EUROPEAN COMMISSION

DIGIT
Connecting Europe Facility

Quick Start Guide

for the

Business Document Metadata Service Location (BDMSL)

Version [2.4]

Status [Final]

Date: 11/04/2022

Document Approver(s):

Approver Name	Role
Joao RODRIGUES	eDelivery
Bogan DUMITRIU	Project Manager

Document Reviewers:

Reviewer Name	Role
Jože RIHTARŠIČ	Developer

Summary of Changes:

Version	Date	Created by	Short Description of Changes
1.5	26/08/2016	Adrien FERIAL	Version
1.6	30/09/2016	Yves ADAM	Align to new template
1.7	19/07/2017	Flavio SANTOS	Add configuration parameters
1.8	16/01/2018	Flavio SANTOS	Add encryption configuration parameters
1.9	27/03/2018	CEF Support	Reuse policy notice added, e-SENS profile replaced by eDelivery profile
1.10	24/04/2018	Flavio SANTOS	Configuration files in detail
2.0	08/05/2019	Jože RIHTARŠIČ	Update for SML 4.0
2.1	04/06/2019	Jože RIHTARŠIČ	configuration for weblogic/oracle and add chapter for BIND DNS configuration added
2.2	21/01/2020	Jože RIHTARŠIČ	Add configuration parameters
2.3	30/11/2020	Jože RIHTARŠIČ	Added description for configuration property: dnsClient.use.legacy.regexp
2.4	07/04/2022	Caroline AEBY	No more CEF

Table of Contents

1. Introduction

BDMSL stands for Business Document Metadata Service Location. BDMSL is the sample implementation of the SML maintained by DG DIGIT. The version of the BDMSL refered in this document is 4.x versions. This version implements the eDelivery BDXL profile (see https://ec.europa.eu/digital-building-blocks/wikis/display/DIGITAL/eDelivery+BDXL)

1.1. Purpose of the Quick Start Guide

This document provides a brief description of the installation of the BDMSL component. Opposite to previous version, this version of the application does not use Liquibase as a database management tool. Before the installation, a database must be created using SQL scripts bundled in the sml-4.x-setup.zip file. The application bussines properties are stored in the database table BDMSL_CONFIGURATION. Application properties such as datasource JNDI, log folder, etc., are located in the smp.config.properties which must be located in the classpath of the server.

This guide illustrates the different steps to install the BDMSL application on a Tomcat server with a MySQL database and Weblogic 12.2.1.3 with an oracle database.

1.2. Pre-requisites

Please install the following software on the target system. For further information and installation details, please refer to the software owner's documentation.

- Java runtime environment (JRE) 8 **only**:
  http://www.oracle.com/technetwork/java/javase/downloads/index.html
- **One** of the supported Database Management Systems:
  - MySQL 5.7 or above
  - Oracle 10g+
- **One** of the supported Application Servers:
  - Tomcat 8
  - WebLogic 12.2

1.3. Binaries repository

The eDelivery BDMSL artefacts can be downloaded from the Digital site[1].

[1] https://ec.europa.eu/digital-building-blocks/wikis/display/DIGITAL/SML

1.4. Source Code Repository

The source code of eDelivery BDMSL is available in the **GIT** repository at the following location:

https://ec.europa.eu/cefdigital/code/projects/EDELIVERY/repos/bdmsl/browse

As mentioned in the prerequisites, the deployment of the eDelivery BDMSL was only tested on Tomcat 8.5 and WebLogic 12.2.1.3 application server.

The deployment of the eDelivery BDMSL is made of the following mandatory steps:

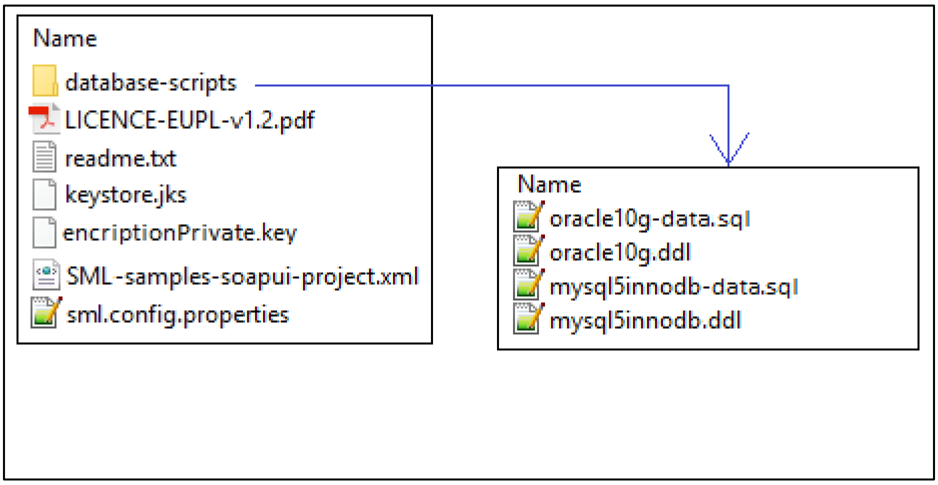
- Database configuration
- Application Server preparation
- BDMSL Initial configuration
- BDMSL file deployment

Remark:

> *The environment variable, **cef_edelivery_path**, refers to the name of the folder where the BDMSL package is installed (**CATALINA_HOME for Tomcat** and **DOMAIN_HOME for Oracle Weblogic**).*

1.5. Database Scripts

The scripts to create (or migrate) the Oracle or MySQL databases are included in the following downloadable zip file from the Digital site (section §1.3): sml-4.x-setup.zip.



Name
database-scripts
LICENCE-EUPL-v1.2.pdf
readme.txt
keystore.jks
encriptionPrivate.key
SML-samples-soapui-project.xml
sml.config.properties

Name
oracle10g-data.sql
oracle10g.ddl
mysql5innodb-data.sql
mysql5innodb.ddl

2. Database Creation

This section describes the steps necessary to create the database, the tables and the BDMSL database user (**dbuser** used for database connection purpose).

For this step you need to use the script included in the zip file downloaded in section §1.5.

2.1. MySQL database

1. Download and copy the mysql5innoDb.ddl script to cef_edelivery_path/database-scripts

2. Open a command prompt and navigate to the cef_edelivery_path/database-scripts folder

3. Execute the following MySQL commands **(WARNING: this step will delete the user schema if it already exists in the database)**:

```
mysql -h localhost -u root_user --password=root_password -e "drop schema if
exists bdmsl_schema;create schema bdmsl_schema;alter database bdmsl_schema
charset=utf8; create user sml_dbuser@localhost identified by
'sml_password';grant all on bdmsl_schema.* to sml_dbuser@localhost;"
```

This creates the bdmsl_*schema* and a *bdmsl database_dbuser* with (all) privileges for the bdmsl_schema**.**

Execute the following command to create the required objects (tables, etc.) in the database:

```
mysql -h localhost -u root_user -proot_password bdmsl_schema <
mysql5innoDb.ddl
```

Execute the following command to set up the initial data:

```
mysql -h localhost -u root_user -proot_password bdmsl_schema <
mysql5innoDb-data.sql
```

2.2. Oracle database

1. Download and copy the **oracle10g.ddl script** to *cef_edelivery_path/sql-scripts*

2. Navigate to *cef_edelivery_path*/**sql-scripts** directory

3. Execute the following commands:

```
sqlplus sys as sysdba (password should be the one assigned during the
Oracle installation )
```

```
=================================================================== Once
logged in Oracle: create user sml_dbuser identified by sml_dbpassword;

grant all privileges to sml_dbuser;

connect sml_dbuser

show user

(BDMSL environment property file) in the folder classes. (should return :
sml_dbuser)

(run the scripts with the @ sign from the location of the scripts)

@oracle10g.ddl  (the Oracle database creation)

@oracle10g-data.sql  (the Oracle init data)


exit
```

3. Tomcat Configuration

In order to deploy the BDMSL on Tomcat, the steps below need to be completed.

