelope>
```

7.3.1.3 Fault

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns0:checkNonFINRMAPermissionResponse
xmlns:ns0="http://ws.rma.mpo.intercope.com/RmaFastPermissionCheck/">
      <result>>false</result>
    </ns0:checkNonFINRMAPermissionResponse>
  </S:Body>
</S:Envelope>
```

7.3.2 checkRMAPermissionRequest Operation

7.3.2.1 Request

```
soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:rmaf="http://ws.rma.mpo.intercope.com/RmaFastPermissionCheck/">
  <soapenv:Header/>
  <soapenv:Body>
    <rmaf:checkRMAPermissionRequest>
      <ownBIC>PTSADESB</ownBIC>
      <correspondentBIC>PTSADESB</correspondentBIC>
      <rmaBusinessService>swift.fin!x</rmaBusinessService>
      <finMessageType>103</finMessageType>
      <rmaDirection>ISSUED</rmaDirection>
    </rmaf:checkRMAPermissionRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

7.3.2.2 Response

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:checkRMAPermissionResponse
xmlns:ns2="http://ws.rma.mpo.intercope.com/RmaFastPermissionCheck/">
      <result>>true</result>
    </ns2:checkRMAPermissionResponse>
  </S:Body>
</S:Envelope>
```

7.3.2.3 Fault

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:checkRMAPermissionResponse
xmlns:ns2="http://ws.rma.mpo.intercope.com/RmaFastPermissionCheck/">
      <result>>false</result>
    </ns2:checkRMAPermissionResponse>
  </S:Body>
</S:Envelope>
```

7.4 Documentation

Web Service: RmaFastPermissionCheckService

Target: <http://ws.rma.mpo.intercope.com/RmaFastPermissionCheck/>

Namespace:

Port *RmaFastPermissionCheckSOAPPort* [Port type](#) [Source code](#)

Location: <http://localhost/RMAFastPermissionCheck/RMAPermissionCheck>

Protocol: SOAP

Default style: document

Transport SOAP over HTTP

protocol:

Operations:

1. [checkNonFINRMAPermission](#) [Detail](#) [Source code](#)
2. [checkRMAPermission](#) [Detail](#) [Source code](#)

Operations

Port type *IRmaFastPermissionCheck* [Source code](#)

1. checkNonFINRMAPermission

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: <http://ws.rma.mpo.intercope.com/RmaFastPermissionCheck/checkNonFINRMAPermission>

Input: checkNonFINRMAPermissionRequest (soap:body, use = literal) [Source code](#)

```
checkNonFINRMAPermissionRequest type checkNonFINRMAPermissionRequest
  • ownBIC type RMA_BIC - type string with restriction length(8)
  • correspondentBIC type RMA_BIC - type string with restriction length(8)
  • rmaBusinessService type RMA_SERVICE - type string with restriction pattern([a-zA-Z0-9.!]+)
  • rmaRequestType type RMA_REQUESTTYPE - type string with restriction pattern([a-zA-Z0-9.!]+)
  • rmaDirection type RMA_DIRECTION - type string with restriction - enum { 'ISSUED', 'RECEIVED' }
```

Output: checkNonFINRMAPermissionResponse (soap:body, use = literal) [Source code](#)

```
checkNonFINRMAPermissionResponse type checkNonFINRMAPermissionResponse
  • result type boolean
```

Fault: checkNonFINRMAPermission_faultMsg (soap:fault, use = literal) [Source code](#)

```
checkNonFINRMAPermissionFaultMessage type checkNonFINRMAPermission_fault
  • checkNonFINRMAPermission_fault type string
```

2. checkRMAPermission

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: <http://ws.rma.mpo.intercope.com/RmaFastPermissionCheck/checkRMAPermission>

Input: checkRMAPermissionRequestMessage (soap:body, use = literal) [Source code](#)

