EUROPEAN COMMISSION

DIGIT
Connecting Europe Facility

Access Point

Administration Guide

Domibus 3.2.5

Version [1.01]

Status [Validated]

Date: 10/10/2017

Document Approver(s):

Approver Name	Role
Adrien FERIAL	CEF Technical Office

Document Reviewers:

Reviewer Name	Role
Yves ADAM	CEF Technical Office

Summary of Changes:

Version	Date	Created by	Short Description of Changes
1.00	29/05/2017		
1.01	09/10/2017	Chaouki BERRAH	Update wildfly/oracle.

Table of Contents

1. INTRODUCTION 6

1.1. Purpose 6
1.2. References 6

2. CONVENTIONS 8

2.1. Example 1: Sample Oracle Statement 8
2.2. Example 2: Sample Configuration file 8

3. PREREQUISITES 9

3.1. Binaries repository 9

4. DOMIBUS DEPLOYMENT 10

4.1. Database Configuration 10
4.1.1. MySQL configuration 10
4.1.2. Oracle configuration 11
4.2. Domibus on WebLogic 12.1.3 12
4.2.1. Single Server Deployment 12
4.2.2. Clustered Deployment 23
4.3. Domibus on Tomcat 34
4.3.1. Pre-Configured Single Server Deployment 34
4.3.2. Single Server Deployment 38
4.3.3. Clustered Deployment 40
4.4. Domibus on WildFly 42
4.4.1. Pre-Configured Single Server Deployment 42
4.4.2. Single Server Deployment 46
4.4.3. Clustered Deployment 52

5. DOMIBUS CONFIGURATION 55

5.1. Security Configuration 56
5.1.1. Policies 56
5.1.2. Certificates 56
5.2. Domibus Properties 57

6. PLUGIN MANAGEMENT 60

6.1. Default Plugins 60
6.1.1. JMS Plugin 60
6.1.2. WS Plugin 60
6.1.2.1. Domibus authentication 60
6.1.2.2. Domibus Authorization 61
6.1.2.3. Enable the authentication in Domibus 61
6.2. Custom Plugin 62
6.2.1. Plugin registration 62

6.2.1.1. Tomcat....62
6.2.1.2. WebLogic....62
6.2.1.3. WildFly....62
6.3. PMode Configuration....63
6.3.1. Configuration....63
6.3.2. Adding a new participant....64
6.3.3. Sample PMode file....64
6.3.4. Domibus PMode configuration to ebMS3 PMode Mapping....68
6.3.5. Upload new Configuration....74
6.4. Administration Tools....76
6.4.1. Administration Dashboard....76
6.4.2. Application Logging....78
6.4.2.1. Domibus log file....78
6.4.2.2. Logging properties....79
6.4.2.3. Error Log page....80
6.4.3. Queue Monitoring....81
6.4.4. Configuration of the queues....87
6.4.4.1. Tomcat....87
6.4.4.2. WebLogic....87
6.4.4.3. WildFly....88
6.4.5. Message Filtering....88

7. DATA ARCHIVING....90

7.1. What's archiving?....90
7.2. Data Retention Policy....90
7.3. Data Extraction....90

8. NON REPUDIATION....91

9. TROUBLESHOOTING....92

9.1. Failed to obtain DB connection from datasource....92
9.2. Exception sending context initialized event to listener instance of class....93
9.3. Neither the JAVA_HOME nor the JRE_HOME environment variable is defined....93
9.4. Cannot access Admin Console....93
9.5. Handshake Failure....93

10. ANNEX 1 – TLS CONFIGURATION....97

10.1. TLS Configuration....97
10.1.1. Transport Layer Security in Domibus....97
10.1.2. Client side configuration (One Way SSL)....97
10.1.3. Client side configuration (Two Way SSL)....98
10.1.4. Server side configuration....99
10.1.4.1. Tomcat 8....99
10.1.4.2. WebLogic....100

10.1.4.3. Wildfly 9.....101
10.1.4.4. Configure Basic and Certificates authentication in SoapUI.....102
10.1.4.5. PMode update.....103

11. DYNAMIC DISCOVERY OF UNKNOWN PARTICIPANTS.....104

11.1. Overview.....104
11.2. Domibus configuration for PEPPOL.....105
11.3. PMode configuration (PEPPOL).....106
11.3.1. Sender PMode.....106
11.3.2. Receiver PMode.....107
11.4. Policy and certificates (PEPPOL).....108
11.5. Message format (PEPPOL).....108
11.6. SMP entry.....109
11.7. Domibus configuration for OASIS.....110
11.8. PMode configuration (OASIS).....111
11.8.1. Sender PMode.....111
Remark:.....112
11.8.2. Receiver PMode.....112
11.9. Policy and certificates (OASIS).....113
11.10. Message format (OASIS).....113

12. ANNEX 1 - USAGE OF CERTIFICATES IN PEPPOL AND OASIS.....114

13. ANNEX 2 – DOCUMENT PARTS.....115

14. LIST OF FIGURES.....116

15. LIST OF TABLES.....116

16. CONTACT INFORMATION.....117

1. INTRODUCTION

This Administration Guide is intended for Server Administrators in charge of installing, managing and troubleshooting an eDelivery Access Point.

1.1. Purpose

The purpose of this guide is to provide detailed information on how to deploy and configure Domibus 3.2.5 on WebLogic, Tomcat and WildFly with MySQL or Oracle. It also provides detailed descriptions of related Security Configurations (Policies, Certificates), Message Filtering, PMode Configuration, Application Monitoring, Custom Plugins Registration, JMS Monitoring, Data Archiving, Troubleshooting and TLS Configuration.

1.2. References

Ref.	Document	Content outline
[REF1]	https://ec.europa.eu/cefdigital/artifact/index.html#nexus-search;gav~eu.domibus~domibus-MSH~3.2.5~~	Location of the release artefacts on the Nexus repository
[REF2]	http://downloads.mysql.com/archives/c-j/	Location to download the MySQL JDBC driver from the Official website
[REF3]	http://www.oracle.com/technetwork/database/features/jdbc/default-2280470.html	Location of the Oracle JDBC driver from the Official website
[REF4]	https://docs.jboss.org/author/display/WFLY9/WildFly+9+Cluster+Howto	Location to the Official documentation on how to setup a cluster on WildFly 9
[REF5]	https://ec.europa.eu/cefdigital/wiki/download/attachments/23003408/%28CEF%20eDelivery%29.%28PKI%29.%28SOD%29.%28v2.5%29.pdf?api=v2	CEF Public Key Infrastructure (PKI) Service Offering Document

Ref.	Document	Content outline
[REF6]	https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Domibus+-+v3.2.5	Location of the Domibus 3.2.5 release on the Single Web Portal
[REF7]	https://access.redhat.com/documentation/en-US/Red_Hat_JBoss_Fuse/6.0/html/XML_Configuration_Reference/files/cxf-http-conf-2_7_0_xsd_Element_http-conf_tlsClientParameters.html	RedHat page for the XML Configuration Reference of the *http-conf:tlsClientParameters* element
[REF8]	http://wiki.ds.unipi.gr/display/ESENSPILOTS/5.1.1+-+Architecture+and+Use+of+BBs+-+Dynamic+Discovery+In+AS4+Gateways	Website describing further the Dynamic Discovery in AS4 Gateways
[REF9]	http://wiki.ds.unipi.gr/display/ESENS/PR+-+SMP+-+1.7.0	Space describing the SMP (Service Metadata Publisher)
[REF10]	http://wiki.ds.unipi.gr/display/ESENS/PR+-+AS4+-+1.11	e-SENS AS4 Profile 1.11

2. CONVENTIONS

The Commands and Configuration files listed in this document usually contain a mix of reserved words (commands, instructions and system related special words) and user defined words (chosen by the user) as well as comments and preferred values for certain variables. The conventions used in this document, to distinguish between them, are the followings:

- To keep this document release agnostic as much as possible, the strings "x-y-z" or "x.y.z" are intended to refer to the version of Domibus discussed in this version of the document, in the present case "Domibus 3.2.5".
- **Bold** is used for "reserved" words and commands
- *Normal italic* together with a short description of the argument, is used for user-defined names (chosen by yourself to designate items like users, passwords, database etc..). Normally contains at least 2 words separated by "_".
- ***Bold and Italic*** is used for advisable values which can be changed by the user depending on their infrastructure.
- Comments are sometimes added to describe the purpose of the commands, usually enclosed in brackets ().

By default, non-OS specific paths will be described using Linux patterns.

2.1. Example 1: Sample Oracle Statement

create user *edelivery_user* **identified by** *edelivery_password*;

grant all privileges to *edelivery_user*;

(Where *edelivery_user* and *edelivery_password* are names chosen by the user)

2.2. Example 2: Sample Configuration file

jdbc.datasource.0.driver.name=com.mysql.jdbc.Driver

jdbc.datasource.0.driver.url=jdbc:mysql://*localhost:3306*/*domibus_schema**

jdbc.datasource.0.driver.password=*edelivery_password*

jdbc.datasource.0.driver.username=*edelivery_user*

(Where:

- *edelivery_user, domibus_schema* and *edelivery_password* are names chosen by the user.

- ***localhost:3306*** represents hostname:port parameters of the MySQL database.)

3. PREREQUISITES

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

- Java runtime environnement (JRE), version 7 or 8:
  http://www.oracle.com/technetwork/java/javase/downloads/index.html
- One of the supported Database Management Systems :
  - MySQL 5,6 or above
  - Oracle 10g+
- If you don't plan to deploy Domibus according to the Pre-Configured Single Server Deployment method, you must also install one of the supported application servers unless you are intending:
  - WebLogic 12c
  - WildFly 9
  - Apache Tomcat 8.0.x
- All Domibus 3.2.5 installation resources, including full distributions and documentation can be found on the Single Web Portal :

  https://ec.europa.eu/cefdigital/wiki/x/DCKtAg

3.1. Binaries repository

All the Domibus 3.2.5 artefacts can be directly download from the Nexus repository of CEF (cf.[REF1]).

4. DOMIBUS DEPLOYMENT

Remark:

*The variable **cef_edelivery_path** refering to the folder where the package is installed will be used later in this document.*

4.1. Database Configuration

For this step you will have to use the following resources (see section §3.1 - "*Binaries repository*" for the download location):

- **domibus-MSH-X.Y.Z-sql-scripts.zip**

A datasource must be configured to allow the application to access its Database.

4.1.1. MySQL configuration

1. Unzip **domibus-MSH-X.Y.Z-sql-scripts.zip** in *cef_edelivery_path***/sql-scripts**
2. Open a command prompt and navigate to this directory: *cef_edelivery_path***/sql-scripts**.
3. (Optional) Storing messages in a database with payloads over 30 MB.

   Domibus temporarily stores the messages in the database. They are not deleted before they are successfully transferred to the final recipient (see §6.3 – "*PMode Configuration*"). Therefore, it is required to increase the maximum allowed size of packets. Update the default properties of **my.ini** (Windows) or **my.cnf** (Linux).

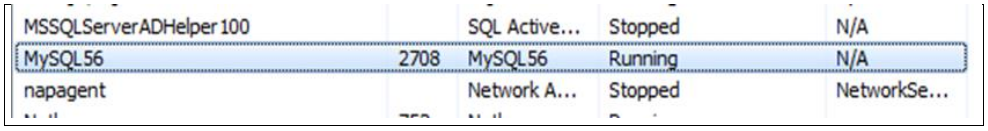
   - **max_allowed_packet** property

```
# The maximum size of one packet or any generated or intermediate string, or any
# parameter sent by the
# mysql_stmt_send_long_data() C API function.
max_allowed_packet=512M
```

   - **innodb_log_file_size** property

```
# Size of each log file in a log group. You should set the combined size
# of log files to about 25%-100% of your buffer pool size to avoid
# unneeded buffer pool flush activity on log file overwrite. However,# note that larger logfile size will increase the time needed for the recovery process
innodb_log_file_size=5120M
```

   - Restart MySQL service (Windows):



MySQL service

4. (Optional) For storing messages in a file system instead of a database see §5.2 – "*Domibus Properties)*

5. Execute the following MySQL commands at the command prompt :

Remark:

User defined names like root_password, domibus_schema etc..., are in italic as described in the Convention section.

```
mysql -h localhost -u root_user --password=root_password -e "drop schema if exists domibus_schema;create schema domibus_schema;alter database domibus_schema charset=utf8 collate=utf8_bin; create user edelivery_user@localhost identified by 'edelivery_password';grant all on domibus_schema.* to edelivery_user@localhost;"
```

The above creates a schema (*domibus_schema*) and a user (*edelivery_user*) having all the privileges on the schema**.**

```
mysql -h localhost -u root_user --password=root_password domibus_schema < mysql5innoDb-x.y.z.ddl
```

The above creates the required tables in *domibus_schema.*

Remark:

If you are using Windows, make sure to have the parent directory of mysql.exe added to your PATH variable.