3.1. Configuring the Extra CLASSPATH for Tomcat

In this Tomcat example, a directory called **cef_edelivery_path** will be created in the root path of the Tomcat installation (**CATALINA_HOME**) and the **CLASSPATH** modified to include this new directory using an existing Tomcat batch file (CATALINA_HOME/bin/setenv.[sh|bat]).

- classes
- keystores

For Linux:

Edit the CATALINA_HOME/bin/setenv.sh file

```
#!/bin/sh
# Set CLASSPATH to include sml environment property file:
# sml.config.properties
export CLASSPATH=$CATALINA_HOME/classes
```

For Windows:

Edit the %CATALINA_HOME%/bin/setenv.bat file

```
REM Set CLASSPATH to include sml environment property file:
REM sml.config.properties


set classpath=%classpath%;%catalina_home%\classes
```

Place the **sml.config.properties** (BDMSL environment property file) in the folder classes.

Example can be downloaded from the Digital site (section §1.3): sml-4.x-setup.zip. Detailed description of environment properties is in section §1.3.

For tomcat/mysql configuration the file must have following properties and values:

```
sml.hibernate.dialect=org.hibernate.dialect.MySQLDialect
sml.datasource.jndi=java:comp/env/jdbc/edelivery
sml.jsp.servlet.class=org.apache.jasper.servlet.JspServletsml.log.folder=./
logs/
```

3.2. Configuring the Datasource for Tomcat

Create a new data source in Tomcat named: java:comp/env/jdbc/edelivery.

For that go to TOMCAT_HOME/conf/context.xml and add the block:

```
<Resource name="jdbc/edelivery" auth="Container" type="javax.sql.DataSource"
          maxTotal="100" maxIdle="30" maxWaitMillis="10000"
          username="root" password="root" driverClassName="com.mysql.jdbc.Driver"
          url="jdbc:mysql://localhost:3306/bdmsl"/>
```

3.3. JDBC Driver

The JDBC driver needs to be downloaded from the manufacturer website:

- For Mysql: https://www.mysql.com/products/connector/

The JDBC driver (.jar file) must be copied to the following directory: cef_edelivery_path/lib.

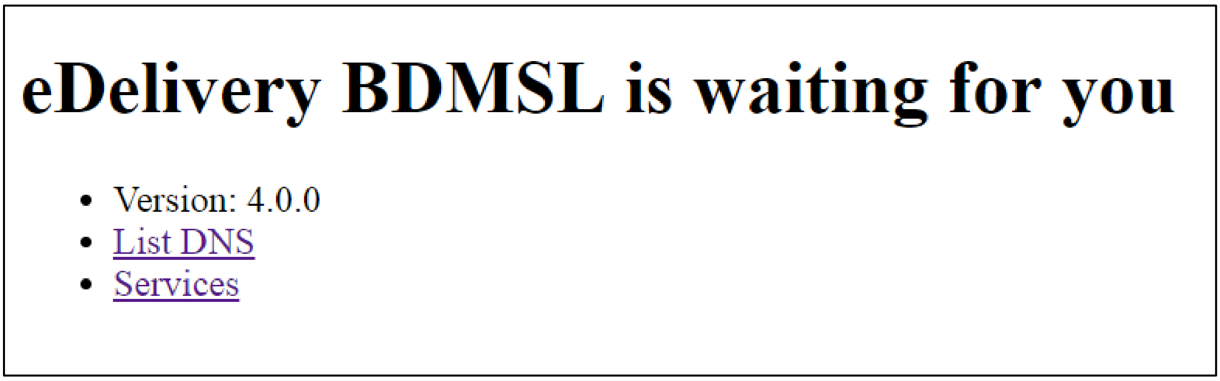
3.4. Deployment

Copy the **cef_bdmsl-webapp-4.X.war** file to the Tomcat **webapps** directory (cef_edelivery/webapps).

3.5. Verification of the Installation

Use your browser to go to the following address: **http://*[hostname]*:*[port]*/bdmsl-webapp-4.0.0/**

If the deployment is successful, the following page is displayed:



eDelivery BDMSL is waiting for you

- Version: 4.0.0
- List DNS
- Services

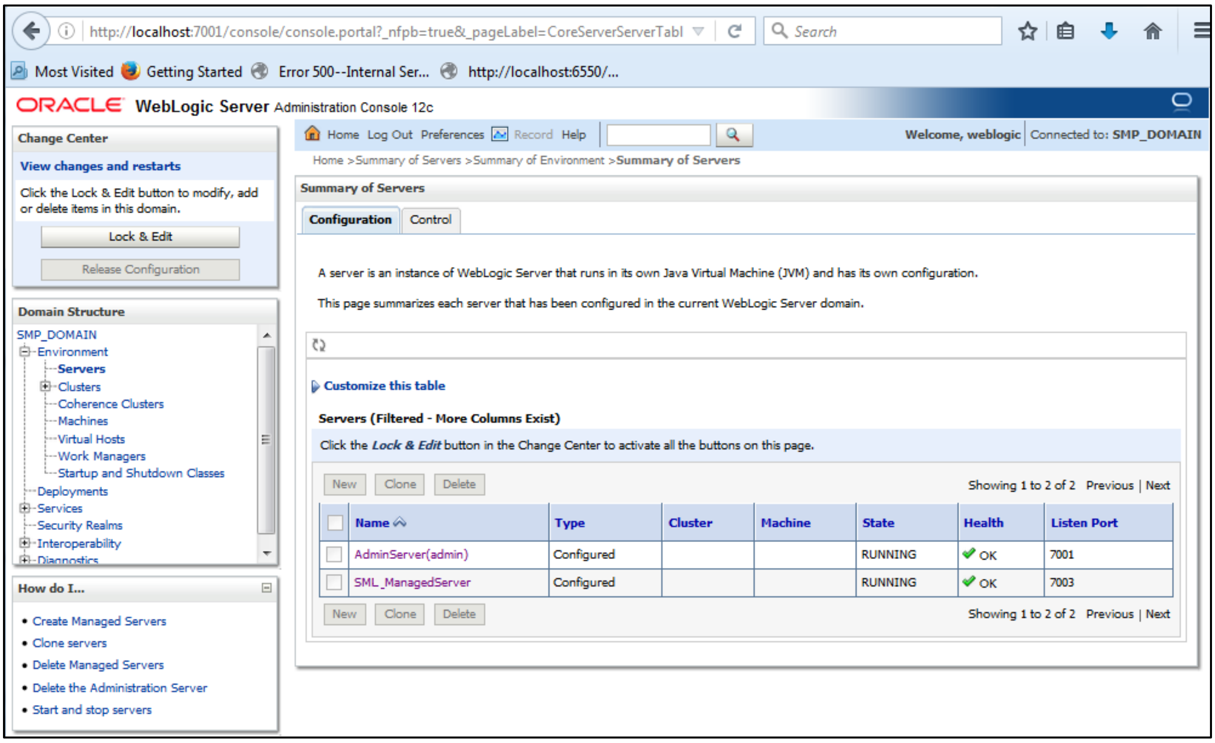
Important: Context path (example above: /**bdmsl-webapp-4.0.0**) should be the same as is deployment WAR file. If the war file is called ***sml.war*** then the URL will be **http://*[hostname]*:*[port]*/sml**.