```
checkRMAPermissionRequest type checkRMAPermissionRequest
  • ownBIC type RMA_BIC - type string with restriction length(8)
  • correspondentBIC type RMA_BIC - type string with restriction length(8)
  • rmaBusinessService type RMA_SERVICE - type string with restriction pattern([a-zA-Z0-9.!]+)
  • finMessageType type RMA_MT - type string with restriction length(3)
  • rmaDirection type RMA_DIRECTION - type string with restriction - enum { 'ISSUED', 'RECEIVED' }
```

Output: checkRMAPermissionResponseMessage (soap:body, use = literal) [Source code](#)

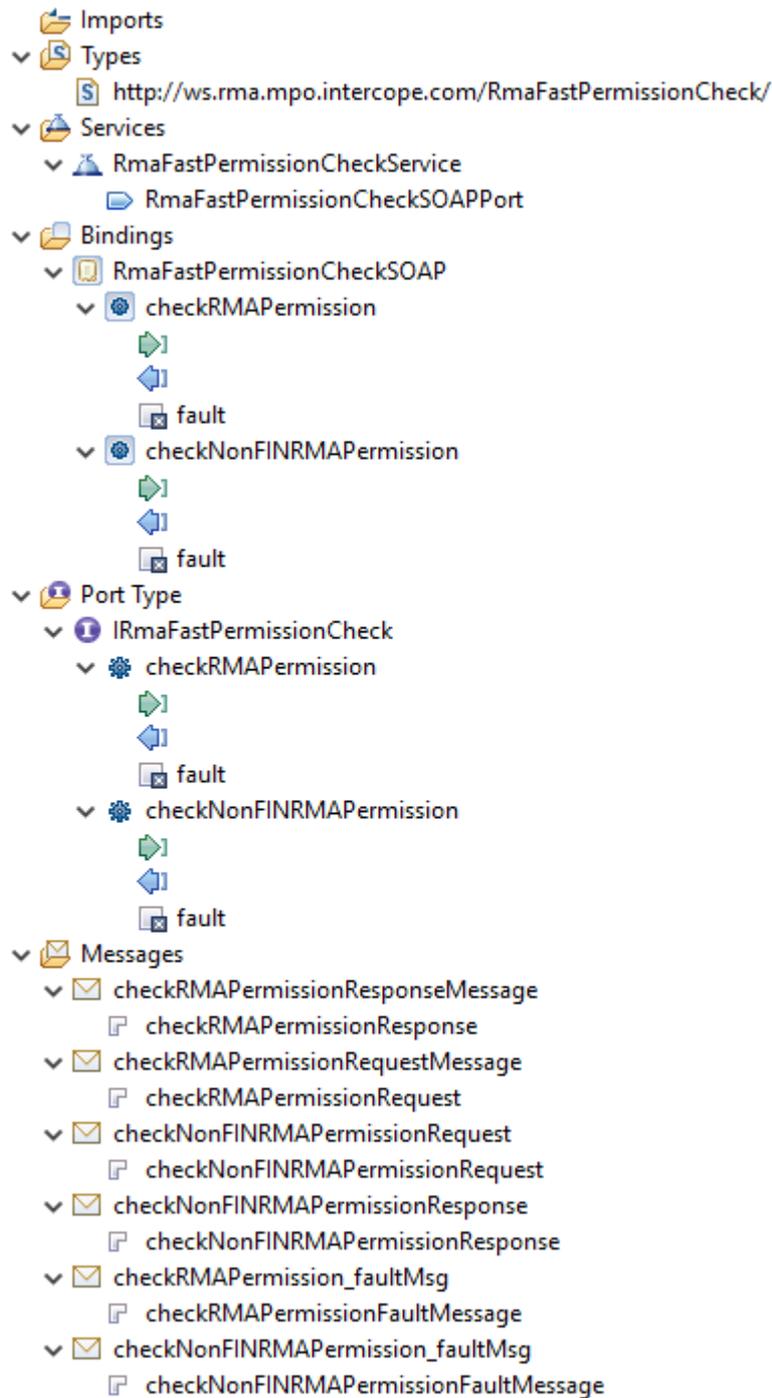
```
checkRMAPermissionResponse type checkRMAPermissionResponse
  • result type boolean
```

Fault: checkRMAPermission_faultMsg (soap:fault, use = literal) [Source code](#)

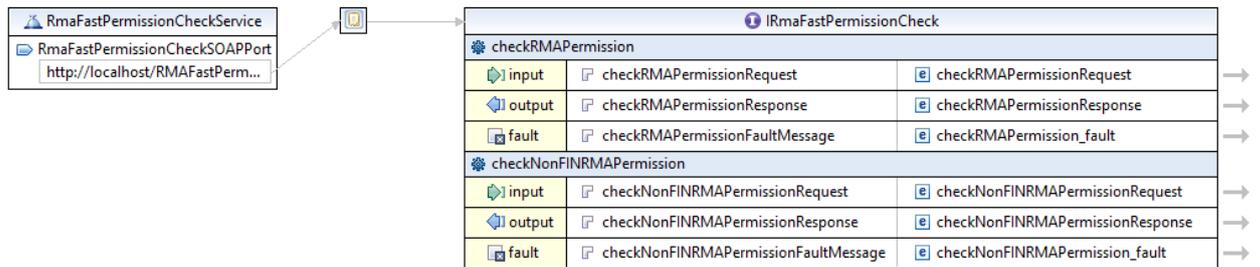
```
checkRMAPermissionFaultMessage type checkRMAPermission_fault
  • checkRMAPermission_fault type string
```

7.5 Programmatical Structure

7.5.1 Outline

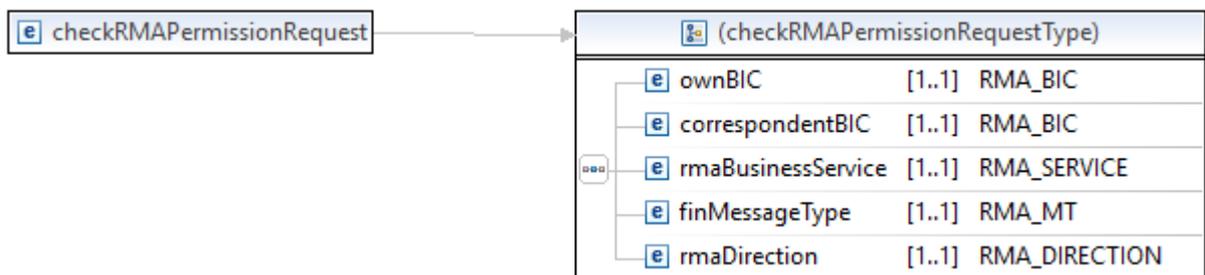


7.5.2 Generic Attributes



7.5.2.1 checkRMAPermission

7.5.2.1.1 Request



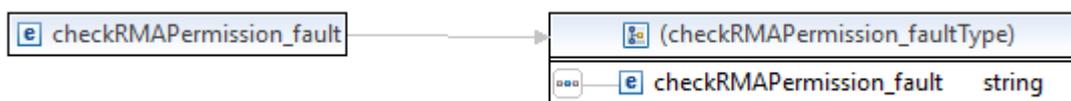
ownBIC	The BIC8 of own institution
correspondentBIC	The BIC8 of the correspondent
rmaBusinessService	The RMA Service for which you want to check the RMA authorization status. Specify one of the following options: LIVE PILOT
finMessageType	FIN MT
rmaDirection	The direction of the message traffic. Specify one of the following options: ISSUED (when you want to check if you allowed the correspondent to send messages to you) RECEIVED (when you want to check if you are allowed to send the message to the correspondent)

7.5.2.1.2 Response



result	<p>In the response the result of the inquiry is stored in the element "result" as Boolean value.</p> <p>TRUE (1) indicates that an authorization exists, FALSE (0) indicates that no authorization exists.</p> <p>Appendix 1 lists the definition of the interface in the Web Service Description Language (WSDL)</p>
--------	---

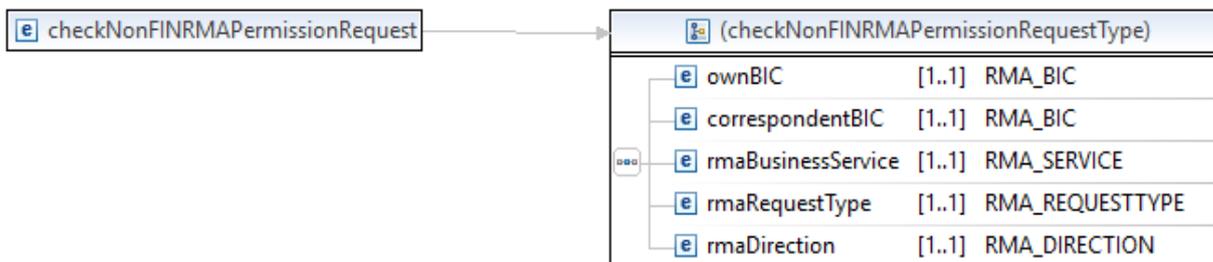
7.5.2.1.3 Fault



checkRMAPPermissionFaultMessage	Error message 'false'
---------------------------------	-----------------------

7.5.2.2 checkNonFINRMAPPermission

7.5.2.2.1 Request

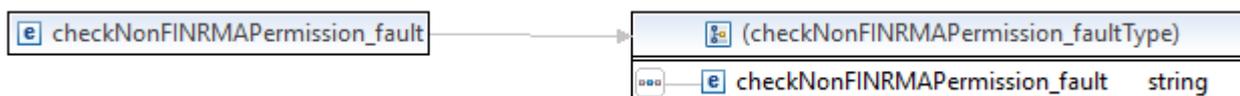


7.5.2.2.2 Response



checkNonFINRMAPPermissionResponse	Response type 'true'
-----------------------------------	----------------------

7.5.2.2.3 Fault



checkNonFINRMAPPermission_fault	Error message 'false'
---------------------------------	-----------------------

7.6 Appendix

7.6.1 RMA Fast Permission Check Web Service wdsI

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="RmaFastPermissionCheck"
  targetNamespace="http://ws.rma.mpo.intercope.com/RmaFastPermissionCheck/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://ws.rma.mpo.intercope.com/RmaFastPermissionCheck/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <xsd:schema
      targetNamespace="http://ws.rma.mpo.intercope.com/RmaFastPermissionCheck/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:tns="http://ws.rma.mpo.intercope.com/RmaFastPermissionCheck/">
      <xsd:element name="checkRMAPermissionResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="result" type="xsd:boolean"
              minOccurs="1" maxOccurs="1">
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="checkNonFINRMAPermissionResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="result" type="xsd:boolean"
              minOccurs="1" maxOccurs="1">
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:simpleType name="RMA_DIRECTION">
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="ISSUED" />
          <xsd:enumeration value="RECEIVED" />
        </xsd:restriction>
      </xsd:simpleType>
      <xsd:simpleType name="RMA_SERVICE">
        <xsd:restriction base="xsd:string">
          <xsd:pattern value="[a-zA-Z0-9.!]+" />
        </xsd:restriction>
      </xsd:simpleType>
      <xsd:simpleType name="RMA_BIC">
        <xsd:restriction base="xsd:string">
          <xsd:length value="8" />
        </xsd:restriction>
      </xsd:simpleType>
      <xsd:simpleType name="RMA_MT">
        <xsd:restriction base="xsd:string">
          <xsd:length value="3" />
        </xsd:restriction>
      </xsd:simpleType>
      <xsd:simpleType name="RMA_REQUESTTYPE">
        <xsd:restriction base="xsd:string">
          <xsd:pattern value="[a-zA-Z0-9.!]+" />
          <!--
            <xsd:minLength value="1"></xsd:minLength>
          -->
        </xsd:restriction>
      </xsd:simpleType>
      <xsd:element name="checkRMAPermissionRequest">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="ownBIC" type="tns:RMA_BIC"
              minOccurs="1" maxOccurs="1">