4.1.2. Oracle configuration

1. Unzip **domibus-MSH-X.Y.Z-sql-scripts.zip** in *cef_edelivery_path***/sql-scripts**

2. Open a command prompt and navigate to this directory: *cef_edelivery_path***/sql-scripts**.

3. Open a command line session, log in and execute the following commands :

```
sqlplus sys as sysdba   (password should be the one assigned during the Oracle installation )
================================================================================
Once logged in Oracle:
create user edelivery_user identified by edelivery_password;
grant all privileges to edelivery_user;
connect edelivery_user
show user;  (should return : edelivery_user)
@oracle10g-x.y.z.ddl (run the scripts with the @ sign from the location of the scripts)
exit
================================================================================
```

4.2. Domibus on WebLogic 12.1.3

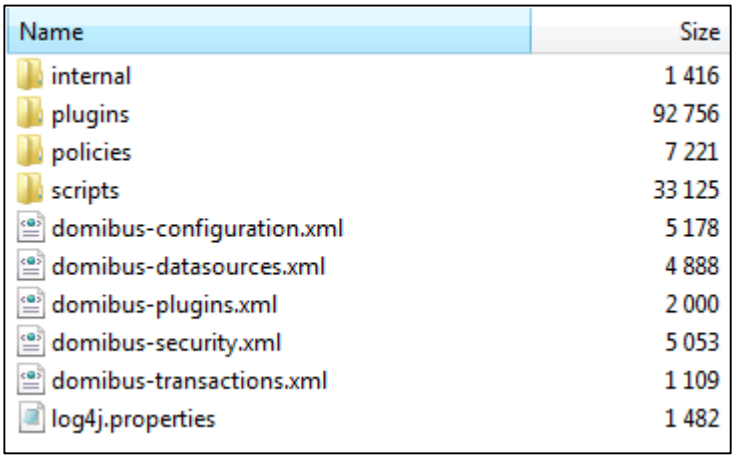
This section does not include the installation of WebLogic server 12.1.3. It is assumed that the WebLogic Server is installed and a Domain is created.
Hereafter the domain location will be referred as *DOMAIN_HOME (user defined name)*.

4.2.1. Single Server Deployment

For this step, you will have to use the following resources (see section §3.1 – "*Binaries repository*" for the download location):

- **domibus-MSH-X.Y.Z-weblogic.war**
- **domibus-MSH-X.Y.Z-weblogic-configuration.zip**
- **domibus-MSH-X.Y.Z-default-ws-plugin.zip (optional)**
- **domibus-MSH-X.Y.Z-default-jms-plugin.zip (optional)**

1. Download and unzip **domibus-MSH- X.Y.Z-weblogic-configuration.zip** in the directory *DOMAIN_HOME***/conf/domibus**



2. Download the **domibus-distribution- X.Y.Z-weblogic.war** in the directory *DOMAIN_HOME***/conf/domibus**

3. Configure your Keystore based on section §5.1.2 – "*Certificates*"

4. Add the following lines in:

   - For Windows : *DOMAIN_HOME***\bin\setDomainEnv.cmd**

- Locate the set DOMAIN_HOME statement and add the following lines after:

```
…
set DOMAIN_HOME
# Added for Domibus ************************************************************
set EXTRA_JAVA_PROPERTIES="%EXTRA_JAVA_PROPERTIES% -
Ddomibus.config.location=%DOMAIN_HOME%/conf/domibus"
# ******************************************************************************
…
```

- For Linux : *DOMAIN_HOME***/bin/setDomainEnv.sh**

- Locate the **export DOMAIN_HOME** statement and add the following lines after:

```
…
export DOMAIN_HOME
# Added for Domibus ************************************************************************
EXTRA_JAVA_PROPERTIES="$EXTRA_JAVA_PROPERTIES -
Ddomibus.config.location=$DOMAIN_HOME/conf/domibus"
export EXTRA_JAVA_PROPERTIES
# ******************************************************************************************
…
```

5. Run the WebLogic Scripting Tool (WLST) in order to create the JMS resources and the Database datasources from the command line

   - Download the WLST Package from the following location: https://ec.europa.eu/cefdigital/artifact/content/repositories/eDelivery/eu/europa/ec/digit/ipcis/wslt-api/1.9.1/wslt-api-1.9.1.zip

   - Configure the WSLT API tool

     - Unzip the **wslt-api-1.9.1.zip**

     - Define the **WL_HOME** as a system environment variable to point to the WebLogic 'wlserver' directory as defined in the **DOMAIN_HOME/bin/SetDomainEnv.[cmd|sh]**

       e.g. WL_HOME=/wls12130/wlserver

   - Take the script **WeblogicSingleServer.properties** from **domibus-distribution-X.Y.Z-weblogic-configuration.zip** under the scripts directory and copy the **WeblogicSingleServer.properties** file into the **wslt-api-1.9.1** directory and adapt the following properties :

     - Adapt the properties for connecting to the WebLogic domain

```
domain.loading.type=connect
domain.connect.url=t3://localhost:7001
domain.connect.username=weblogic_name
domain.connect.password=weblogic_password
domain.name=my_domain1
```

     - Adapt the jdbc.datasource properties for the datasources

       For Oracle database:

```
jdbc.datasource.0.name=eDeliveryDs
jdbc.datasource.0.driver.name=oracle.jdbc.xa.client.OracleXADataSource
jdbc.datasource.0.driver.url=jdbc:oracle:thin:@127.0.0.1:1521:xe
jdbc.datasource.0.driver.password=edelivery_password
jdbc.datasource.0.driver.username=edelivery_username
```

Remark:

MySQL configuration is commented by default. To enable MySQL, remove the comment (#) from the lines below. Don't forget to add the comment (#) for Oracle to disable it.

For MySQL:

```
jdbc.datasource.0.driver.name=com.mysql.jdbc.Driver
jdbc.datasource.0.driver.url=jdbc:mysql://localhost:3306/domibus_schema
jdbc.datasource.0.driver.password=edelivery_password
jdbc.datasource.0.driver.username=edelivery_username
jdbc.datasource.0.transaction.protocol=LoggingLastResource
jdbc.datasource.0.pool.connection.test.onreserv.sql=SQL SELECT 1
```

- Adapt the property for location of the filestore
  persistent.filestore.0.location

  e.g.

```
persistent.filestore.0.location=DOMAIN_HOME/filestore
```

Remark:

Make sure that the path for the filestore contains forward slashes (/).

- Adapt if necessary the JMX security configuration

  e.g.

```
#####################################################################
## Policy configuration
#####################################################################
security.policies.0.mode = CREATE
security.policies.0.resource = type=<jmx>, operation=invoke,
application=,
mbeanType=weblogic.management.runtime.JMSDestinationRuntimeMBean
security.policies.0.realm = myrealm
security.policies.0.authorizer = XACMLAuthorizer
security.policies.0.expression=
Rol(Admin)|Grp(Administrators)|Grp(JMSManagers)
security.policies.items = 1
#####################################################################
## Users configuration
#####################################################################
security.users.0.realm=myrealm
security.users.0.name=jmsManager
security.users.0.password=jms_Manager1
security.users.0.comment=
security.users.0.authenticator=DefaultAuthenticator
security.users.items=1
#####################################################################
## Groups configuration
#####################################################################
security.groups.0.realm=myrealm
security.groups.0.name=JMSManagers
security.groups.0.description=
security.groups.0.authenticator=DefaultAuthenticator
security.groups.items=1
#####################################################################
## Groups Membership configuration
#####################################################################
security.group.member.0.user=jmsManager
security.group.member.0.groups=JMSManagers
security.group.member.0.realm=myrealm
security.group.member.0.authenticator=DefaultAuthenticator
security.group.member.items=1
```

- Start the WebLogic domain from within *DOMAIN_HOME*
  - For Windows

    startWebLogic.cmd
  - For Linux

    startWebLogic.sh
- Execute the following command from within the **wlstapi-1.9.1/bin** directory
  - For Windows

    wlstapi.cmd ..\scripts\import.py --property ..\WeblogicSingleServer.properties
  - For Linux

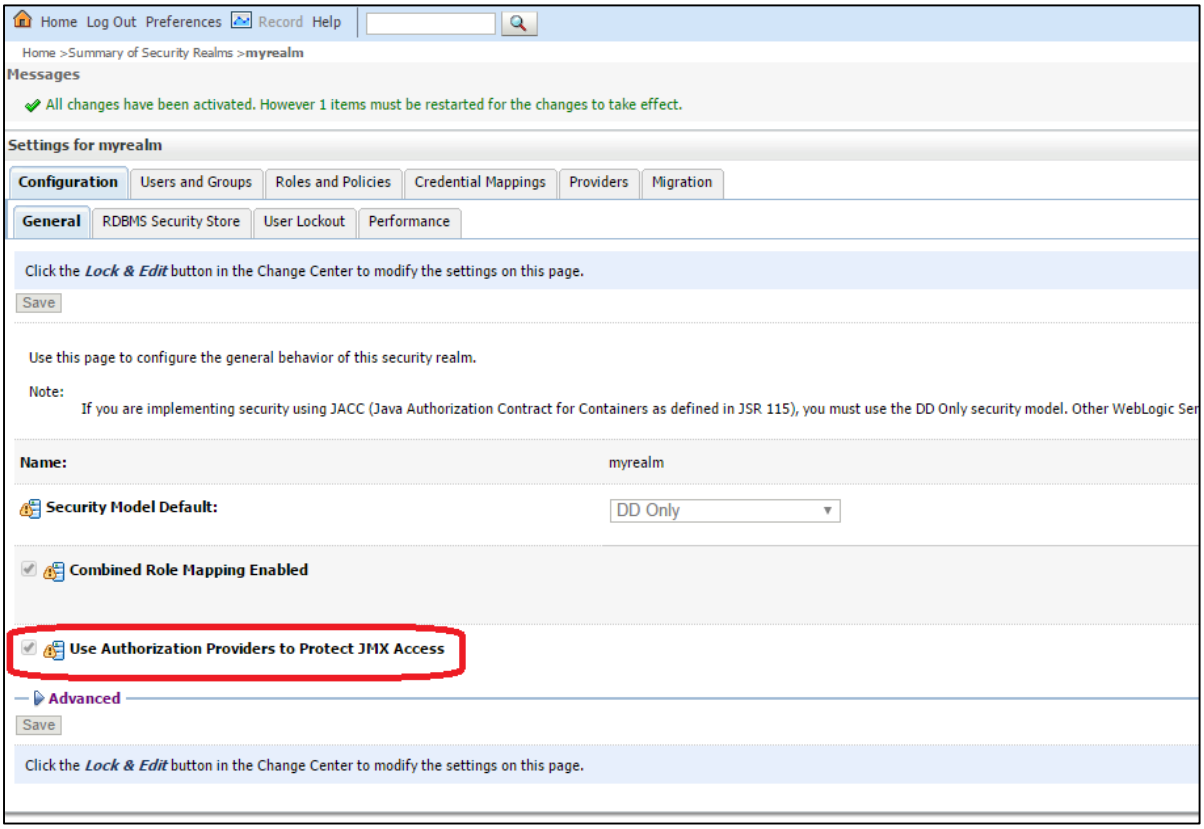
    wlstapi.sh ../scripts/import.py --property ../WeblogicSingleServer.properties

Expected Result:

```
Saving all your changes ...
Saved all your changes successfully.
Activating all your changes, this may take a while ...
The edit lock associated with this edit session is released
once the activation is completed.
Activation completed
Location changed to serverRuntime tree. This is a read-only tree with DomainMBean as the root.
For more help, use help('domainConfig')

Disconnected from weblogic server: AdminServer
```

6. Activate the use of the authorization providers to protect the JMX access



7. The database dialect is pre-configured to use the Oracle database. If you are using a MySQL database, you should adapt the dialect in *DOMAIN_HOME***/conf/domibus/domibus-datasources.xml** as highlighted in the example below:

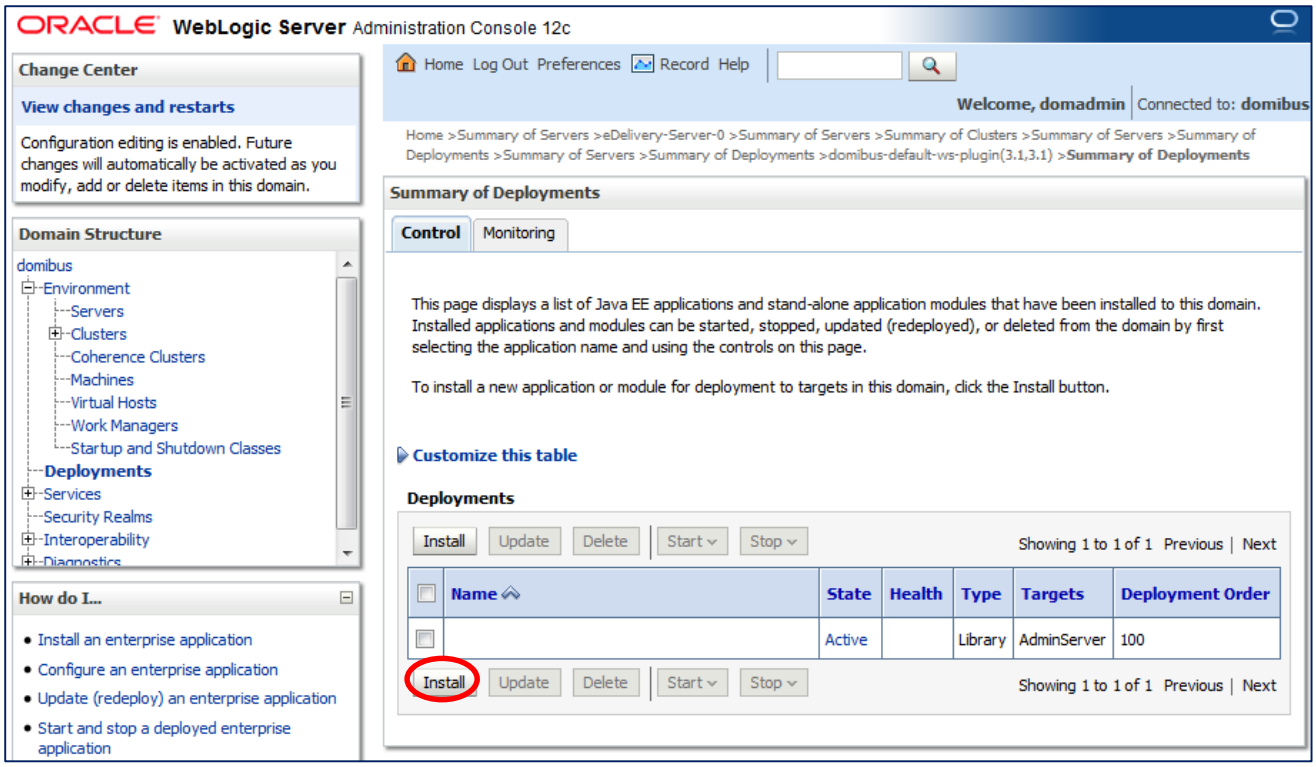
```
../
<property name="jpaVendorAdapter">
```

```
    <bean
class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter">
      <property name="showSql" value="false"/>
      <property name="generateDdl" value="false"/>
      <property name="databasePlatform"
value="org.hibernate.dialect.MySQL5InnoDBDialect"/>
    </bean>
   </property>
        <property name="jpaProperties">
     <props>
      <prop
key="hibernate.connection.driver_class">com.mysql.jdbc.Driver</prop>
      <prop
key="hibernate.dialect">org.hibernate.dialect.MySQL5InnoDBDialect</prop>
      <prop key="hibernate.format_sql">true</prop>
      <prop
key="transaction.factory class">org.hibernate.engine.transaction.internal.j
ta.CMTTransactionFactory</prop>
          <prop
key="hibernate.transaction.manager_lookup_class">org.hibernate.transaction.
WeblogicTransactionManagerLookup</prop>
                <prop
key="hibernate.transaction.jta.platform">org.hibernate.service.jta.platform
.internal.WeblogicJtaPlatform</prop>
    </props>
   </property>
/..
```