4. WEBLOGIC CONFIGURATION

This section does not include the installation of a WebLogic 12.2.x application server. It is assumed that the WebLogic Server is installed and a WebLogic domain is created with an administration server and a managed server on which the BDMSL will be deployed.
Hereafter the domain location will be referred as *DOMAIN_HOME* (user-defined name).
In the examples below, we will use the following Domain and Server names:

- Domain Name : SMLDOMAIN
- Administration Server : AdminServer
- SMP Managed Server : SML_ManagedServer

As shown below:



Click the Lock & Edit button to modify, add or delete items in this domain.

A server is an instance of WebLogic Server that runs in its own Java Virtual Machine (JVM) and has its own configuration.

This page summarizes each server that has been configured in the current WebLogic Server domain.

Servers (Filtered - More Columns Exist)

Click the *Lock & Edit* button in the Change Center to activate all the buttons on this page.

□	Name	Type	Cluster	Machine	State	Health	Listen Port
□	AdminServer(admin)	Configured			RUNNING	✓ OK	7001
□	SML_ManagedServer	Configured			RUNNING	✓ OK	7003

In order to deploy the SMP on the WebLogic Application Server platform, two preliminary steps need to be completed:

- Configuring the Extra CLASSPATH for WebLogic
- Configure datasource

This is described in the following two sections.

4.1. Configuring the Extra CLASSPATH for WebLogic

Under the DOMAIN_HOME directory, create the following sub-directories:

- classes
- logs

Edit the WebLogic DOMAIN_HOME/bin/setDomainEnv.sh.
For Linux:
Add the **EXPORT CLASSPATH=${CLASSPATH}:${DOMAIN_HOME}/classes/** statement at the end of the CLASSPATH definition as shown below:

```
../
if [ "${PRE_CLASSPATH}" != "" ] ; then
CLASSPATH="${PRE_CLASSPATH}${CLASSPATHSEP}${CLASSPATH}"
export CLASSPATH
fi
CLASSPATH=${CLASSPATH}:${DOMAIN_HOME}/classes
export CLASSPATH
/..
```

For Windows:

```
../
If NOT "%PRE_CLASSPATH%"=="" (
set CLASSPATH=%PRE_CLASSPATH%;%CLASSPATH%
)
set CLASSPATH=%CLASSPATH%;%DOMAIN_HOME%\classes
/..
```

Place the **sml.config.properties** (BDMSL environment property file) in the folder classes.

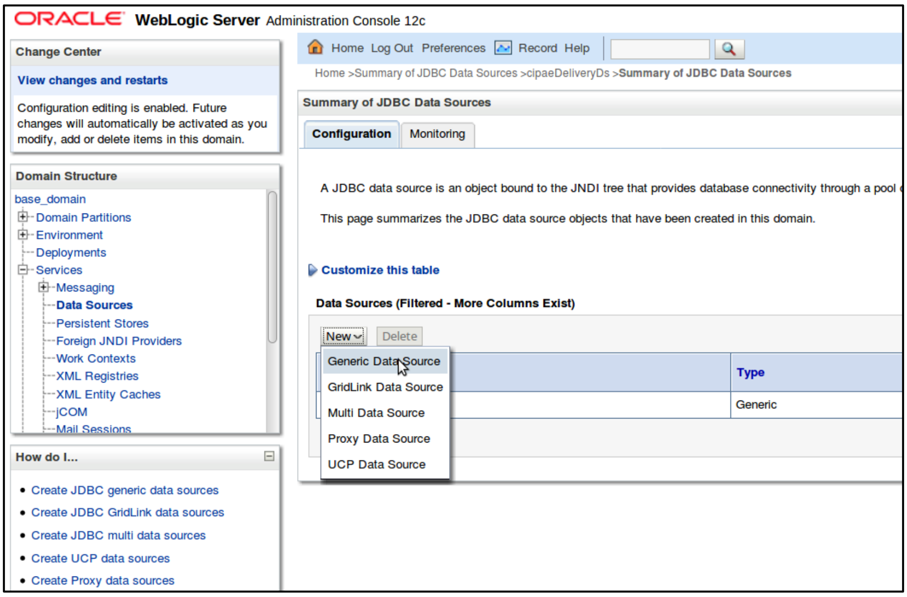
An example can be downloaded from the Digital site (section §1.3): sml-4.x-setup.zip. Detailed description of environment properties is in section §1.3.

For weblogic/oracle configuration, the file must have following properties and values:

```
sml.hibernate.dialect=org.hibernate.dialect.Oracle10gDialect
sml.datasource.jndi=jdbc/cipaeDeliveryDs
sml.jsp.servlet.class=weblogic.servlet.JSPServlet
sml.log.folder=./logs/
```

4.2. Configuring datasource for WebLogic

Clik on Services/Data sources on left Domain structure panel. Then on configuration tab click on button 'New' and select 'Generic data source'.



Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

A JDBC data source is an object bound to the JNDI tree that provides database connectivity through a pool

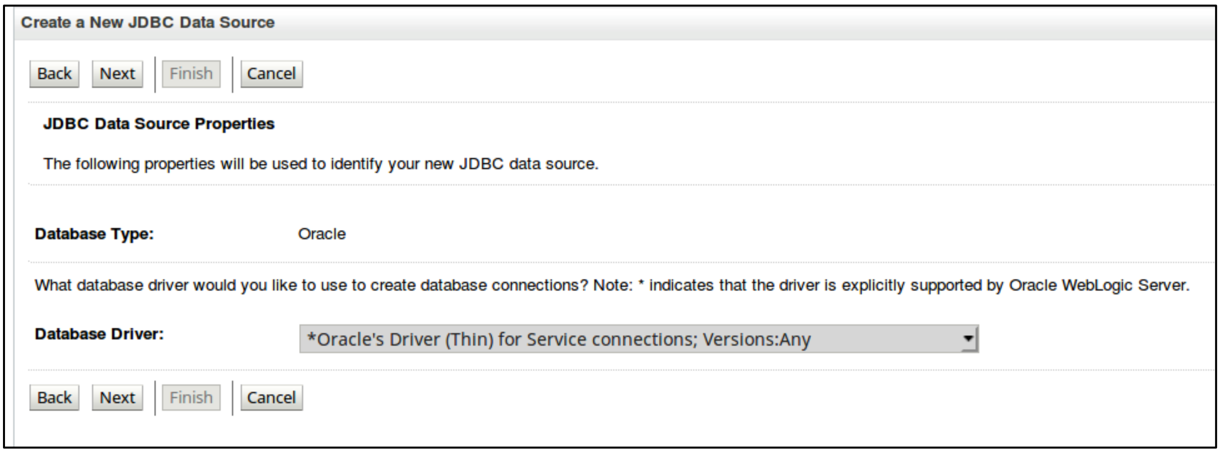
This page summarizes the JDBC data source objects that have been created in this domain.