```

```
</xsd:element>
<xsd:element name="correspondentBIC"
  type="tns:RMA_BIC" minOccurs="1" maxOccurs="1">
</xsd:element>
<xsd:element name="rmaBusinessService"
  type="tns:RMA_SERVICE" minOccurs="1" maxOccurs="1">
</xsd:element>
<xsd:element name="finMessageType"
  type="tns:RMA_MT" minOccurs="1" maxOccurs="1">
</xsd:element>
<xsd:element name="rmaDirection"
  type="tns:RMA_DIRECTION" minOccurs="1" maxOccurs="1">
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="checkNonFINRMAPermissionRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="ownBIC" type="tns:RMA_BIC"
        minOccurs="1" maxOccurs="1">
      </xsd:element>
      <xsd:element name="correspondentBIC"
        type="tns:RMA_BIC" minOccurs="1" maxOccurs="1">
      </xsd:element>
      <xsd:element name="rmaBusinessService"
        type="tns:RMA_SERVICE" minOccurs="1" maxOccurs="1">
      </xsd:element>
      <xsd:element name="rmaRequestType"
        type="tns:RMA_REQUESTTYPE" minOccurs="1" maxOccurs="1">
      </xsd:element>
      <xsd:element name="rmaDirection"
        type="tns:RMA_DIRECTION" minOccurs="1" maxOccurs="1">
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="checkRMAPermission_fault">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="checkRMAPermission_fault"
        type="xsd:string">
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="checkNonFINRMAPermission_fault">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element
        name="checkNonFINRMAPermission_fault" type="xsd:string">
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
</xsd:schema>
</wsdl:types>
<wsdl:message name="checkRMAPermissionResponseMessage">
  <wsdl:part element="tns:checkRMAPermissionResponse"
    name="checkRMAPermissionResponse" />
</wsdl:message>
<wsdl:message name="checkRMAPermissionRequestMessage">
  <wsdl:part element="tns:checkRMAPermissionRequest"
    name="checkRMAPermissionRequest" />
</wsdl:message>
<wsdl:message name="checkNonFINRMAPermissionRequest">
  <wsdl:part name="checkNonFINRMAPermissionRequest"
    element="tns:checkNonFINRMAPermissionRequest"></wsdl:part>
</wsdl:message>
```

```

    <wsdl:message name="checkNonFINRMAPermissionResponse">
      <wsdl:part name="checkNonFINRMAPermissionResponse"
element="tns:checkNonFINRMAPermissionResponse"></wsdl:part>
    </wsdl:message>
    <wsdl:message name="checkRMAPermission_faultMsg">
      <wsdl:part name="checkRMAPermissionFaultMessage"
element="tns:checkRMAPermission_fault"></wsdl:part>
    </wsdl:message>
    <wsdl:message name="checkNonFINRMAPermission_faultMsg">
      <wsdl:part name="checkNonFINRMAPermissionFaultMessage"
element="tns:checkNonFINRMAPermission_fault"></wsdl:part>
    </wsdl:message>
    <wsdl:portType name="IRmaFastPermissionCheck">
      <wsdl:operation name="checkRMAPermission">
        <wsdl:input message="tns:checkRMAPermissionRequestMessage" />
        <wsdl:output
          message="tns:checkRMAPermissionResponseMessage" />
          <wsdl:fault name="fault"
message="tns:checkRMAPermission_faultMsg"></wsdl:fault>
        </wsdl:operation>
      <wsdl:operation name="checkNonFINRMAPermission">
        <wsdl:input message="tns:checkNonFINRMAPermissionRequest"></wsdl:input>
        <wsdl:output
message="tns:checkNonFINRMAPermissionResponse"></wsdl:output>
          <wsdl:fault name="fault"
message="tns:checkNonFINRMAPermission_faultMsg"></wsdl:fault>
        </wsdl:operation>
      </wsdl:portType>
    <wsdl:binding name="RmaFastPermissionCheckSOAP"
type="tns:IRmaFastPermissionCheck">
      <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http" />
      <wsdl:operation name="checkRMAPermission">
        <soap:operation
SoapAction="http://ws.rma.mpo.intercope.com/RmaFastPermissionCheck/checkRMAPermi
ssion" />
        <wsdl:input>
          <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
        <wsdl:fault name="fault">
          <soap:fault use="literal" name="fault" />
        </wsdl:fault>
      </wsdl:operation>
      <wsdl:operation name="checkNonFINRMAPermission">
        <soap:operation
soapAction="http://ws.rma.mpo.intercope.com/RmaFastPermissionCheck/checkNonFINRM
APermission" />
        <wsdl:input>
          <soap:body use="literal" />
        </wsdl:input>
        <wsdl:output>
          <soap:body use="literal" />
        </wsdl:output>
        <wsdl:fault name="fault">
          <soap:fault use="literal" name="fault" />
        </wsdl:fault>
      </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="RmaFastPermissionCheckService">
      <wsdl:port binding="tns:RmaFastPermissionCheckSOAP"
name="RmaFastPermissionCheckSOAPPort">
        <soap:address
location="http://localhost/RMAFastPermissionCheck/RMAPermissionCheck" />
      </wsdl:port>
    </wsdl:service>
  </wsdl:definitions>