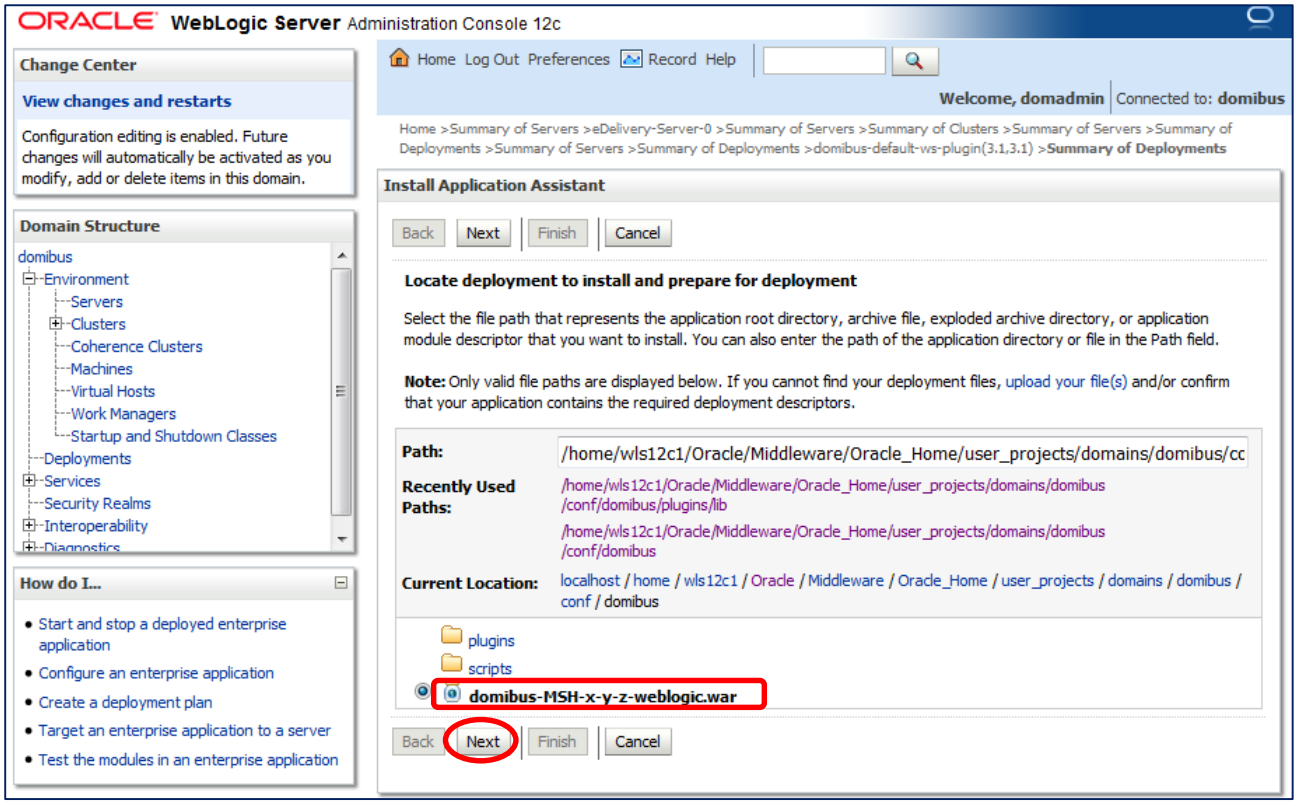
8. Install the WS Plugin. For more details (see section §6.2.1.2 – "*WebLogic*").

9. Deploy **domibus-MSH-X.Y.Z-weblogic.war**

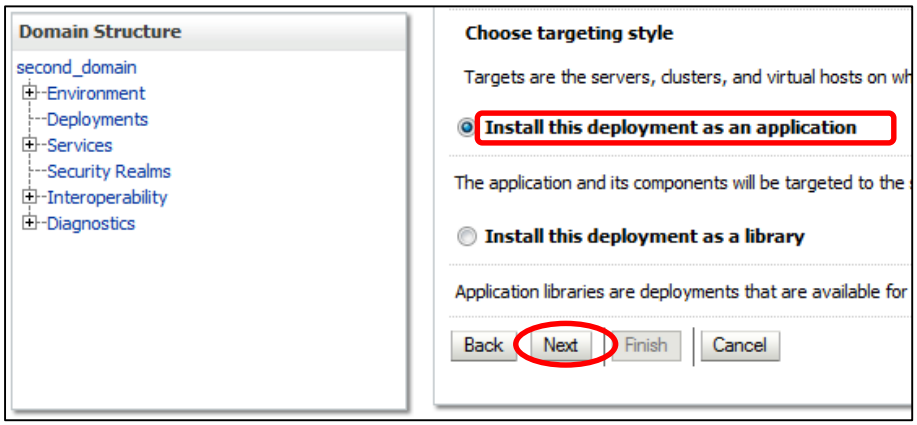
- Click **Install**



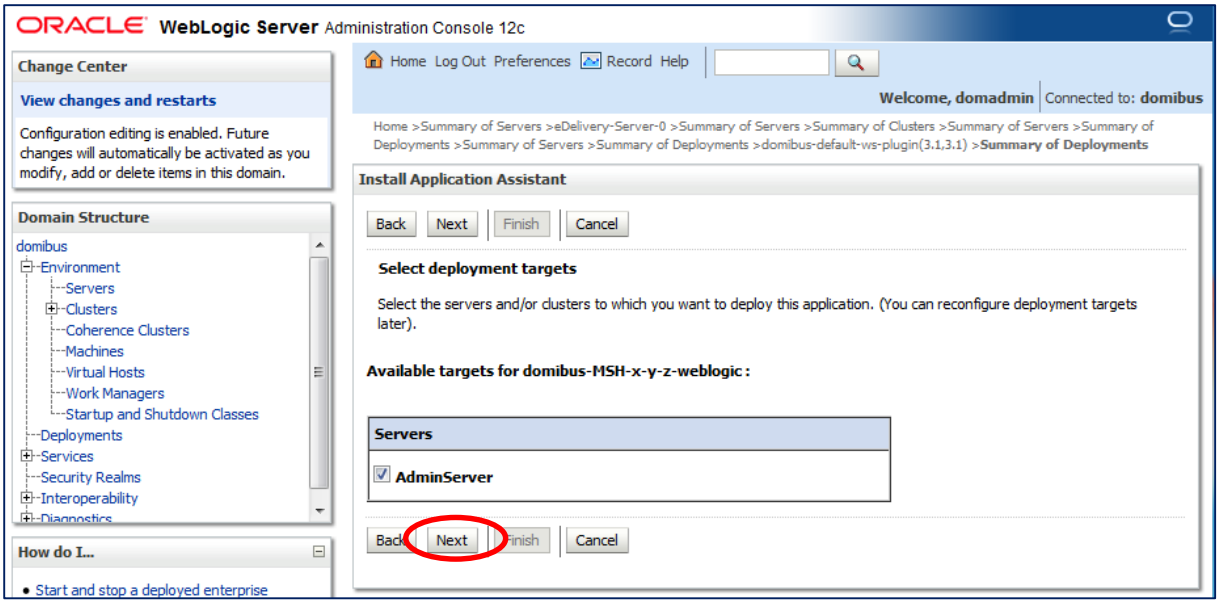
- Navigate to the location of the **.war** file and click **Next**



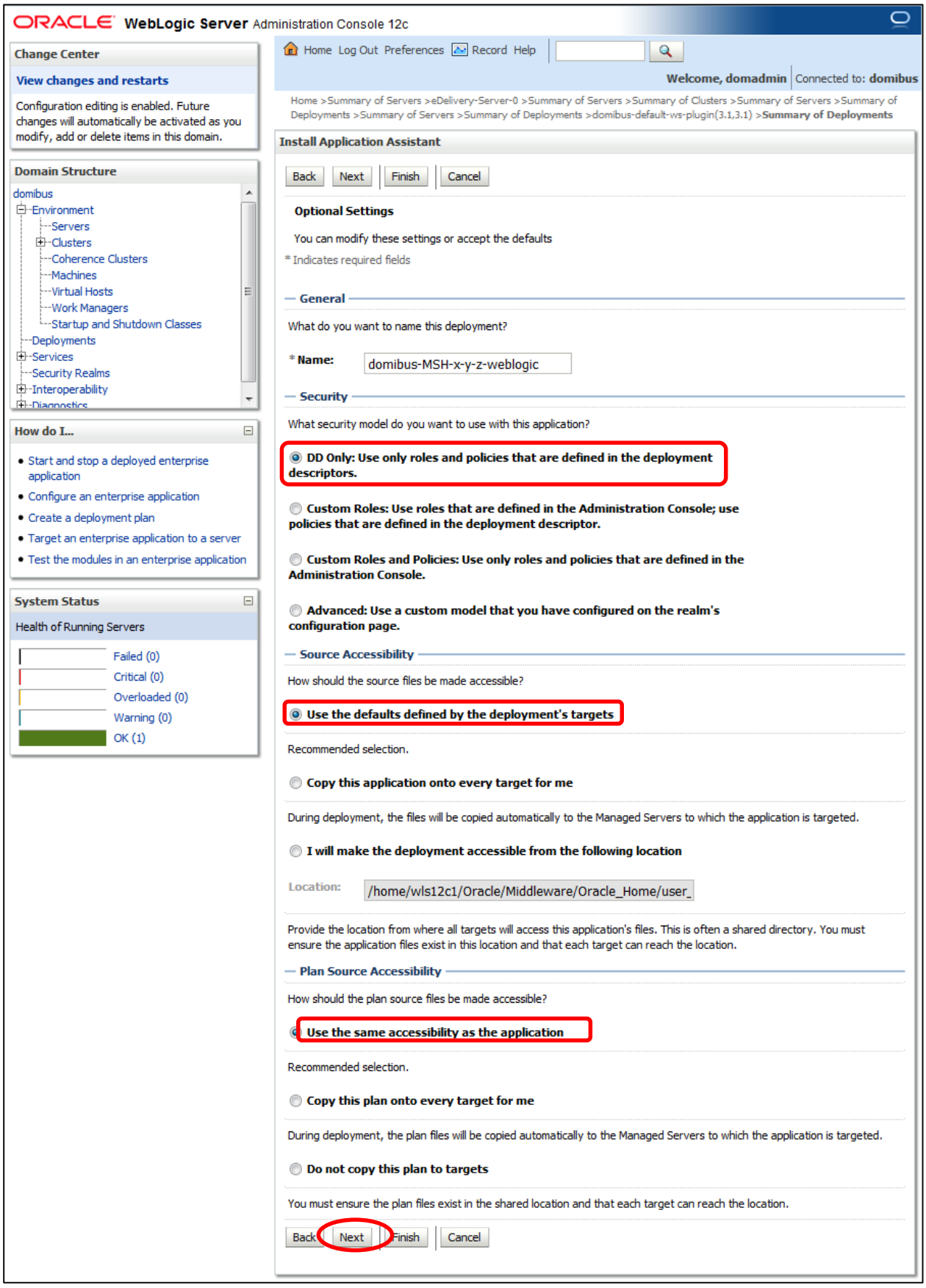
- Choose **Install this deployment as an application** and click **Next**