New datasource wizard ' *Create a New Data Source* ' is triggered which will guide you thought Datasource creation. In the first wizard page, enter the following values:

Set Name value: *cipaeDeliveryDS*
JNDI name: jdbc/*cipaeDeliveryDS*
Database Type: *oracle*

Click then on next.

In next wizard page select Database driver: Oracle's Driver (Thin) and click next twice.



JDBC Data Source Properties

The following properties will be used to identify your new JDBC data source.

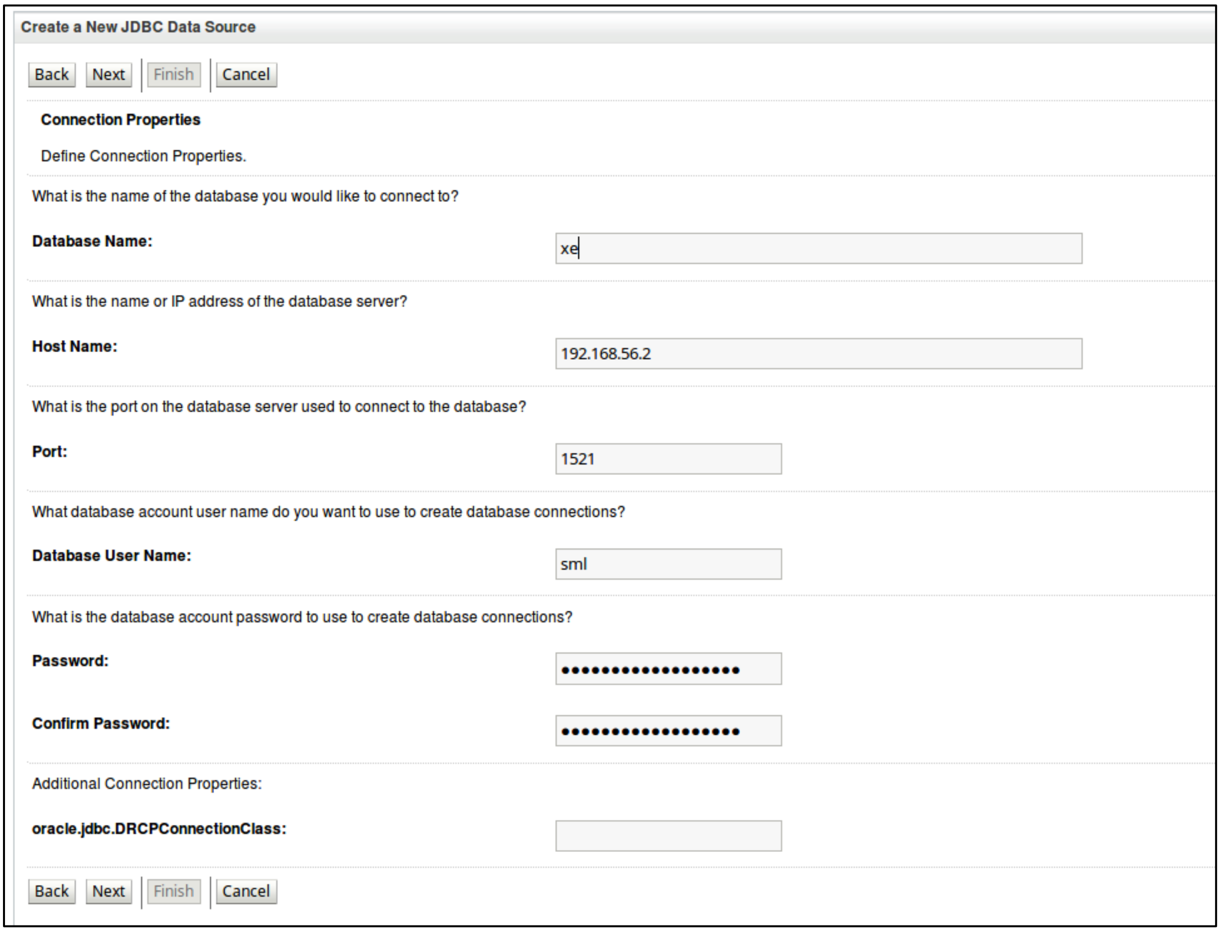
Database Type: Oracle

What database driver would you like to use to create database connections? Note: * indicates that the driver is explicitly supported by Oracle WebLogic Server.

Database Driver: *Oracle's Driver (Thin) for Service connections; Versions:Any

In the following wizard page, enter the datasource values (the values below are just an example: use the values from your oracle configuration):

Database Name: xe
Port: 1521
Database user sml_dbUser
Pasword: sml_dbPassword
Confirm password: sml_dbPassword



Create a New JDBC Data Source

Connection Properties

Define Connection Properties.

What is the name of the database you would like to connect to?

Database Name: xe

What is the name or IP address of the database server?

Host Name: 192.168.56.2

What is the port on the database server used to connect to the database?

Port: 1521

What database account user name do you want to use to create database connections?

Database User Name: sml

What is the database account password to use to create database connections?

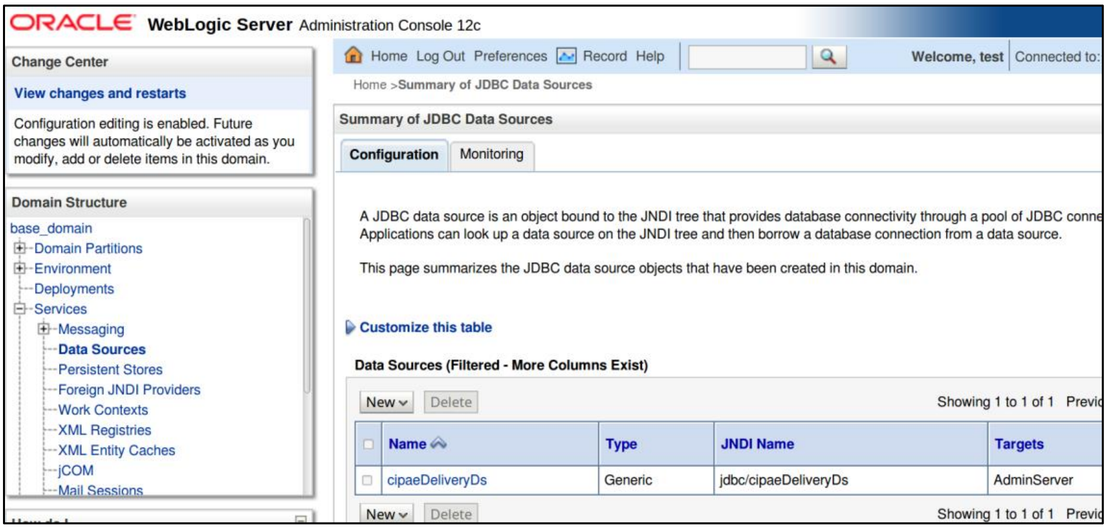
Password: ••••••••••••••••••

Confirm Password: ••••••••••••••••••

Additional Connection Properties:

oracle.jdbc.DRCPConnectionClass:

Then click 'Next' followed by click on 'Finish' button. Then a new Datasource configuration appears in the datasource table:



Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.