```

8 Target2 Securities Signature Validation Web Service

8.1 Introduction

BOX includes a module for a comprehensive validation of Target2 Securities (T2S) Signatures.

The T2S Signature validation Web Service is used to validate a business signature (or a batch of signatures) provided by Alliance Access to the Alliance Gateway. The signature is generated and needed for the end-to-end digital signature exchange with the T2S service. The provided W3C Signature element contains the digest and references to the data that needs to be signed.

The respective signature can be retrieved by the calling application and stored in its T2S message header.

Please note, this document assumes, that you are familiar with the concepts of Web Services.

8.2 Configuration

The Web Service is delivered as a separate package in a war file with the name t2sws.war. This package contains the following main JAVA classes under \WEB-INF\lib

- configuration.properties
- T2SWebService.properties
- t2sws.wsdl

Before deploying the Service, the following configuration tasks must be performed:

To connect to the BOX database, you have the option of using either a JNDI Connection Pool (configured in your WAS) or a direct database connection, whereby we recommend using a JNDI Connection Pool, which simplifies the configuration.

For the Connection Pool you just need to configure the JNDI Name:

```
DB.JNDIConnectionPool=<JNDI name>
```

For the direct database connection, the following parameters must be configured:

```
DB.qualifier=<the schema/owner/qualifier of the RMA tables>
DB.username=<username used to connect>
DB.password=<password used to connect>
DB.URL=<the JDBC URL, dependent on your RDBMS>
DB.driver=<the full qualified JDBC Driver class name>
```

The configuration file **T2SWebService.properties** is used to specify the MQ connection details.

```
t2sws.mq.env.hostname=localhost
t2sws.mq.env.port=1414
t2sws.mq.env.channel=SYSTEM.DEF.SVRCONN
t2sws.mq.si.queuemanager=QM_BOX
t2sws.mq.si.inputqueuename=T2SWS_SI_IN
t2sws.mq.si.replytoqueuename=T2SWS_SI_OUT
```

Please note, any change of configuration requires the signing of the respective file. Please refer to the [Security_Implementation_Guide_v3r22.pdf](#) for precise information on how to do this.