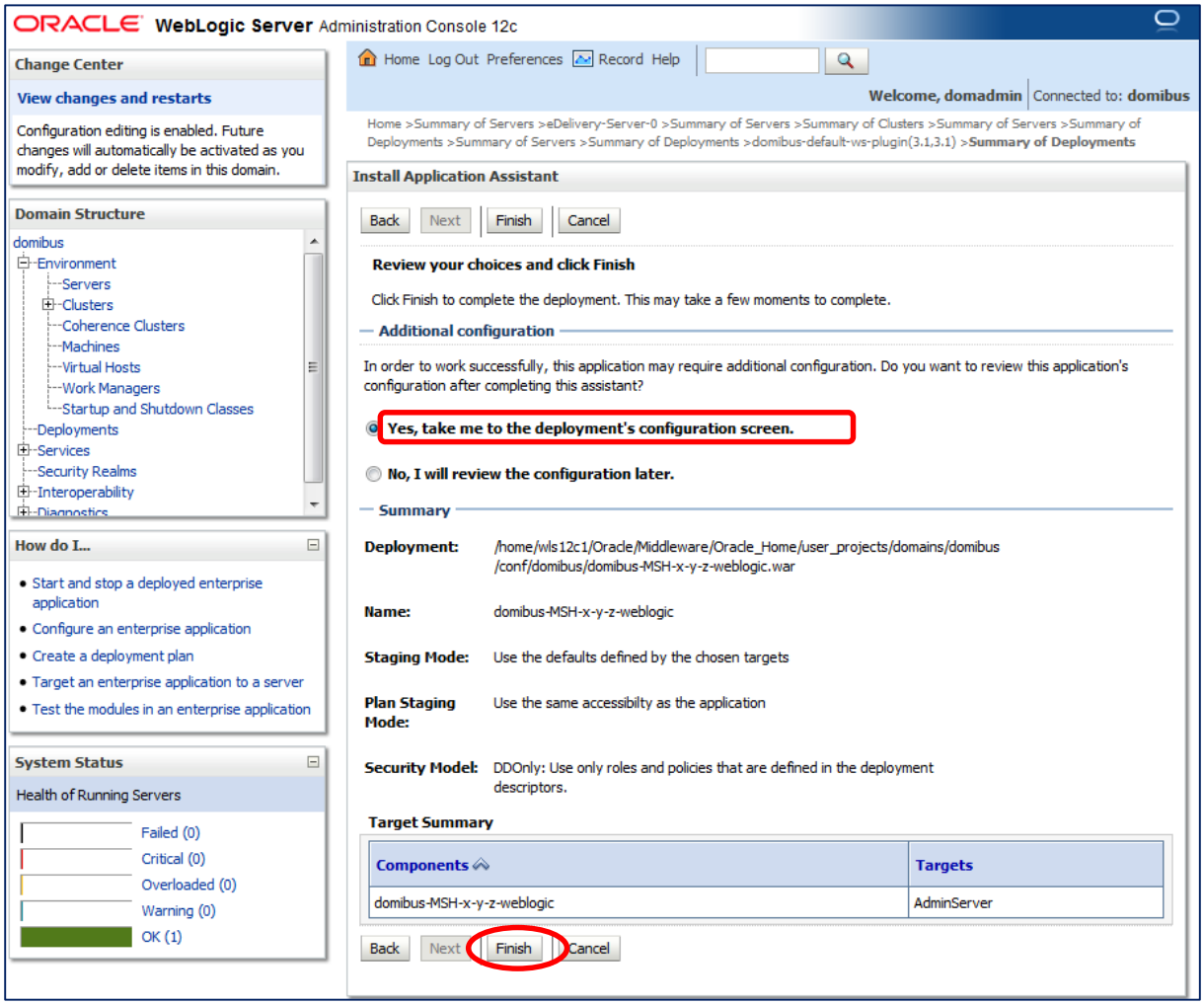
- Select the following option and click **Next**



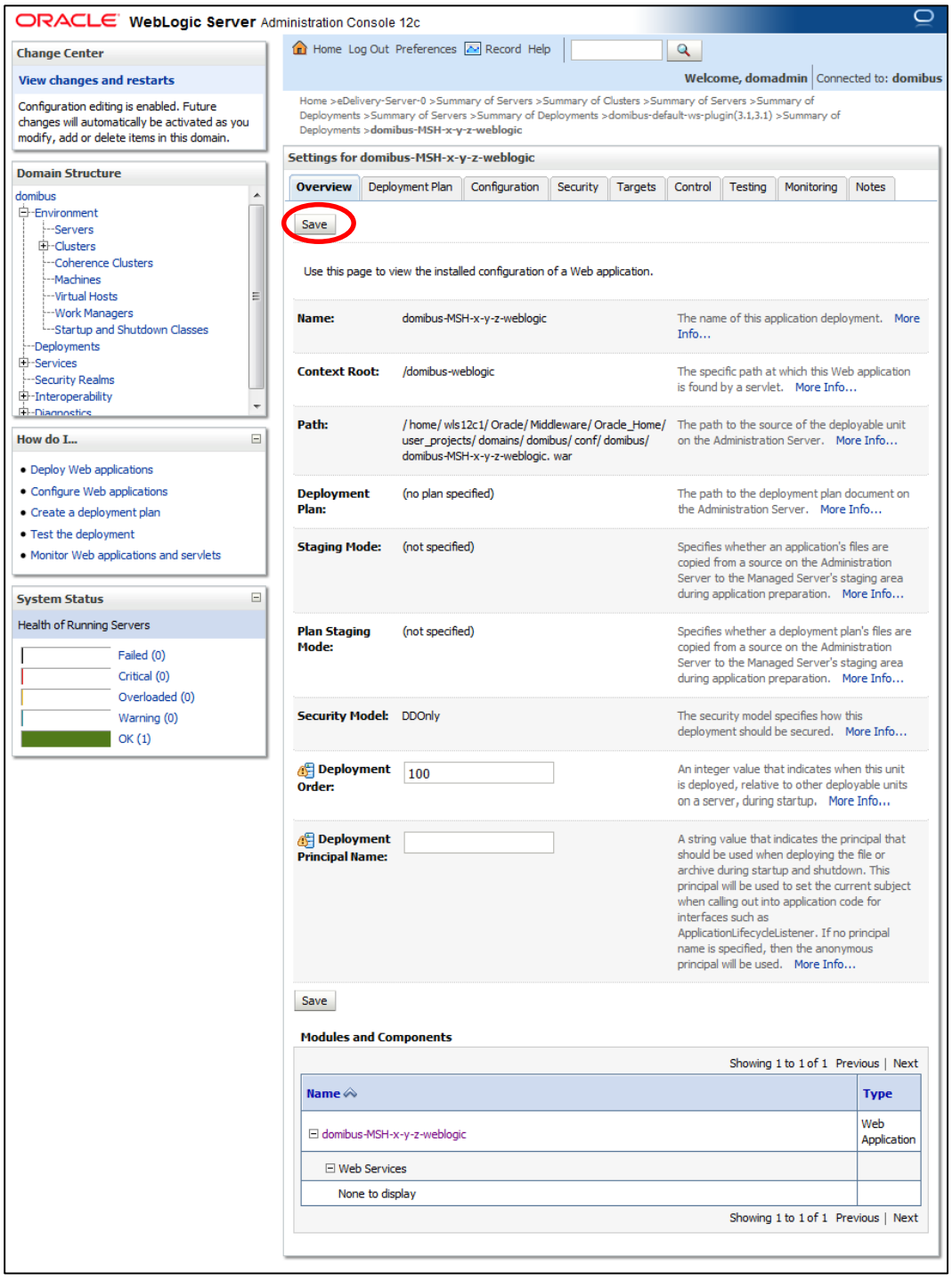
- Select the following options and click **Next**



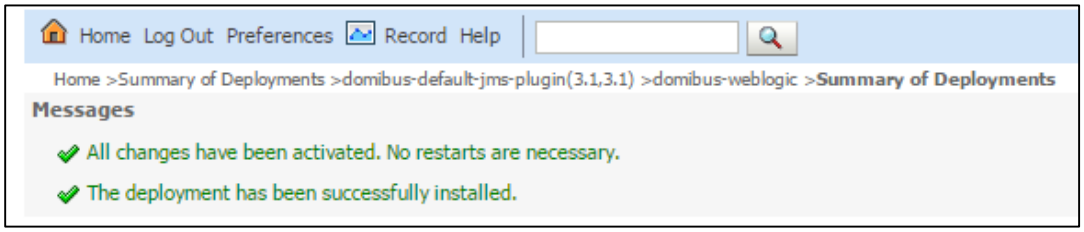
- Select the following option and click **Finish**



- Here is an overview of the resulting settings, you can now click **Save**



The expected positive response to the deployment request should be the following:



10. Verify the installation by navigating into your browser to http://*localhost*:*7001*/domibus-weblogic/home

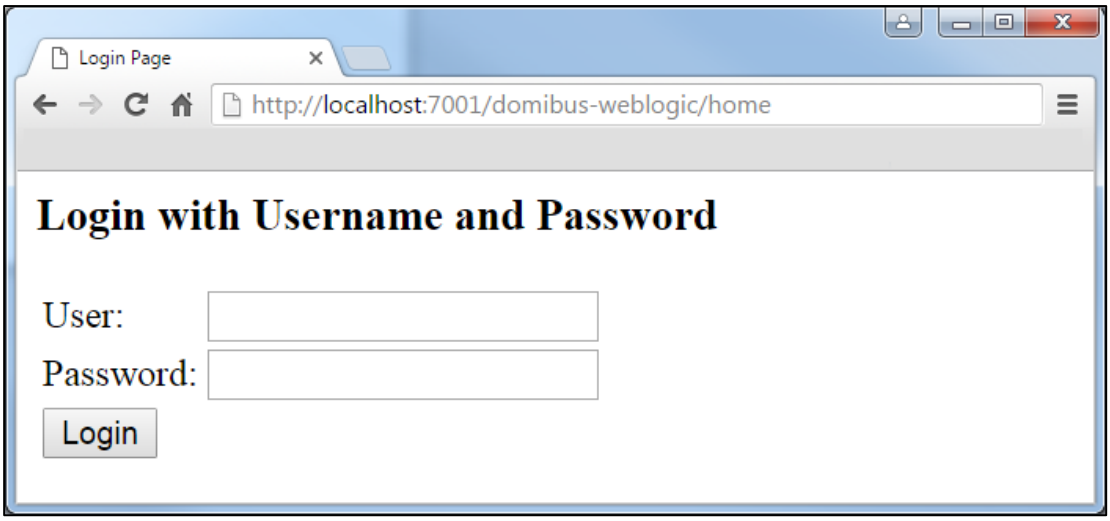
    If you can access the page it means the deployment was successful.

    (by default: User = ***admin***; Password = ***123456)***

    Remark:

    > *It is recommended to change the passwords for the default users. See §6.4.1 –* "Administration Dashboard" *for further information.*

Expected result:



4.2.2. Clustered Deployment

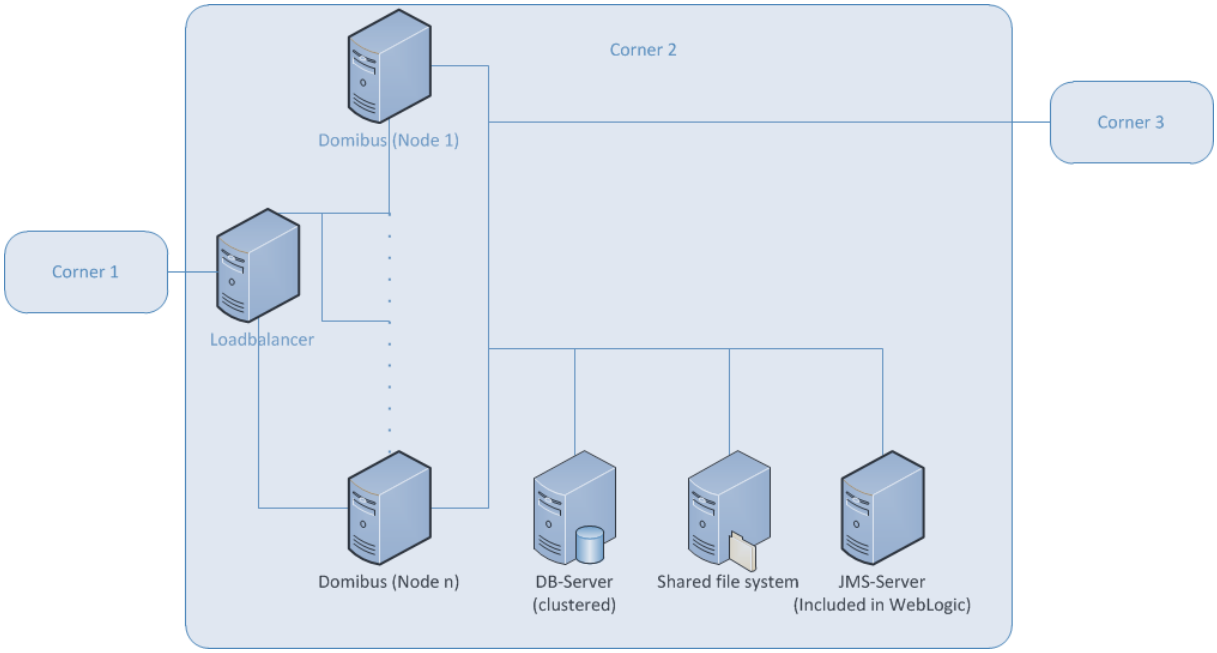


Figure 1 - Diagram representing the Deployment of Domibus in a Cluster on WebLogic

Remark:

In this section we assume that a Domain and a WebLogic Cluster is already setup.