Summary of JDBC Data Sources

A JDBC data source is an object bound to the JNDI tree that provides database connectivity through a pool of JDBC conne
Applications can look up a data source on the JNDI tree and then borrow a database connection from a data source.

This page summarizes the JDBC data source objects that have been created in this domain.

Data Sources (Filtered - More Columns Exist)

Showing 1 to 1 of 1 Previ

□	Name	Type	JNDI Name	Targets
□	cipaeDeliveryDs	Generic	jdbc/cipaeDeliveryDs	AdminServer

Showing 1 to 1 of 1 Previ

4.3. Deployment

Deploy the **.war** file within WebLogic using the Oracle Weblogic deployer feature or using the Weblogic Administration Console.

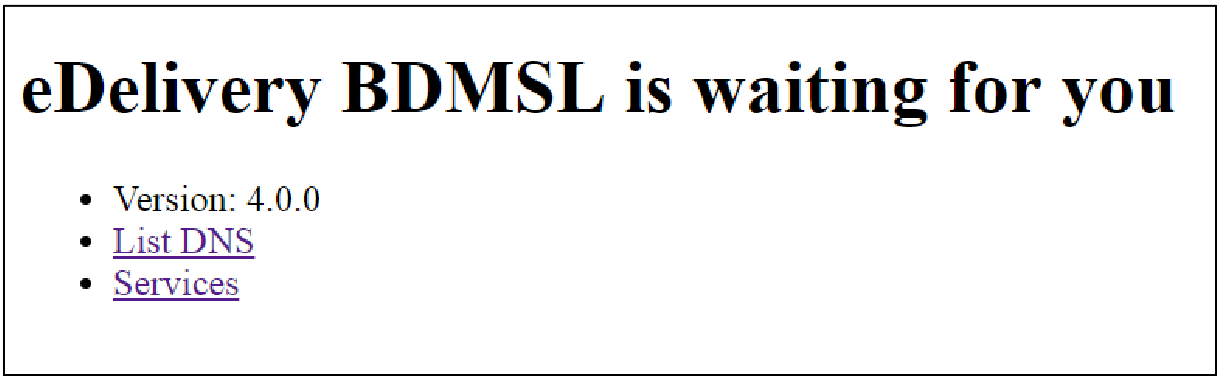
An example of using the Oracle the **weblogic.deployer** is shown below:

```
java weblogic.Deployer -adminurl
t3://${WebLogicAdminServerListenAddress}:${WebLogicAdminServerPort} \
-username ${WebLogicAdminUserName} \
-password ${WebLogicAdminUserPassword} \
-deploy -name bdmsl-webapp-4.X.war \
-targets ${SMP_ManagedServer} \
-source $TEMP_DIR/bdmsl-webapp-4.X.war
```

4.4. Verification of the Installation

Use your browser to navigate to the following address: **http://*[hostname]*:*[port]*/edelivery-sml/**

If the deployment is successful, the following page is displayed:



eDelivery BDMSL is waiting for you

- Version: 4.0.0
- List DNS
- Services

5. Configuration

5.1. Environment parameters

BDMSL application has environment parameters stored in property file sml.config.properties. Configuration is in property file because they are required before database connection. In the setup bundle sml-4.x-setup.zip (section §1.5), there is example of configuration preset for Tomcat/MySql installation:

```
# *********************************
# Hibernate dialect configuration
# *********************************
# Oracle hibernate example
#sml.hibernate.dialect=org.hibernate.dialect.Oracle10gDialect
# Mysql dialect
sml.hibernate.dialect=org.hibernate.dialect.MySQLDialect

# *********************************
# Datasource JNDI configuration
# *********************************
# weblogic datasource JNDI example
#sml.datasource.jndi=jdbc/cipaeDeliveryDs
# tomcat datasource JNDI example
sml.datasource.jndi=java:comp/env/jdbc/edelivery

# *********************************
# JSP implementation configuration
# *********************************
# Weblogic
#sml.jsp.servlet.class=weblogic.servlet.JSPServlet
# tomcat, jboss
sml.jsp.servlet.class=org.apache.jasper.servlet.JspServlet


# *********************************
# Logging implementation
# *********************************
sml.log.folder=./logs/
```

The configuration file has the following parameters:

- **sml.hibernate.dialect**: hibernate dialect for accessing the database
- **sml.datasource.jndi**: datasource JNDI name configured in section §1.5
- **sml.jsp.servlet.class**: application server implementation of JSP framework
- **sml.log.folder**: logging folder.

5.2. BDMSL parameters

BDMSL application contains its parameters in database table BDMSL_CONFIGURATION. Parameters can be updated:

- via the sql script as showed below:

```
mysql -h localhost -u root_user -proot_password bdmsl_schema -e "update
bdmsl_configuration set value='true', last_updated_on=NOW() where
property='unsecureLoginAllowed'";
```

- or by calling the webservice operation: BDMSLAdminServices/SetProperty(). For more details, check the ICD document.

All properties are refreshed without server restart, except CRON schedule definitions: sml.property.refresh.cronJobExpression, certificateChangeCronExpression and dataInconsistencyAnalyzer.cronJobExpression.

Properties are refreshed as defined by the cron property: sml.property.refresh.cronJobExpression. By default, properties are refreshed (if changed) every hour. If a property is changed by the sql script, make sure that the value ***last_updated*** is also changed, otherwise the properties will not be updated.

Property	Example	Mandatory	Description	Enc.
adminPassword	$2a$10$...Bi	FALSE	BCrypt Hashed password to access admin services	FALSE
authentication.bluecoat.enabled	FALSE	TRUE	Is blue coat enabled. Possible values: true/false.	FALSE
authorization.smp.certSubjectRegex	^.*(CN=SMP_| OU=PEPPOL TEST SMP).*$	TRUE	User with ROOT-CA is granted SMP_ROLE only if its certificates Subject matches configured regexp	FALSE
unsecureLoginAllowed	FALSE	TRUE	true if the use of HTTPS is not required. If the VALUES is set to true, then the user unsecure-http-client is automatically created. Possible VALUES: true/false	FALSE
authorization.domain.legacy.enabled	TRUE	TRUE	If legacy authorization is enabled, then domain authorization is done based only on domain certificate table data comparing certificate Subject or Issuer Values. In case of false: BDMSL must have SML truststore configured. And the Domain Trust is verified also by the BDMSL truststore. In case of false value Clien-Cert header cannot be used.	FALSE
cert.revocation.validation.graceful	TRUE	TRUE	In case of authorization.domain.legacy.enable	FALSE