8.3 Operations

The Web Service is accessible as “T2S Signature Web Services” under a URL which has to be configured before deployment. The service calculates the checksum for T2S Messages and signs these.

8.4 Documentation

Web Service: t2sws

Target <http://box.intercope.com/t2sws/>
Namespace:
*Port **t2swsSOAP** [Port type](#) [Source code](#)*
Location: <https://localhost/t2sws>
Protocol: SOAP
Default style: document
Transport protocol: SOAP over HTTP
Operations:
 1. [computeSignature](#) [Detail](#) [Source code](#)
 2. [login](#) [Detail](#) [Source code](#)
 3. [logout](#) [Detail](#) [Source code](#)
 4. [validateSignature](#) [Detail](#) [Source code](#)

Operations

*Port type **t2sws** [Source code](#)*

1. computeSignature

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: <http://box.intercope.com/t2sws/computeSignature>

Input: computeSignatureRequest (soap:body, use = literal) [Source code](#)

parameters type *computeSignature*

- SessionId type *SessionIdType* - type *string* with restriction
- ServiceProvider type *ServiceProviderType* - type *string* with restriction - enum { 'SIA-Colt' }
- Signer type *string*
- SignatureData type *CDATA* - type *string* with restriction
holds the complete ds:Signature element (without ds:SignatureValue, ds:KeyInfo, the ds:Reference to it) embedded in a CDATA section

Output: computeSignatureResponse (soap:body, use = literal) [Source code](#)

parameters type *computeSignatureResponse*

- SignatureData type *CDATA* - type *string* with restriction
holds the complete ds:Signature element embedded in a CDATA section

Fault: computeSignatureFault (soap:fault, use = literal) [Source code](#)

parameters type *computeSignatureFault*

- Error type *string*
- Description type *string*

2. login

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://box.intercope.com/t2sws/login

Input: loginRequest (soap:body, use = literal) [Source code](#)

```
parameters type login
  ClientPrefix type string
  UserName type string
  Secret type SecretType
    plain type string
    plain text password
    encrypted type string
    encrypted password
```

Output: loginResponse (soap:body, use = literal) [Source code](#)

```
parameters type loginResponse
  SessionId type SessionIdType - type string with restriction
```

Fault: loginFault (soap:fault, use = literal) [Source code](#)

```
parameters type loginFault
  LoginFault type string
```

3. logout

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://box.intercope.com/t2sws/logout

Input: logoutRequest (soap:body, use = literal) [Source code](#)

```
parameters type logout
  SessionId type SessionIdType - type string with restriction
```

Output: logoutResponse (soap:body, use = literal) [Source code](#)

```
parameters type logoutResponse
  Success type boolean
```

4. validateSignature

[Source code](#)

Operation type: *Request-response*. The endpoint receives a message, and sends a correlated message.

SOAP action: http://box.intercope.com/t2sws/validateSignature

Input: validateSignatureRequest (soap:body, use = literal) [Source code](#)

```
parameters type validateSignature
  SessionId type SessionIdType - type string with restriction
  ServiceProvider type ServiceProviderType - type string with restriction - enum { 'SIA-Colt' }
  Signer type string
  SignatureData type CDATA - type string with restriction
  holds the complete ds:Signature element embedded in a CDATA section
  Output: validateSignatureResponse (soap:body, use = literal) Source code
```

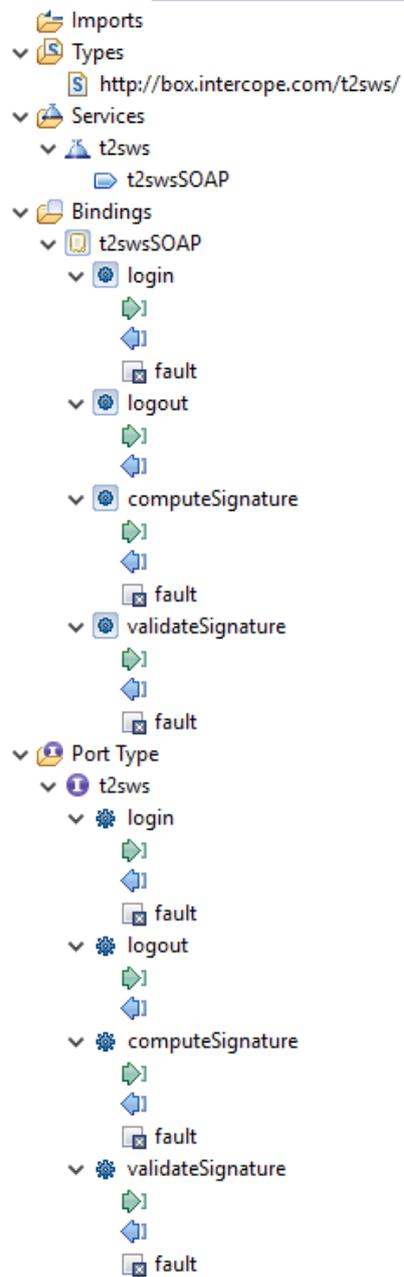
```
parameters type validateSignatureResponse
  Valid type boolean
```

Fault: validateSignatureFault (soap:fault, use = literal) [Source code](#)

```
parameters type validateSignatureFault
  Error type string
  Description type string
```