For this step, you will have to use the following resources (see section §3.1 – "*Binaries repository*" for the download location):

- **domibus-MSH-X.Y.Z-weblogic.war**
- **domibus-MSH-X.Y.Z-weblogic-configuration.zip**
- **domibus-MSH-X.Y.Z-default-ws-plugin.zip (optional)**
- **domibus-MSH-X.Y.Z-default-jms-plugin.zip (optional)**

1. Follow steps **1**, **2**, **3** and **4** from §4.2.1 - "*Single Server Deployment*".

2. Run the WebLogic Scripting Tool (WLST) in order to create the necessary JMS resources and Database datasources from the command line

    - Download the WLST Package from the following location: https://ec.europa.eu/cefdigital/artifact/content/repositories/eDelivery/eu/europa/ec/digit/ipcis/wslt-api/1.9.1/wslt-api-1.9.1.zip

    - Configure the WSLT API tool

        - Unzip the **wslt-api-1.9.1.zip**

        - Define the WL_HOME (SET or export command depending on your operating system) environment variable to point to the WebLogic **wlserver** directory

          e.g. WL_HOME=/wls12130/wlserver

- Take the script **WeblogicCluster.properties** from **domibus-distribution-X.Y.Z-weblogic-configuration.zip** under the scripts directory and copy the **WeblogicCluster.properties** file into the **wslt-api-1.9.1** directory and apply the following changes :

  - Adapt the properties for connecting to the WebLogic domain

```
domain.loading.type=connect
domain.connect.url=t3://localhost:7001
domain.connect.username=weblogic_user
domain.connect.password=weblogic_password
domain.name=mydomain1
```

  - Adapt the jdbc.datasource properties for the datasources

For Oracle database:

```
jdbc.datasource.0.name= eDeliveryDs
jdbc.datasource.0.driver.name=oracle.jdbc.xa.client.OracleXADataSource
jdbc.datasource.0.driver.url=jdbc:oracle:thin:@127.0.0.1:1521:xe
jdbc.datasource.0.driver.password=edelivery_password
jdbc.datasource.0.driver.username=edelivery_username
jdbc.datasource.0.targets=cluster_name
```

For MySQL database:

Remark:

MySQL configuration is commented by default. To enable MySQL, remove the comment (#) from the lines below. Don't forget to add the comment (#) for Oracle to disable it.

```
jdbc.datasource.0.name= eDeliveryDs
jdbc.datasource.0.driver.name=com.mysql.jdbc.Driver
jdbc.datasource.0.driver.url=jdbc:mysql://localhost:3306/domibus_schema
jdbc.datasource.0.driver.password=edelivery_password
jdbc.datasource.0.driver.username=edelivery_username
jdbc.datasource.0.targets=cluster_name
jdbc.datasource.0.transaction.protocol=LoggingLastResource
jdbc.datasource.0.pool.connection.test.onreserv.sql=SQL SELECT 1
```

- Adapt the properties for target and location of the filestore

```
persistent.filestore.0.target=cluster_name
persistent.filestore.0.location=DOMAIN_HOME/filestores
```

Remark:

If you are using Windows, make sure that the path for the filestore content forward slash (/).

- Adapt if necessary the JMX security configuration

e.g.

```
#################################################################
## Policy configuration
#################################################################
security.policies.0.mode = CREATE
security.policies.0.resource = type=<jmx>, operation=invoke,
application=,
mbeanType=weblogic.management.runtime.JMSDestinationRuntimeMBean
security.policies.0.realm = myrealm
security.policies.0.authorizer = XACMLAuthorizer
security.policies.0.expression=
Rol(Admin)|Grp(Administrators)|Grp(JMSManagers)
security.policies.items = 1
#################################################################
## Users configuration
#################################################################
security.users.0.realm=myrealm
security.users.0.name=jmsManager
security.users.0.password=jms_Manager1
security.users.0.comment=
security.users.0.authenticator=DefaultAuthenticator
security.users.items=1
#################################################################
## Groups configuration
#################################################################
security.groups.0.realm=myrealm
security.groups.0.name=JMSManagers
security.groups.0.description=
security.groups.0.authenticator=DefaultAuthenticator
security.groups.items=1
#################################################################
## Groups Membership configuration
#################################################################
security.group.member.0.user=jmsManager
security.group.member.0.groups=JMSManagers
security.group.member.0.realm=myrealm
security.group.member.0.authenticator=DefaultAuthenticator
security.group.member.items=1
```

- Adapt the property for JMS Server

e.g.

```
jms.server.0.target=cluster_name
```

- Adapt the property for JMS Module

e.g.

```
jms.module.0.targets=cluster_name
```

- Start the WebLogic domain from within *DOMAIN_HOME*

  - For Windows

```
startWebLogic.cmd
```

  - For Linux

```
startWebLogic.sh
```

- Execute the following command from within the **wlstapi-1.9.1/bin** directory

  - For Windows

```
wlstapi.cmd ..\scripts\import.py --
property ..\WeblogicCluster.properties
```

  - For Linux

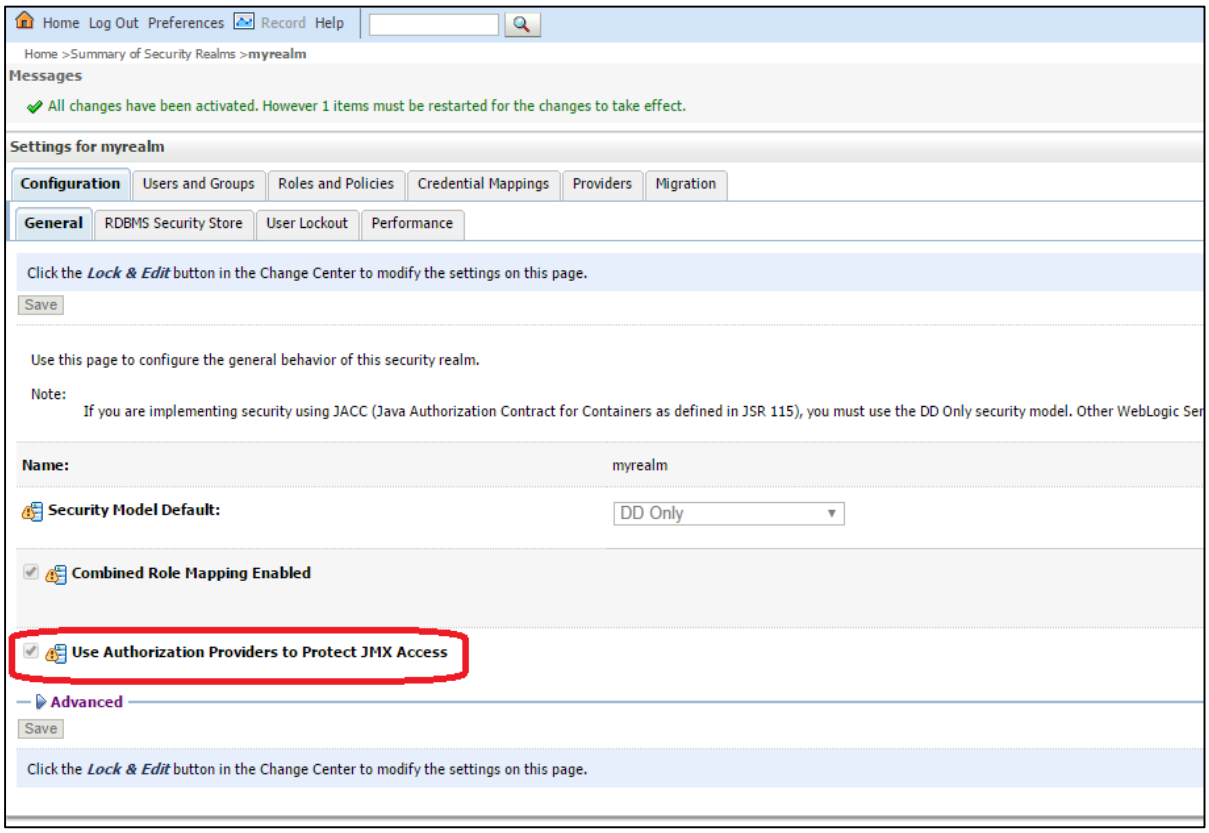
```
wlstapi.sh ../scripts/import.py --
property ../WeblogicCluster.properties
```

Expected Result:

```
Saving all your changes ...
Saved all your changes successfully.
Activating all your changes, this may take a while ...
The edit lock associated with this edit session is released
once the activation is completed.
Activation completed
Location changed to serverRuntime tree. This is a read-only tree with DomainMBean as the root.
For more help, use help('domainConfig')

Disconnected from weblogic server: AdminServer
```

3. Activate the use of the authorization providers to protect the JMX access



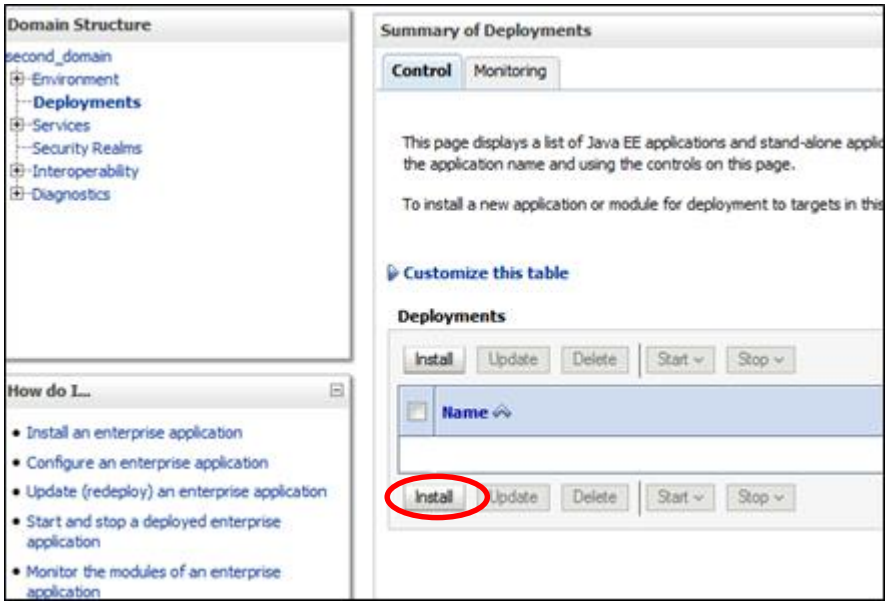
4. The database dialect is pre-configured to use the Oracle database. If you are using the MySQL database you should adapt the dialect as highlighted in the text below in *DOMAIN_HOME***/conf/domibus-datasources.xml** file :

```
../
<property name="jpaVendorAdapter">
            <bean class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter">
                <property name="showSql" value="false"/>
                <property name="generateDdl" value="false"/>
                <property name="databasePlatform"
value="org.hibernate.dialect.MySQL5InnoDBDialect"/>
            </bean>
        </property>
        <property name="jpaProperties">
            <props>
                <prop key="hibernate.connection.driver_class">com.mysql.jdbc.Driver</prop>
                <prop
key="hibernate.dialect">org.hibernate.dialect.MySQL5InnoDBDialect</prop>
                <prop key="hibernate.format_sql">true</prop>
                <prop
key="transaction.factory_class">org.hibernate.engine.transaction.internal.jta.CMTTransactio
nFactory</prop>
          <prop
key="hibernate.transaction.manager_lookup_class">org.hibernate.transaction.WeblogicTransact
ionManagerLookup</prop>
                <prop
key="hibernate.transaction.jta.platform">org.hibernate.service.jta.platform.internal.Weblog
icJtaPlatform</prop>
            </props>
        </property>
/..
```

5. Install the WS plugin. For more details refer to chapter §6.2.1.2 – "*WebLogic*",

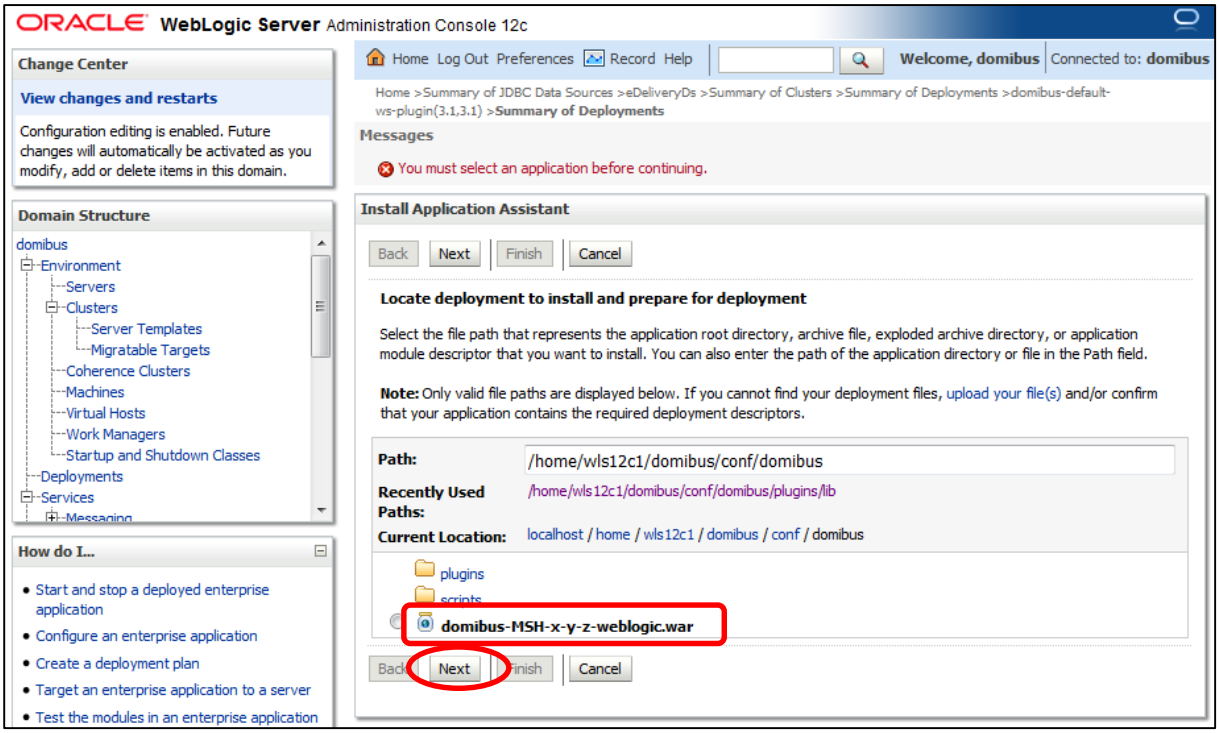
6. Deploy **domibus-MSH-X.Y.Z-weblogic.war**.