Property	Example	Mandatory	Description	Enc.
			d is ser to false. All certificate in truststore chain are validated and CRL url is retrieved from the certificates directly. Graceful validation of certificate revocation. If URL retrieving does not succeed, do not throw error.	
cert.revocation.validation.crl.protocols	http://,https://",	TRUE	In case of authorization.domain.legacy.enabled is set to false. All certificate in truststore chain are validated and CRL url is retrieved from the certificates directly. Comma separated list of allowed crl protocols for fetching the CRL list.	FALSE
configurationDir	./	TRUE	The path to the folder containing all the configuration files (keystore and sig0 key)	FALSE
sml.property.refresh.cronJobExpression	0 53 */1 * * *	TRUE	Property refresh cron expression (def 7 minutes to each hour)!	FALSE
certificateChangeCronExpression	0 0 2 ? * *	TRUE	Cron expression for the changeCertificate job. Example: 0 0 2 ? * * (everyday at 2:00 am)	FALSE
dataInconsistencyAnalyzer.cronJobExpression	0 0 3 ? * *	TRUE	Cron expression for dataInconsistencyChecker job. Example: 0 0 3 ? * * (everyday at 3:00 am)	FALSE
dataInconsistencyAnalyzer.recipientEmail	email@domain.com	TRUE	Email address to receive Data Inconsistency Checker results	FALSE
dataInconsistencyAnalyzer.senderEmail	automated-notifications@nsome-mail.eu	TRUE	Sender email address for reporting Data Inconsistency Analyzer.	FALSE
dataInconsistencyAnalyzer.serverInstance	localhost	TRUE	Server instance (hostname) to generate report.	FALSE
mail.smtp.host	mail.server.com	TRUE	Email server - configuration for submitting the emails.	FALSE
mail.smtp.port	25	TRUE	Smtp mail port - configu1ration for submitting the emails.	FALSE
mail.smtp.protocol	smtp	TRUE	smtp mail protocol- configuration for submitting the emails.	FALSE
mail.smtp.username		FALSE	smtp mail protocol- username for submitting the emails.	FALSE
mail.smtp.password		FALSE	smtp mail protocol - encrypted password for submitting the emails.	TRUE

Property	Example	Mandatory	Description	Enc.
mail.smtp.properties		FALSE	smtp mail ;-separated properties: ex: mail.smtp.auth:true;mail.smtp.starttls.enable:true;mail.smtp.quitwait:false.	FALSE
dnsClient.SIG0Enabled	FALSE	TRUE	true if the SIG0 signing is enabled. Required fr DNSSEC. Possible VALUES: true/false	FALSE
dnsClient.SIG0KeyFileName	SIG0.private	TRUE	The actual SIG0 key file. Should be just the filename if the file is in the classpath or in the configurationDir	FALSE
dnsClient.SIG0PublicKeyName	sig0.acc...ec.test.eu.	TRUE	The public key name of the SIG0 key	FALSE
dnsClient.enabled	FALSE	TRUE	true if registration of DNS records is required. Must be true in production. Possible VALUES: true/false	FALSE
dnsClient.use.legacy.regexp	FALSE	TRUE	If value is 'true', then OASIS_BDXL regexp '^.*$' is used for NAPTR value generation else it is used the regular expression '.*' as defined in IETF RFC 4848.	FALSE
dnsClient.tcp.timeout	TRUE	FALSE	DNS TCP timeout in seconds. If the value is not given then tcp timeout is set to default value 60s.	FALSE
dnsClient.publisherPrefix	publisher	TRUE	This is the prefix for the publishers (SMP). This is to be concatenated with the associated DNS domain in the table bdmsl_certificate_domain	FALSE
dnsClient.server	ddnsext.tech.ec.europa.eu	TRUE	The DNS server.	FALSE
dnsClient.show.entries	TRUE	FALSE	If true than service ListDNS transfer and show the DNS entries. (Not recommended for large zones) Possible VALUES: true/false	FALSE
smp.update.max.part.size	1000	FALSE	Maximum number of participants on SMP which are automatically updated/deleted when calling services: ManageServiceMetadataService/Update ManageServiceMetadataService/Delete If SMP has more participants, then for	FALSE

Property	Example	Mandatory	Description	Enc.
			- delete: the participants must be deleted first using delete participant service; - update (only for SMP logical address when using NAPTR records): the creation of new SMP ID and migration participant to new SMP is only option.	
encriptionPrivateKey	encriptionPrivateKey.private	TRUE	Name of the 256 bit AES secret key to encrypt or decrypt passwords.	FALSE
useProxy	FALSE	TRUE	true if a proxy is required to connect to the internet. Possible VALUES: true/false	FALSE
httpProxyHost	localhost	TRUE	The http proxy host	FALSE
httpProxyPassword	vXA7JjCyEN1Qwg==	TRUE	Base64 encrypted password for Proxy.	TRUE
httpProxyPort	8012	TRUE	The http proxy port	FALSE
httpProxyUser	user	TRUE	The proxy user	FALSE
signResponse	FALSE	TRUE	true if the responses must be signed. Possible values: true/false	FALSE
keystoreAlias	senderalias	TRUE	The alias in the keystore for signing reponses.	FALSE
keystoreFileName	keystore.jks	TRUE	The JKS keystore file. Should be just the filename if the file is in the classpath or in the configurationDir	FALSE
keystorePassword	vXA7JjCy0EN1Qwg==	TRUE	Base64 encrypted password for Keystore.	TRUE
truststoreFileName	truststore.p12	TRUE	The truststore file (JKS or p12) should be just the filename if the file is in the classpath or in the configurationDir.	FALSE
truststorePassword	vXA7JjCy0EN1Qwg==	TRUE	Base64 encrypted password for Truststore.	TRUE

5.3. How to generate a private key file

SML provides a tool to create a private key to encrypt proxy and signing keystore passwords. In order to create a private key, please follow the steps below:

- Download one of the latest BDMSL war files (eg: bdmsl-webapp-4.0.x.war) from the repository https://ec.europa.eu/digital-building-blocks/wikis/display/DIGITAL/SML
- Extract the war file using any extracting tool
- Run the following commands to create a private key:

    1. cd bdmsl-webapp-4.0.x
    2. java -cp "WEB-INF/lib/*" eu.europa.ec.bdmsl.common.util.PrivateKeyGenerator c:\temp\encriptionPrivateKey.private

    Required parameter = Full directory path where the private key will be created

Example:

Printed result:

Private key created at c:\temp\encriptionPrivateKey.private

Once the private key is generated, please copy the private key file name to the value of the property `encriptionPrivateKey` in the table BDMSL_Configuration, and copy the private file to the path configured in the property `configurationDir`.

5.4. How to encrypt a password

If using webservices for setting passwords, the passwords are encrypted automatically. Below you will find the procedure for manual password encryption.

After generating a private key at item “§5.3- How to generate a private key file”, please configure the proxy or keystore (used to sign response) password if needed as follows:

- Inside the folder already extracted from BDMSL .war file, please run the command below:

    java -cp "WEB-INF/lib/*" eu.europa.ec.bdmsl.common.util.EncryptPassword c:\temp\privateKey.private Password123

    1st parameter = private key location

    2nd parameter = plain text password

- To configure the proxy password, please copy the printed encrypted and base64 encoded password to the value of the `httpProxyPassword` property in the table `BDMSL_CONFIGURATION`.

    Example:

httpProxyPassword = vXA7JjCy0iDQmX1UEN1Qwg==

- To configure the keystore password, please copy the printed encrypted and base64 encoded password to the value of the `keystorePassword` property in the table BDMSL_CONFIGURATION.