8.5 Programmatical Structure

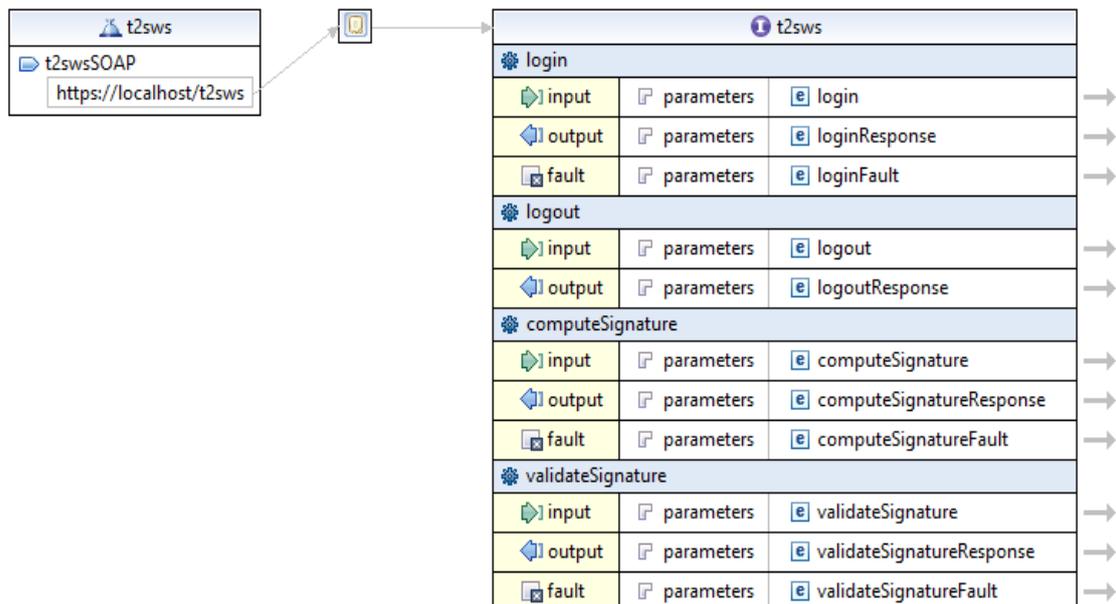
8.5.1 Outline



INTERCOPE

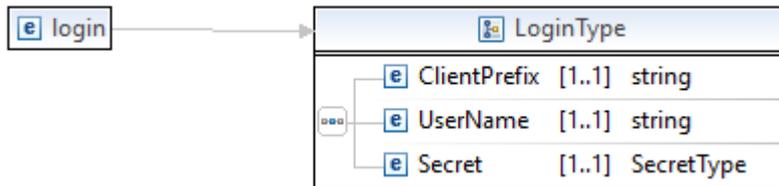
- ▼ Messages
 - ▼ loginRequest
 - parameters
 - ▼ loginResponse
 - parameters
 - ▼ loginFault
 - parameters
 - ▼ logoutRequest
 - parameters
 - ▼ logoutResponse
 - parameters
 - ▼ computeSignatureRequest
 - parameters
 - ▼ computeSignatureResponse
 - parameters
 - ▼ validateSignatureRequest
 - parameters
 - ▼ validateSignatureResponse
 - parameters
 - ▼ computeSignatureFault
 - parameters
 - ▼ validateSignatureFault
 - parameters

8.5.2 Generic Attributes



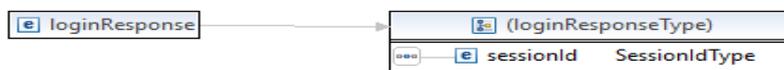
8.5.3 Login Operation

8.5.3.1 login



clientPrefix	Client Prefix name of the institute.
userName	UserName of the account, which is used for the retrieval.
secret	Password for the account, it can be plain text or encrypted.

8.5.3.2 LoginResponse



sessionId	During the log-in, an actual user object is created. The session Id is the pointer to this object. If 'true' login operation will create a JSESSIONID which can be used by load balancers to correctly maintain session stickiness
-----------	--

8.5.3.3 loginFault

If an error occurs during log-in, the error text is returned ("fault").



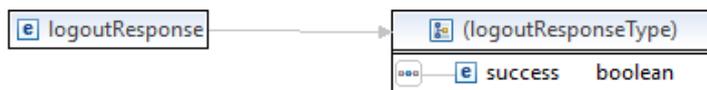
8.5.3.4 Logout Operation

After completing the methods the application must logout. This will free all allocated resources for this user object.



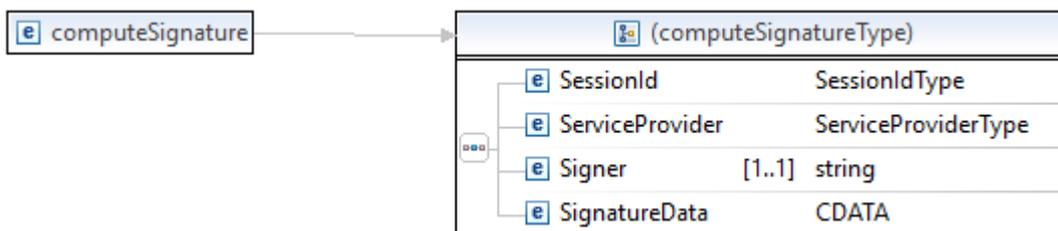
sessionId	During the log-in, an actual user object is created. The session Id is the pointer to this object. If 'true' login operation will create a JSESSIONID which can be used by load balancers to correctly maintain session stickiness
-----------	--

8.5.3.5 logoutResponse



success	type="xsd:boolean" true/false.
---------	--------------------------------

8.5.4 computeSignature



SessionID	During the log-in, an actual user object is created. The session Id is the pointer to this object. if 'true' login operation will create a JSESSIONID which can be used by load balancers to correctly maintain session stickiness
ServiceProvider	Type of Service Provider
Signer	e.g. Alliance Access
SignatureData	business signature (or a batch of signatures)

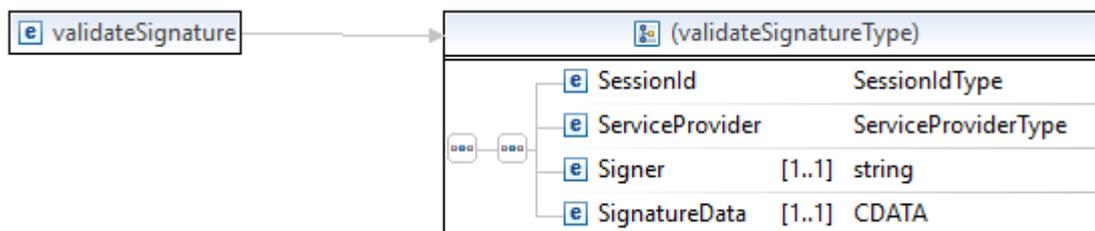


computeSignatureResponse	
--------------------------	--



Error	Error description of the computeSignatureResponse.
Description	Error description of the computeSignatureResponse.