    - Click **Install**

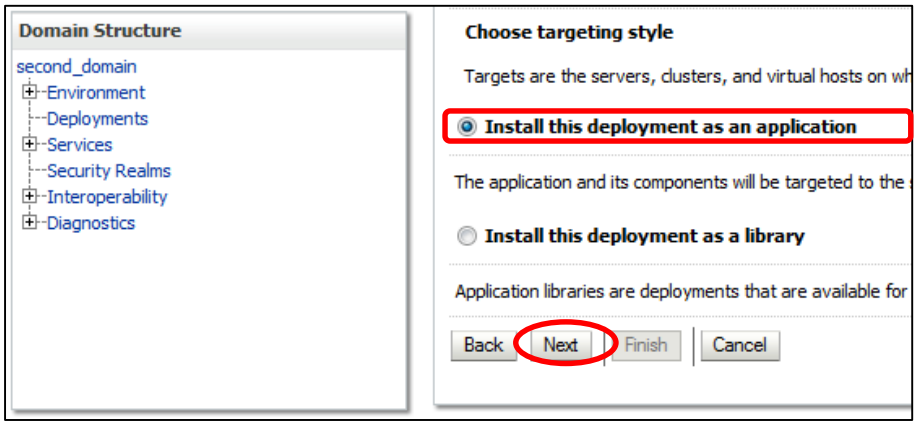


- Navigate to location DOMAIN_HOME**/conf/domibus** where the **domibus-MSH-X.Y.Z-weblogic.war** file has been previously copied

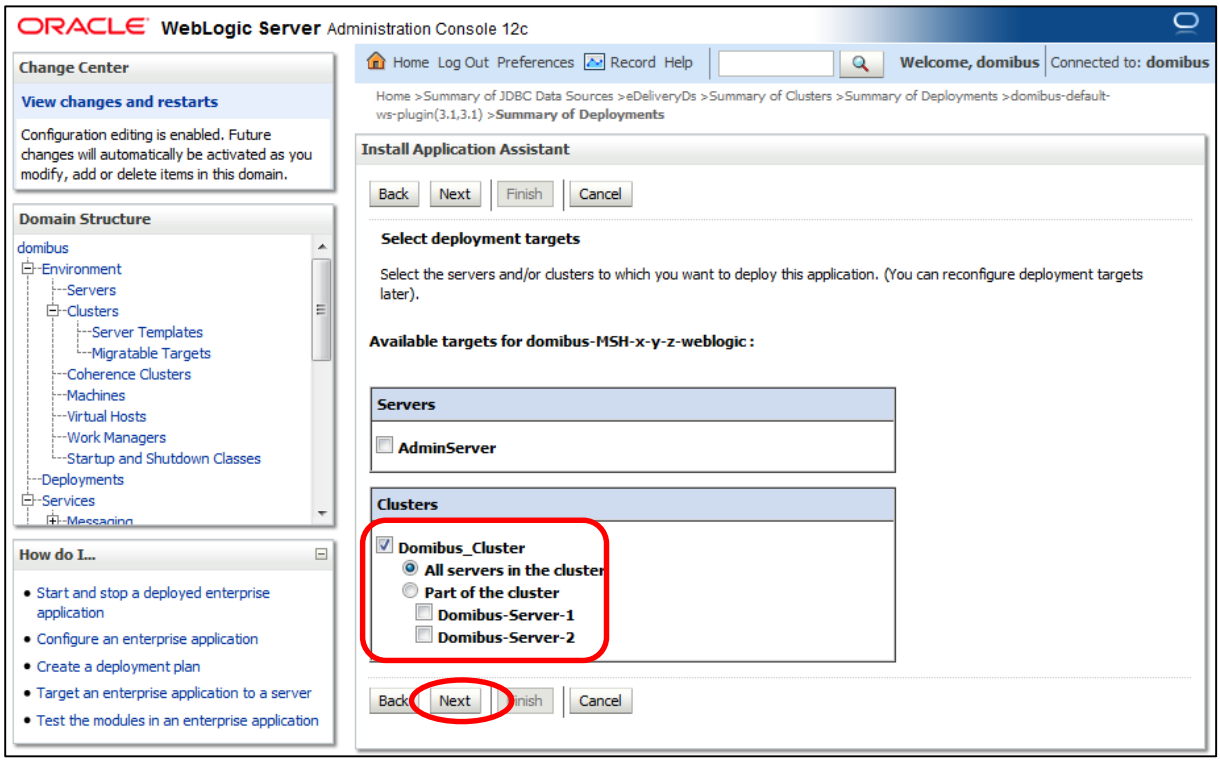
- Select the **domibus-MSH-X.Y.Z-weblogic.war** file and click **Next**



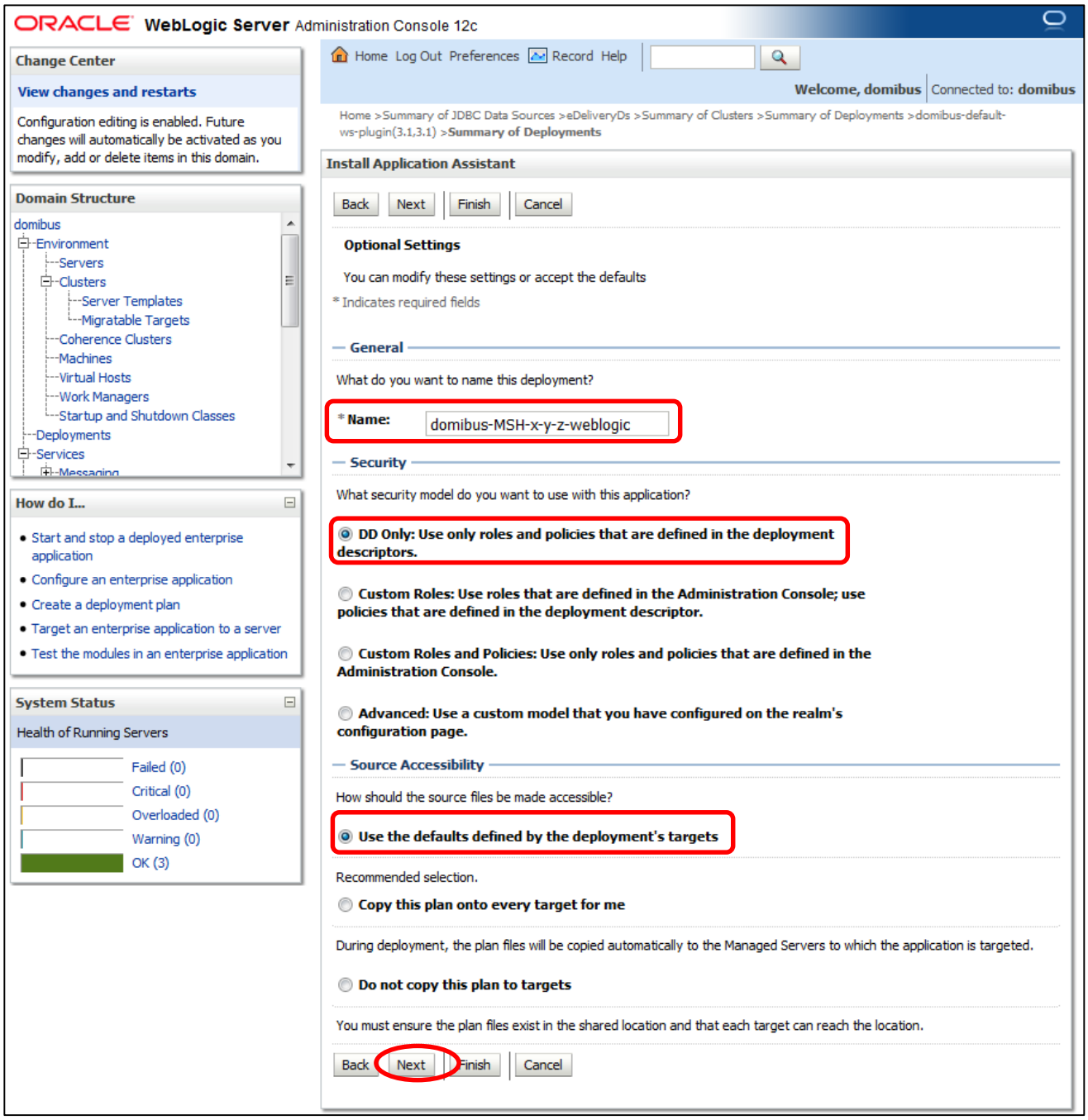
- Choose **Install this deployment as an application** and click **Next**



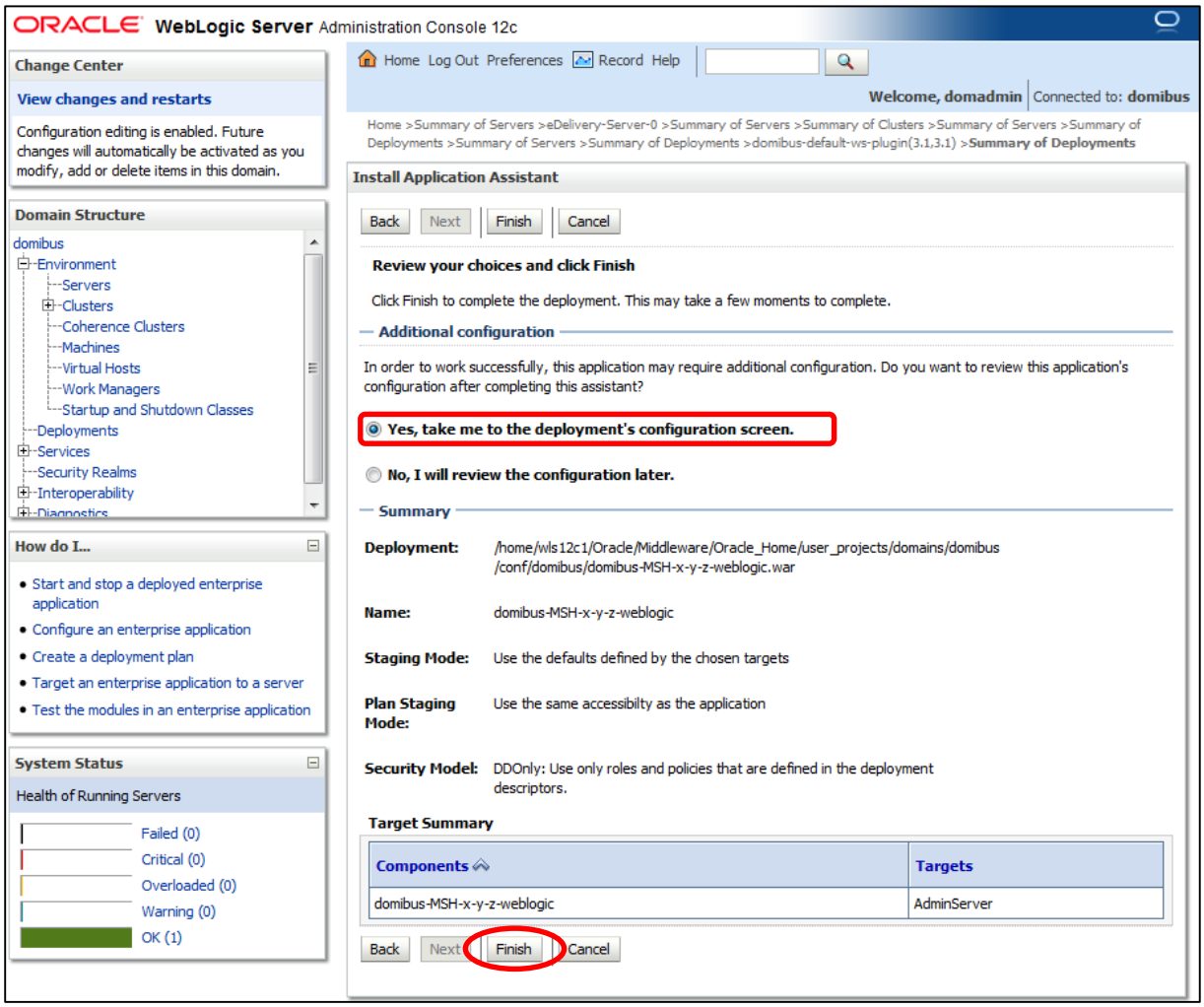
- Select your cluster for the deployment target and click **Next**