  Example:

  keystorePassword = vXA7JjCy0iDQmX1UEN1Qwg==

5.5. Certificate to sign responses

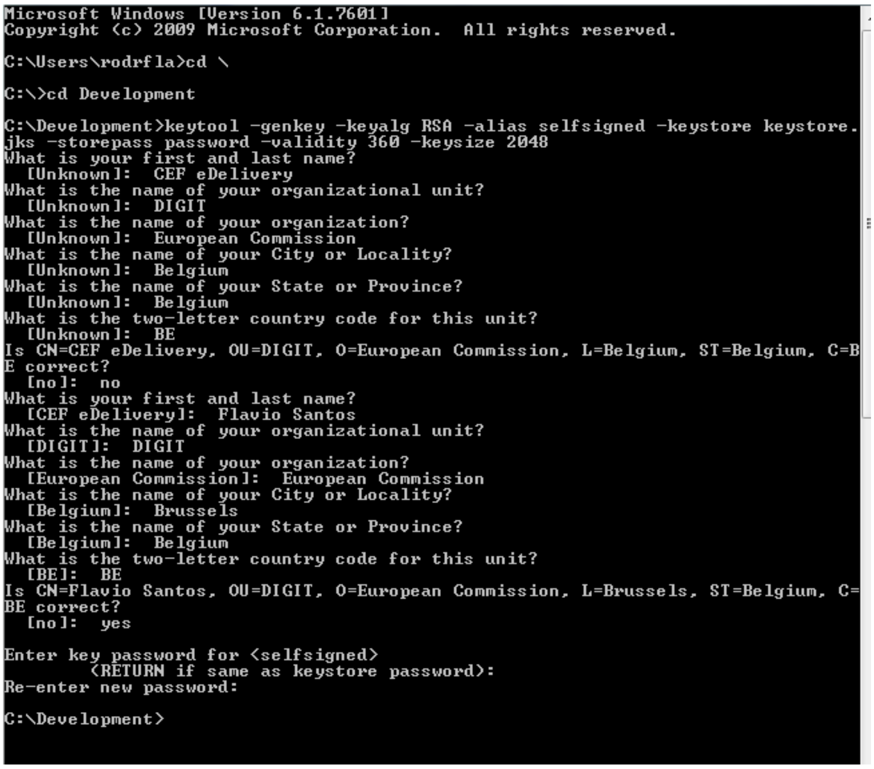
If the flag `signResponse=true` in the table `BDMSL_CONFIGURATION`, a keystore file name, its alias and password must be provided in the same table.

For testing purposes only, it is possible to create a self-signed keystore as follows:

- Open the command console on whatever operating system you are using and navigate to the directory where keytool.exe is located (usually where the JRE is located, e.g. c:\Program Files\Java\jre8\bin on Windows machines).
- Run the following command (where validity is the number of days before the certificate will expire):

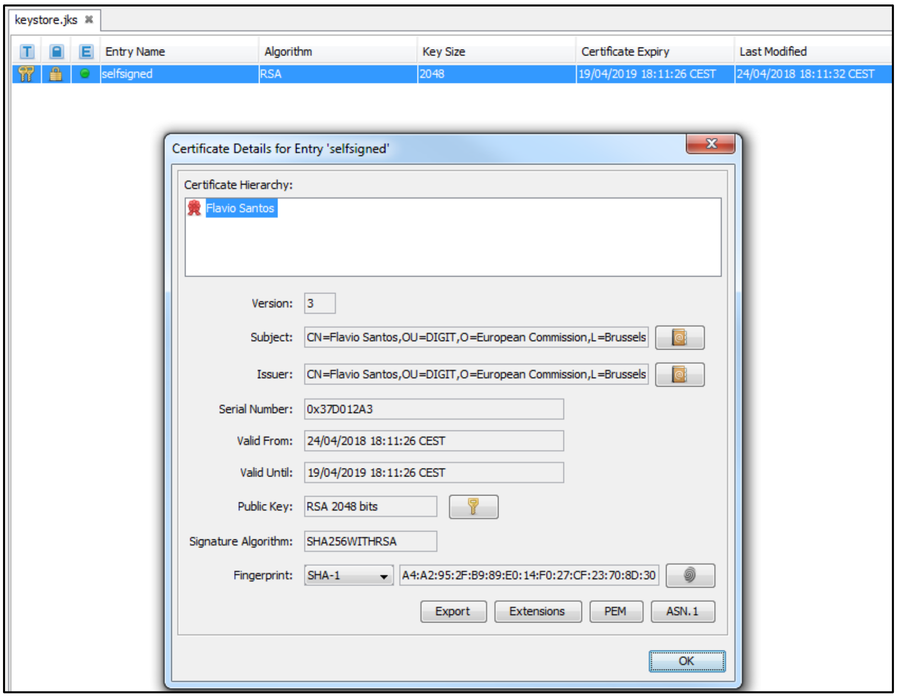
```
keytool -genkey -keyalg RSA -alias selfsigned -keystore keystore.jks -storepass password -validity 360 -keysize 2048
```

- Fill in the prompts for your organization information as below:



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\rodrfla>cd \

C:\>cd Development

C:\Development>keytool -genkey -keyalg RSA -alias selfsigned -keystore keystore.
jks -storepass password -validity 360 -keysize 2048
What is your first and last name?
  [Unknown]:  CEF eDelivery
What is the name of your organizational unit?
  [Unknown]:  DIGIT
What is the name of your organization?
  [Unknown]:  European Commission
What is the name of your City or Locality?
  [Unknown]:  Belgium
What is the name of your State or Province?
  [Unknown]:  Belgium
What is the two-letter country code for this unit?
  [Unknown]:  BE
Is CN=CEF eDelivery, OU=DIGIT, O=European Commission, L=Belgium, ST=Belgium, C=B
E correct?
  [no]:  no
What is your first and last name?
  [CEF eDelivery]:  Flavio Santos
What is the name of your organizational unit?
  [DIGIT]:  DIGIT
What is the name of your organization?
  [European Commission]:  European Commission
What is the name of your City or Locality?
  [Belgium]:  Brussels
What is the name of your State or Province?
  [Belgium]:  Belgium
What is the two-letter country code for this unit?
  [BE]:  BE
Is CN=Flavio Santos, OU=DIGIT, O=European Commission, L=Brussels, ST=Belgium, C=
BE correct?
  [no]:  yes

Enter key password for <selfsigned>
        (RETURN if same as keystore password):
Re-enter new password:

C:\Development>
```

- This will create a keystore.jks file containing a private key and your sparklingly fresh self-signed certificate. Now you just need to configure your Java application to use the .jks file.



Entry Name	Algorithm	Key Size	Certificate Expiry	Last Modified
selfsigned	RSA	2048	19/04/2019 18:11:26 CEST	24/04/2018 18:11:32 CEST

Certificate Details for Entry 'selfsigned'

Certificate Hierarchy:
Flavio Santos

Version: 3
Subject: CN=Flavio Santos,OU=DIGIT,O=European Commission,L=Brussels
Issuer: CN=Flavio Santos,OU=DIGIT,O=European Commission,L=Brussels
Serial Number: 0x37D012A3
Valid From: 24/04/2018 18:11:26 CEST
Valid Until: 19/04/2019 18:11:26 CEST
Public Key: RSA 2048 bits
Signature Algorithm: SHA256WITHRSA
Fingerprint: SHA-1 A4:A2:95:2F:B9:89:E0:14:F0:27:CF:23:70:8D:30

5.6. Files to be copied under application server

In the configuration directory that you specified in the `configurationDir` property, you need to put the following files:

- `keystore.jks` (the name can be changed in the property `keystoreFileName`): this keystore must contain your private key with the alias and password defined in the `keystoreAlias` and `keystorePassword` properties.

- `sig0.private` (the name can be changed in the property `dnsClient.SIG0KeyFileName`): this file is only required if you use DNSSEC (i.e. property `dnsClient.SIG0Enabled` set to true).

- `encriptionPrivateKey.private` (the name can be changed in the property `encriptionPrivateKey`): this private key file is only required if you use Proxy or Sign Response.

Once the needed files have been copied, restart the server(s).

5.7. DNS integration

BDMSL was developed and tested with using a BIND9 DNS server. The DNS integration can be switched on/off by setting attribute **dnsClient.enabled** to *true/false*. If the property is set to true, the parameter **dnsClient.server** must contain the hostname/ip address of the DNS server.

To secure the DNS integration, BDMSL has implemented SIG(0). This option can be enabled/disabled by the following parameter: **dnsClient.SIG0Enabled**, with values: *true/false.*

If the option is set to false, the DNS should allow updates to **any** ip address (this is **NOT** advised in production environment) or restrict the update permission to the requester **ip address**.

Below is example of configuration for BIND9 zone example.edelivery.eu.local without the use of SIG(0) (in this case the BDMSL should have **dnsClient.SIG0Enabled=false**):