8.5.5 validateSignature



SessionID	During the log-in, an actual user object is created. The session Id is the pointer to this object. if 'true' login operation will create a JSESSIONID which can be
-----------	--

	used by load balancers to correctly maintain session stickiness
ServiceProvider	Service Provider Name
Signer	Signer of Signature Request
SignatureData	plain text password/encrypted password holds the complete ds: Signature element embedded in a CDATA section



validateSignatureResponse	String, valid
---------------------------	---------------



Error	Error description of the validateSignature failure.
Description	string

8.6 Appendix

The Appendix lists the definition of the interface in the Web Service Description Language (WSDL).

8.6.1 Target2 Securities Signature Validation wdsi

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions xmlns:t2sws="http://box.intercope.com/t2sws/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:sia="http://www.sia.eu/t2s-s-1.0"
  name="t2sws"
  targetNamespace="http://box.intercope.com/t2sws/"
  <wsdl:types>
    <xsd:schema targetNamespace="http://box.intercope.com/t2sws/"
  elementFormDefault="qualified">
      <!--
        <xsd:import namespace="http://www.w3.org/2000/09/xmldsig#"
  schemaLocation="http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/xmldsig-
  core-schema.xsd" />
        <xsd:import namespace="http://www.sia.eu/t2s-s-1.0"
  schemaLocation="t2s-s.xsd" />
        <xsd:element name="dummy" type="sia:VersionType"/>
        -->
        <xsd:element name="login" type="t2sws:LoginType">
        </xsd:element>
        <xsd:element name="loginResponse">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="SessionId" type="t2sws:SessionIdType" />
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="loginFault">
          <xsd:complexType>
  
```

```

    <xsd:sequence>
      <xsd:element name="LoginFault"
type="xsd:string"></xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="logout">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="SessionId"
type="t2sws:SessionIdType"></xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="logoutResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Success" type="xsd:boolean"></xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:complexType name="SecretType">
  <xsd:choice>
    <xsd:element name="plain" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation>plain text password</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="encrypted" type="xsd:string">
      <xsd:annotation>
        <xsd:documentation>encrypted password</xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:choice>
</xsd:complexType>
<xsd:simpleType name="SessionIdType">
  <xsd:restriction base="xsd:string"></xsd:restriction>
</xsd:simpleType>
<xsd:element name="computeSignature">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="SessionId"
type="t2sws:SessionIdType" />
      <xsd:element name="ServiceProvider"
type="t2sws:ServiceProviderType">
    </xsd:element>
      <xsd:element name="Signer" type="xsd:string"
maxOccurs="1" minOccurs="1">
    </xsd:element>
      <xsd:element name="SignatureData" type="t2sws:CDATA">
        <xsd:annotation>
          <xsd:documentation>holds the complete ds:Signature
element (without ds:SignatureValue, ds:KeyInfo, the ds:Reference to it) embedded
in a CDATA section</xsd:documentation>
        </xsd:annotation></xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="computeSignatureResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="SignatureData" type="t2sws:CDATA">
        <xsd:annotation>
          <xsd:documentation>holds the complete ds:Signature
element embedded in a CDATA section</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>

```

```

</xsd:element>
<xsd:complexType name="LoginType">
  <xsd:sequence>
    <xsd:element name="ClientPrefix" type="xsd:string"
      maxOccurs="1" minOccurs="1">
      <xsd:annotation>
        <xsd:documentation></xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="UserName" type="xsd:string"
      maxOccurs="1" minOccurs="1">
      <xsd:annotation>
        <xsd:documentation></xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="Secret" type="t2sws:SecretType"
      maxOccurs="1" minOccurs="1">
      <xsd:annotation>
        <xsd:documentation></xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="MessageIdType">
  <xsd:restriction base="xsd:int">
    <xsd:minInclusive value="1"></xsd:minInclusive>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ShortNameType">
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="1"></xsd:minLength>
    <xsd:maxLength value="127"></xsd:maxLength>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="DisplayNameType">
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="1"></xsd:minLength>
    <xsd:maxLength value="255"></xsd:maxLength>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="resultType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="SUCCESS" />
    <xsd:enumeration value="ERROR" />
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="ServiceProviderType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="SIA-Colt"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:element name="validateSignature">
  <xsd:annotation>
    <xsd:documentation></xsd:documentation>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:sequence>
      <xsd:sequence>
        <xsd:element name="SessionId"
          type="t2sws:SessionIdType" />
        <xsd:element name="ServiceProvider"
          type="t2sws:ServiceProviderType">
        </xsd:element>
        <xsd:element name="Signer" type="xsd:string"
          maxOccurs="1" minOccurs="1">
        </xsd:element>
        <xsd:element name="SignatureData"
          maxOccurs="1" minOccurs="1" type="t2sws:CDATA">
          <xsd:annotation>

```