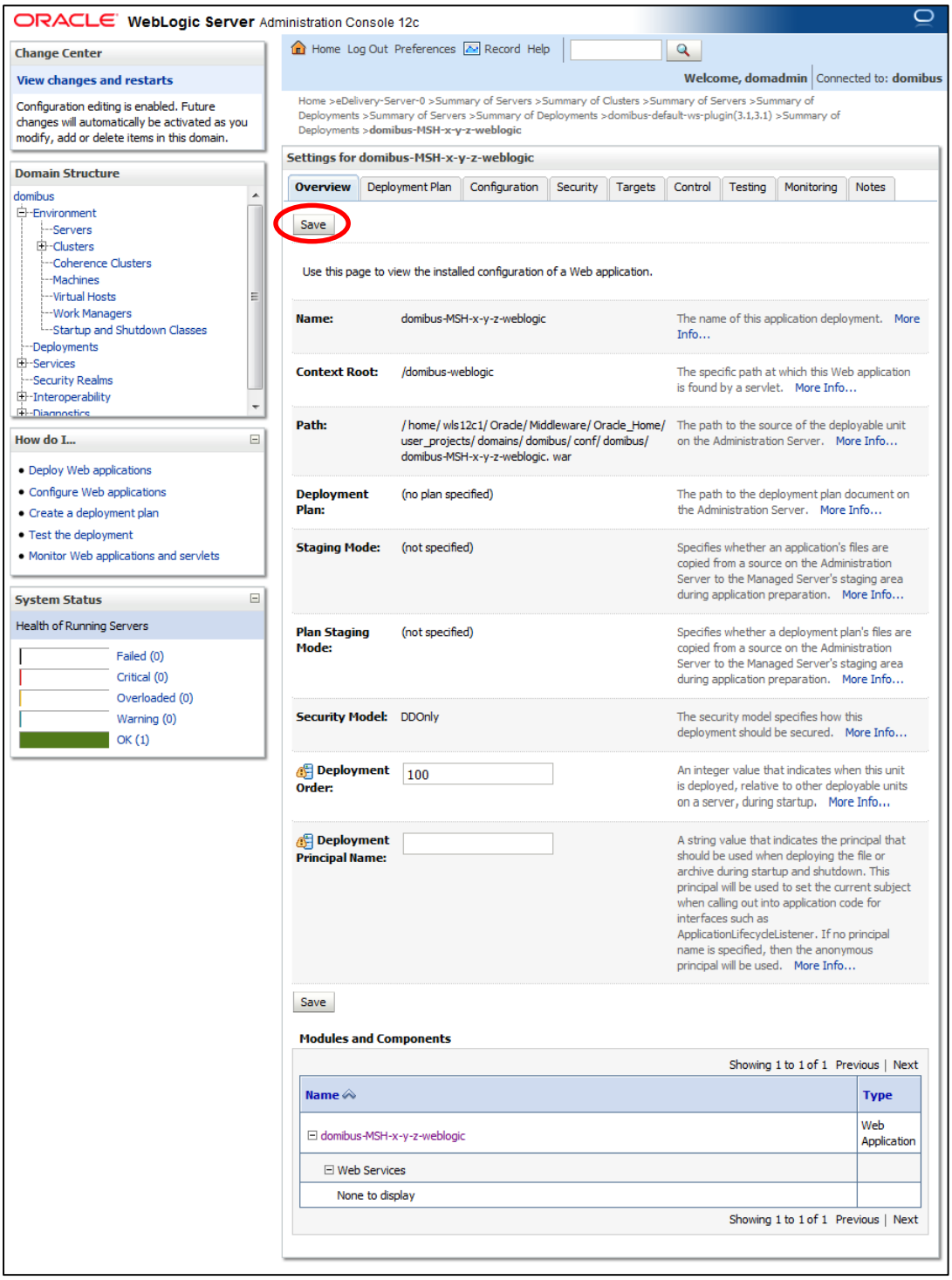
- Select the following options and click **Next**



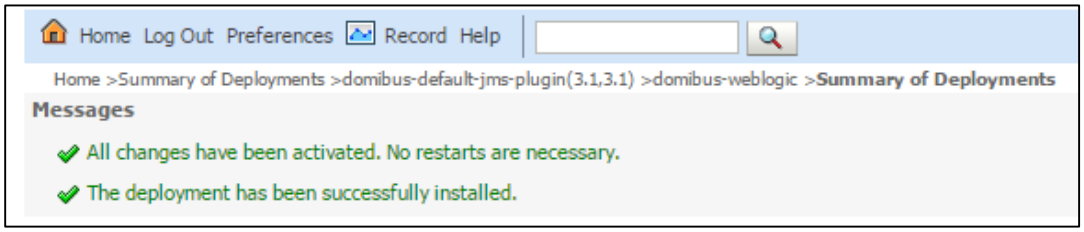
- Select the following option and click **Finish**



- Here is an overview of the resulting settings, you can now click **Save**



The expected positive response to the deployment request should be the following:



7. Verify the installation by navigating into your browser to http://*localhost*:*7001*/domibus-weblogic/home

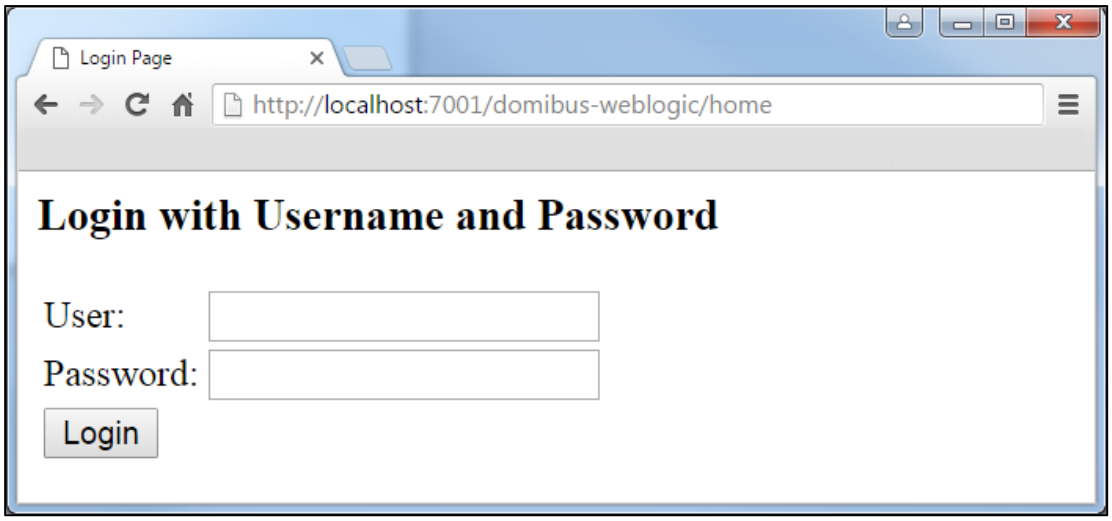
   If you can access the page it means the deployment was successful.

   (by default: User = ***admin***; Password = ***123456)***

   Remark:

   > *It is recommended to change the passwords for the default users. See §6.4.1 -* Administration Dashboard*" for further information.*

Expected result:



4.3. Domibus on Tomcat

Remark:

> *As Tomcat isn't a full Java EE application server and doesn't offer JMS capabilities by default, Domibus uses ActiveMQ as an in-memory JMS broker when deployed on a Tomcat servlet container. The configuration for the ActiveMQ JMS broker can be found in cef_edelivery_path/**domibus/internal/activemq.xml**.*

4.3.1. Pre-Configured Single Server Deployment

For this step, you will have to use the following resources (see section §3.1 – "*Binaries repository*" for the download location):

- **domibus-MSH-X.Y.Z-tomcat-full.zip**

1. Unzip the archive
    - Unzip **domibus-MSH-X.Y.Z-tomcat-full.zip** to a location on your physical machine: *cef_edelivery_path*.

2. Prepare the database
    - For MySQL database:

Add MySQL JDBC driver (available on MySQL official web site cf. [REF2]) in the folder *cef_edelivery_path***/domibus/lib**.

Remark:

> *The version of the JDBC driver has to be mysql-connector-java-5.1.40.jar or higher.*

Edit the xml file *cef_edelivery_path***/domibus/conf/domibus/domibus-datasources.xml** and adjust the highlighted parts in the text below according to your environment:

```
<bean id="domibusJDBC-XADataSource" …….>
<property name="uniqueResourceName">
            <value>domibusJDBC-XA</value>
</property>
<property name="xaDataSourceClassName">
<value>com.mysql.jdbc.jdbc2.optional.MysqlXADataSource</value>
</property>
<property name="xaProperties">
<props>
                <prop key="serverName">db_host</prop>
                <prop key="port">db_port</prop>
                <prop key="user">edelivery_user</prop>
                <prop key="password">edelivery_password</prop>
```

```
                <prop
key="url">jdbc:mysql://db_host:db_port/domibus_schema?pinGlobalTxToPhysicalConn
ection=true</prop>
</props>
</property>
……
<property name="testQuery">
<value>select 1</value>
</property>
</bean>

<bean id="entityManagerFactory" ……>
<property name="jpaVendorAdapter">
<bean class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter">
<property name="databasePlatform"
value="org.hibernate.dialect.MySQL5InnoDBDialect"/>
              </bean>
      </property>
</bean>
……
```

- For Oracle database:

Add the Oracle JDBC driver (e.g. ***ojdbc7.jar***) (available on the Oracle official web site cf.[REF3]) in folder *cef_edelivery_path***/domibus/lib**.

Edit the xml file *cef_edelivery_path***/domibus/conf/domibus/domibus-datasources.xml** and adjust the highlighted parts in the text below according to your environment:

```
<bean id="domibusJDBC-XADataSource" …….>
       <property name="uniqueResourceName">
         <value>domibusJDBC-XA</value>
       </property>
       <property name="xaDataSourceClassName">
         <value>oracle.jdbc.xa.client.OracleXADataSource</value>
       </property>
       <property name="xaProperties">
         <props>
          <prop key="serverName">db_host</prop>
          <!--prop key="port">db_port</prop-->
          <prop key="user">edelivery_user</prop>
          <prop key="password">edelivery_password</prop>
          <prop
key="url">jdbc:oracle:thin:edelivery_user/edelivery_password@//localhost:1521/XE
          </prop>
          </props>
</property>
……
       <property name="testQuery">
         <value>SELECT 1 FROM DUAL</value>
       </property>
……
</bean>
<bean id="entityManagerFactory" ……>
       <property name="jpaVendorAdapter">
<bean class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter">
       <property name="databasePlatform"
value="org.hibernate.dialect.Oracle10gDialect"/>
</bean>
       </property>
```

```
</bean>
```

3. Configure your Keystore based on section §5.1.2 – "*Certificates".*

4. Set JVM parameters

Domibus expects a single JVM parameter **$domibus.config.location**, pointing towards the *cef_edelivery_path***/domibus/conf/domibus** folder.

You can do this by editing the first command lines of *cef_edelivery_path***\domibus\bin\setenv.bat** (Windows) or *cef_edelivery_path***/domibus/bin/setenv.sh** (Linux). Set **CATALINA_HOME** equal to the absolute path of the installation *cef_edelivery_path***/domibus**

- **For Windows : Edit** *cef_edelivery_path***\domibus\bin\setenv.bat** by adding the following:

```
…
set CATALINA_HOME=cef_edelivery_path\domibus
set  JAVA_OPTS=%JAVA_OPTS% -Dfile.encoding=UTF-8 -Xms128m -Xmx1024m -
XX:PermSize=64m
set JAVA_OPTS=%JAVA_OPTS% -Ddomibus.config.location=%CATALINA_HOME%\conf\domibus
…
```

- **For Linux : Edit** *cef_edelivery_path***/domibus/bin/setenv.sh** by adding the following:

```
…
export CATALINA_HOME=cef_edelivery_path/domibus
export JAVA_OPTS="$JAVA_OPTS –Xms128m –Xmx1024m "
export JAVA_OPTS="$JAVA_OPTS -
Ddomibus.config.location=$CATALINA_HOME/conf/domibus"
…
```

5. Launch the Domibus application:

   - For Windows :

```
cd cef_edelivery_path\domibus\bin\
startup.bat
```

   - For Linux :

```
cd cef_edelivery_path /domibus/bin/chmod  u+x *.sh ./startup.sh
```

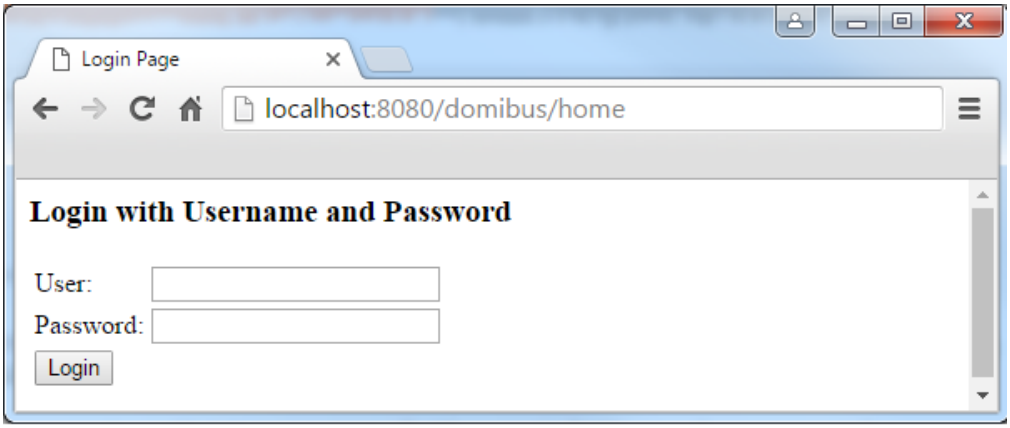
6. Display the Domibus home page on your browser: http://localhost:8080/domibus/home
   (by default: User = ***admin***; Password = ***123456)***

   Remark:

   It is recommended to change the passwords for the default users. See §6.4.1 – "Administration Dashboard" for further information.