```
zone "example.edelivery.eu.local" {
    type master;
    file "/var/lib/bind/db.example.edelivery.eu.local ";
    allow-update { 10.22.1.3;}
    allow-transfer { 10.22.0.0/16; };
};
```

5.7.1. Securing DNS integration with SIG(0)

SIG0 are asymmetric key-pairs, usually with a filename ending with .key for a public key, and a filename ending with .private for a private key.

In general: keys can be any of the asymmetric key algorithms: DSA, RSAMD5, RSASHA1. But BDMSL supports only DSA.

SIG(0) key pair can be created with dnssec-keygen utility (which is supplied as part of a BIND9 DNS server)

Example:

```
dnssec-keygen  -a DSA -b 1024 -n HOST -T KEY sig0.example.edelivery.eu.
local
```

The command produces the following files:

- Ksig0.example.edelivery.eu.local.+003+03054.key
- Ksig0.example.edelivery.eu.local.+003+03054.private

The content of the file: is as follows

```
Ksig0.example.edelivery.eu.+003+03054.key
```

It is the DNS Key entry, which should be put to DNS zone as in the example below:

```
sig0.example.edelivery.eu.local.      604800   IN      KEY     512 3 3
CLC4l6DtbztWAIJIMkYrv4MClWvj2BUclxqCd86vzX/f0ka+oS73dFCp
tb9Yv9oYjGmG1JLNv4EKuPiGPa8O/CQWrbJ5I7Yts3GDMgZNRswxMije
H6OoYkZ6ywRpjv8nommw6JMzDaDhcU5/tLQXhvz3U/c7W5QepAXfHb6Z
gGwL4TkqR/RGp5xcxayID4b/+DJvqi04BjNO9WR3XGRHWZ5aO0pRcRjx
imDtlnIjpsykE59o03UyQ+YT1CYNPjNlmOoT1JVgBEFGgouAm7yEZq3A
HWsqZEHCeucvQKBADmIk5rHwfZJwv7dzXrZR2U5AqE/AxqhrWyTpItRg
oGEkc+piGciuPRtwRZPkD6+GcFn/2knJ3YuRBOiog0+5mtbqaIPOew+B
+BtQk6X5E5tNnEuQJeRjjxznGYdzN7hTDFPvtwGEQvDUoU4SP/6YHoAd
AaH5Vs+YTRHjdISvnJIV6VRxIbQFJWaf3Z+UT4ns0+4pIGXm7C0ADA2a
1wGpj4QF8A37VAofcFWlUErtNv9YmVHQcA2l
```

When public key correctly registered to dns server it can be tested with dig util as example below:

```
$dig sig0.example.edelivery.eu.local  @localhost KEY


ANSWER
; <<>> DiG 9.10.3-P4-Ubuntu <<>> sig0.example.edelivery.eu.local @localhost
KEY
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 36443
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
; sig0.example.edelivery.eu.local.             IN      KEY

;; ANSWER SECTION:
sig0.example.edelivery.eu.local.      604800  IN      KEY     512 3 3
CLC4l6DtbztWAIJIMkYrv4MClWvj2BUclxqCd86vzX/f0ka+oS73dFCp
tb9Yv9oYjGmG1JLNv4EKuPiGPa8O/CQWrbJ5I7Yts3GDMgZNRswxMije
H6OoYkZ6ywRpjv8nommw6JMzDaDhcU5/tLQXhvz3U/c7W5QepAXfHb6Z
gGwL4TkqR/RGp5xcxayID4b/+DJvqi04BjNO9WR3XGRHWZ5aO0pRcRjx
imDtlnIjpsykE59o03UyQ+YT1CYNPjNlmOoT1JVgBEFGgouAm7yEZq3A
HWsqZEHCeucvQKBADmIk5rHwfZJwv7dzXrZR2U5AqE/AxqhrWyTpItRg
oGEkc+piGciuPRtwRZPkD6+GcFn/2knJ3YuRBOiog0+5mtbqaIPOew+B
+BtQk6X5E5tNnEuQJeRjjxznGYdzN7hTDFPvtwGEQvDUoU4SP/6YHoAd
AaH5Vs+YTRHjdISvnJIV6VRxIbQFJWaf3Z+UT4ns0+4pIGXm7C0ADA2a
1wGpj4QF8A37VAofcFWlUErtNv9YmVHQcA2l

;; AUTHORITY SECTION:
example.edelivery.eu.local.            604800  IN      NS      ns.
example.edelivery.eu.local.

;; ADDITIONAL SECTION:
ns.example.com.local.         604800  IN      A       192.168.56.3
```

To allow DNS updates for the zone “example.edelivery.eu.local " only by requests signed by private key of the **sig0.example.edelivery.eu.local** we have to update the DNS zone configuration as example:

```
zone "example.edelivery.eu.local" {
    type master;
    file "/var/lib/bind/db.example.edelivery.eu.local ";
    allow-update { key "sig0.example.edelivery.eu.local.";}
    allow-transfer { 10.22.0.0/16; };
};
```

5.7.2. Enabling SIG(0) in BDMSL

To enable BDMSL to use SIG(0) following parameters must be set:

Value of the parameter **dnsClient.SIG0PublicKeyName** must be DNS name of the DNS KEY entry, For the example above this value is:

dnsClient.SIG0PublicKeyName= sig0.example.edelivery.eu.local

Next, the private key must be put into to the BDMS configuration folder and Value of the parameter **dnsClient.SIG0KeyFileName** must be the name of the private key filename.

As example:

dnsClient.SIG0KeyFileName= Ksig0.example.edelivery.eu.local.+003+03054.private

Finaly we have to enable SIG(0) with parameter:

dnsClient.SIG0Enabled=true

Note that BDMSL for transfer is not using. BDMSL use transfer DNS records for generating inconsistency report and for when calling http get resource /listDNS
Therefore allow-transfer in DNS configuration must be set any or secured by IP.

```
zone "example.edelivery.eu.local" {
    type master;
    file "/var/lib/bind/db.example.edelivery.eu.local ";
    allow-update { key "sig0.example.edelivery.eu.local.";}
    allow-transfer { 10.22.0.0/16; };
};
```

6. Contact information

eDelivery Support Team

By email: CEF-EDELIVERY-SUPPORT@ec.europa.eu

Support Service: 8am to 6pm (Normal EC working Days)