```

                <xsd:documentation>holds the complete
ds:Signature element embedded in a CDATA section</xsd:documentation>
                </xsd:annotation>
            </xsd:element>
        </xsd:sequence>
    </xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="validateSignatureResponse">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="Valid" type="xsd:boolean"></xsd:element>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="computeSignatureFault">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="Error" type="xsd:string">
            </xsd:element>
            <xsd:element name="Description"
                type="xsd:string">
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="validateSignatureFault">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="Error" type="xsd:string">
            </xsd:element>
            <xsd:element name="Description"
                type="xsd:string">
            </xsd:element>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:simpleType name="CDATA">
    <xsd:restriction base="xsd:string"></xsd:restriction>
</xsd:simpleType>
</xsd:schema>
</wsdl:types>
<wsdl:message name="loginRequest">
    <wsdl:part element="t2sws:login" name="parameters" />
</wsdl:message>
<wsdl:message name="loginResponse">
    <wsdl:part element="t2sws:loginResponse" name="parameters" />
</wsdl:message>
<wsdl:message name="loginFault">
    <wsdl:part name="parameters" element="t2sws:loginFault"></wsdl:part>
</wsdl:message>
<wsdl:message name="logoutRequest">
    <wsdl:part name="parameters" element="t2sws:logout"></wsdl:part>
</wsdl:message>
<wsdl:message name="logoutResponse">
    <wsdl:part name="parameters" element="t2sws:logoutResponse"></wsdl:part>
</wsdl:message>
<wsdl:message name="computeSignatureRequest">
    <wsdl:part name="parameters" element="t2sws:computeSignature"></wsdl:part>
</wsdl:message>
<wsdl:message name="computeSignatureResponse">
    <wsdl:part name="parameters"
element="t2sws:computeSignatureResponse"></wsdl:part>
</wsdl:message>
<wsdl:message name="validateSignatureRequest">
    <wsdl:part name="parameters"
element="t2sws:validateSignature"></wsdl:part>
</wsdl:message>
<wsdl:message name="validateSignatureResponse">
```

```
<wsdl:part name="parameters"
element="t2sws:validateSignatureResponse"></wsdl:part>
</wsdl:message>
<wsdl:message name="computeSignatureFault">
  <wsdl:part name="parameters">
    <wsdl:part name="parameters"></wsdl:part>
  </wsdl:part>
</wsdl:message>
<wsdl:message name="validateSignatureFault">
  <wsdl:part name="parameters">
    <wsdl:part name="parameters"></wsdl:part>
  </wsdl:part>
</wsdl:message>
<wsdl:portType name="t2sws">
  <wsdl:operation name="login">
    <wsdl:input message="t2sws:loginRequest" />
    <wsdl:output message="t2sws:loginResponse" />
    <wsdl:fault name="fault" message="t2sws:loginFault"></wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="logout">
    <wsdl:input message="t2sws:logoutRequest"></wsdl:input>
    <wsdl:output message="t2sws:logoutResponse"></wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="computeSignature">
    <wsdl:input message="t2sws:computeSignatureRequest"></wsdl:input>
    <wsdl:output message="t2sws:computeSignatureResponse"></wsdl:output>
    <wsdl:fault name="fault">
      <wsdl:fault name="fault"></wsdl:fault>
    </wsdl:operation>
  <wsdl:operation name="validateSignature">
    <wsdl:input message="t2sws:validateSignatureRequest"></wsdl:input>
    <wsdl:output message="t2sws:validateSignatureResponse"></wsdl:output>
    <wsdl:fault name="fault">
      <wsdl:fault name="fault"></wsdl:fault>
    </wsdl:operation>
  </wsdl:portType>
<wsdl:binding name="t2swsSOAP" type="t2sws:t2sws">
  <soap:binding style="document"
  transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="login">
    <soap:operation
      soapAction="http://box.intercope.com/t2sws/login" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
    <wsdl:fault name="fault">
      <soap:fault use="literal" name="fault" />
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="logout">
    <soap:operation
      soapAction="http://box.intercope.com/t2sws/logout" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="computeSignature">
    <soap:operation
      soapAction="http://box.intercope.com/t2sws/computeSignature" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
</wsdl:service>
</wsdl:definitions>
```

```
<wsdl:fault name="fault">
  <soap:fault use="literal" name="fault" />
</wsdl:fault>
</wsdl:operation>
<wsdl:operation name="validateSignature">
  <soap:operation
    soapAction="http://box.intercope.com/t2sws/validateSignature" />
  <wsdl:input>
    <soap:body use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body use="literal" />
  </wsdl:output>
  <wsdl:fault name="fault">
    <soap:fault use="literal" name="fault" />
  </wsdl:fault>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="t2sws">
  <wsdl:port binding="t2sws:t2swsSOAP" name="t2swsSOAP">
    <soap:address location="https://localhost/t2sws" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

9 Disclaimer

INTERCOPE International Communication Products Engineering GmbH (Intercope) and the stylized logo is the registered trademark of Intercope and its subsidiaries, in Germany and certain other countries. All other trademarks mentioned in this document are the acknowledged property of their respective owners.

Intercope provides this publication "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of non-infringement, merchantability or fitness for a particular purpose.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Intercope may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This information may contain sample application programs in source language, which illustrate programming and implementation techniques. You may copy, modify, and distribute these samples programs in any form without payment to Intercope, for the purposes of developing, using, marketing or distributing application programs for which the sample programs are written. These examples have not been thoroughly tested under all conditions. Intercope, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

The sample programs are provided "AS IS", without warranty of any kind. Intercope shall not be liable for any damages arising out of use of the sample programs.

Intercope grants the right to reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. Intercope does not allow derivative works of these publications, or to reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of Intercope.

Without written permission of Intercope no part of this publication may be modified and/or reproduced in any way.

INTERCOPE GmbH
Himmelstrasse 12-16,
22299 Hamburg,
Germany

+49 40 514 52 0
info@intercope.com

<https://www.intercope.com>

Copyright © 2019 INTERCOPE International Communication Products Engineering GmbH.

All Rights Reserved.