   If you can access the page it means the deployment was successful.

Expected result:



4.3.2. Single Server Deployment

For this step, you will have to use the following resources (see §3.1 – "*Binaries repository*" for the download location):

- **domibus-MSH-X.Y.Z-tomcat-configuration.zip**
- **domibus-MSH-X.Y.Z-tomcat.war**

We assume that an Apache Tomcat 8.0.x is already installed and the installation location is now considered as your *cef_edelivery_path/domibus***.**

1. Download and unzip the artefact **domibus-MSH-X.Y.Z-tomcat-configuration.zip** into the directory *cef_edelivery_path/domibus/conf/domibus*
2. Configure the MySQL or Oracle datasource as indicated in §4.3.1 – "*Pre-Configured Single Server Deployment*"
3. Configure your Keystore based on §5.1.2 – "*Certificates*".
4. Execute *step 4* from §4.3.1 – "*Pre-Configured Single Server Deployment*"
5. Rename **domibus-MSH-X.Y.Z-tomcat.war** into **domibus.war** and deploy it to *cef_edelivery_path***/domibus /webapps**.

6. Launch the Domibus application:
    - For Windows :

cd *cef_edelivery_path***\domibus\bin**
startup.bat

    - For Linux :

cd *cef_edelivery_path* **/domibus/bin/**
chmod +x *.sh
./startup.sh

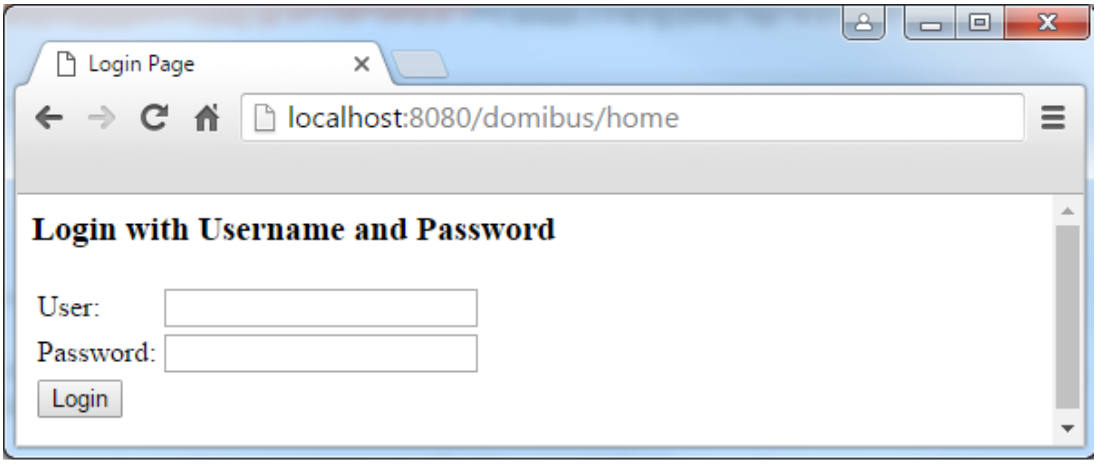
7. Display the Domibus home page on your browser: http://localhost:8080/domibus/home (by default: User = ***admin***; Password = ***123456)***

   Remark:

   > *It is recommended to change the passwords for the default users. See §6.4.1 –* "Administration Dashboard" *for further information.*

   If you can access the page it means the deployment was successful.

Expected result:



4.3.3. Clustered Deployment

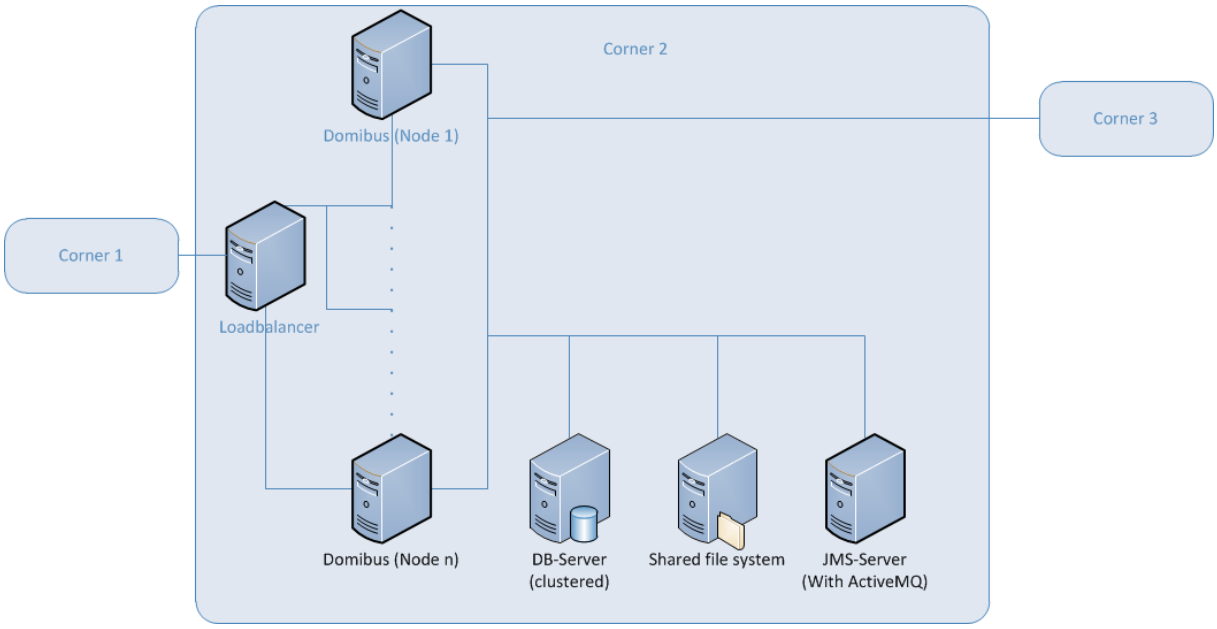


Figure 2 - Diagram representing the Deployment of Domibus in a Cluster on Tomcat

Remark:

In this section we assume that a JMS Broker and a Loadbalancer are configured separately (e.g. httpd).

For this step, you will have to use the following resources (see §3.1 – "*Binaries repository"* for the download location):

- **domibus-MSH-X.Y.Z-tomcat-full.zip**
- **domibus-MSH-X.Y.Z-tomcat.war**

1. Follow steps **1**, **2, 3** and **4** from the §4.3.2 – "*Single Server Deployment*"

2. Set JVM parameters

Domibus expects a single JVM parameter **$domibus.config.location**, pointing towards the *cef_edelivery_path***/domibus/conf/domibus** folder.

You can do this by editing *cef_edelivery_path***\domibus\bin\setenv.bat** (Windows) or *cef_edelivery_path***/domibus/bin/setenv.sh** (Linux). Set **CATALINA_HOME** equal to the absolute path of the installation *cef_edelivery_path***/Domibus**

- **For Windows: Edit** *cef_edelivery_path***\domibus\bin\setenv.bat** by adding the following:

Remark:

your_node_id refers to the installed node in the cluster which starts normally at 01(then 02, etc.)

```
…
set CATALINA_HOME=cef_edelivery_path\domibus
set  JAVA_OPTS=%JAVA_OPTS% -Dfile.encoding=UTF-8 -Xms128m -Xmx1024m -
XX:PermSize=64m
set JAVA_OPTS=%JAVA_OPTS% -Ddomibus.config.location=%CATALINA_HOME%\conf\domibus
set JAVA_OPTS=%JAVA_OPTS% -Ddomibus.node.id=your_node_id
…
```

- **For Linux : Edit** *cef_edelivery_path*/**domibus/bin/setenv.sh** by adding the following:

```
…
export CATALINA_HOME=cef_edelivery_path/domibus
export JAVA_OPTS=$JAVA_OPTS -Xms128m -Xmx1024m
export JAVA_OPTS="$JAVA_OPTS -
Ddomibus.config.location=$CATALINA_HOME/conf/domibus"
export JAVA_OPTS="$JAVA_OPTS -Ddomibus.node.id=your_node_id"
…
```

3. Integrate JMS Broker with Domibus nodes

   - Modify *cef_edelivery_path*/**domibus/conf/domibus/internal/activemq.xml**

Set the uri to the running JMS-broker.

```
<transportConnector uri="tcp://your_ip:your_port" disableAsyncDispatch="true"/>
```

   - Modify *cef_edelivery_path*/**domibus/conf/domibus/domibus-datasources.xml**

Set the broker to the running JMS-broker.

```
<amq:xaConnectionFactory id="xaJmsConnectionFactory" brokerURL="tcp://your_ip:your_port
" userName="admin_username" password="admin_password"/>
```

Remove the highlighted parts below:

```
<bean id="domibusJMS-XAConnectionFactory"
class="com.atomikos.jms.AtomikosConnectionFactoryBean" init-method="init"
destroy-method="close" depends-on="broker">
    <property name="uniqueResourceName" value="domibusJMS-XA"/>
    <property name="xaConnectionFactory" ref="xaJmsConnectionFactory"/>
    <property name="maxPoolSize" value="20"/>
</bean>
<!--  lets create an ActiveMQ Broker -->
<bean id="broker"
      class="org.apache.activemq.xbean.BrokerFactoryBean">
    <property name="config"
      value="file:///${domibus.config.location}/internal/activemq.xml"/>
</bean>
```

4. Change parameters in *cef_edelivery_path*/**domibus/conf/domibus/domibus-transactions.xml**

For clustered deployment:

```
Uncomment the following lines:
<prop>${domibus.work.location:${domibus.config.location}}/
work/transactions/${domibus.node.id}</prop>
<prop>${domibus.work.location:${domibus.config.location}}/
work/transactions/${domibus.node.id}/log</prop>
Comment the following line:
<prop>${domibus.work.location:${domibus.config.location}}/
work/transactions</prop>
<prop>${domibus.work.location:${domibus.config.location}}/
work/transactions/log</prop>
```

5. Follow step **6** and **7** from the §4.3.2 – "*Single Server Deployment*"

4.4. Domibus on WildFly

4.4.1. Pre-Configured Single Server Deployment

In this section we assume that WildFly is installed at the location *cef_edelivery_path***/domibus**

For this step, you will have to use the following resources (see section *3.1 Binaries repository* for the download location):

- **domibus-MSH-X.Y.Z-wildfly-full.zip**

1. Download and unzip the **domibus-MSH-X.Y.Z-wildfly-full.zip** archive in your *cef_edelivery_path* location.

2. Configure the MySQL database (Option 1).

   - Drivers:

     Create the directory *cef_edelivery_path***/domibus/modules/system/layers/base/com/mysql/main** if it does not exist**.** Under this directory:

     - Download and copy the MySQL JDBC driver. (Available on MySQL official web site cf.[REF2]) in the folder.

       Remark:

       The version of the driver has to be mysql-connector-java-5.1.40.jar or higher.

     - Create or edit the file *cef_edelivery_path***/domibus/modules/system/layers/base/com/mysql/main/module.xml** and copy the following module configuration. Make sure to put the name of the driver you are using as an argument of **resource-root** element. e.g. ***mysql-connector-java-5.1.40.jar***:

```
<module xmlns="urn:jboss:module:1.1" name="com.mysql">
      <resources>
            <resource-root path="mysql-connector-java-5.1.40.jar"/>
      </resources>
      <dependencies>
            <module name="javax.api"/>
            <module name="javax.transaction.api"/>
     </dependencies>
</module>
```

- Add your DBMS driver metadata to the Drivers section of the *cef_edelivery_path***/domibus/standalone/configuration/standalone-full.xml**

```
<subsystem xmlns="urn:jboss:domain:datasources:3.0">
   ……
   <datasources>
       ……
       <drivers>
           <driver name="com.mysql" module="com.mysql">
             <driver-class>com.mysql.jdbc.Driver</driver-class>
             <xa-datasource-class>
                 com.mysql.jdbc.jdbc2.optional.MysqlXADataSource
             </xa-datasource-class>
         </driver>
       <drivers>
      ……
   </datasources>
   ……
</subsystem>
```

- Datasources
  - Add the datasources as indicated below to *cef_edelivery_path***/domibus/standalone/configuration/standalone-full.xml**.

Remark:

*Please make sure you modify the connection details for the **MysqlXADS** datasource for MySQL according to your environment.*

```
<subsystem xmlns="urn:jboss:domain:datasources:3.0">
<datasources>
……..
<xa-datasource jndi-name="java:/jdbc/cipaeDeliveryDs" pool-
name="eDeliveryMysqlXADS" enabled="true" use-ccm="true" statistics-enabled="true">
      <xa-datasource-property name="ServerName">db_host</xa-datasource-property>
           <xa-datasource-property name="DatabaseName">domibus_schema</xa-
datasource-property>
           <xa-datasource-
class>com.mysql.jdbc.jdbc2.optional.MysqlXADataSource</xa-datasource-class>
            <driver>com.mysql</driver>
                  <security>
                      <user-name>edelivery_user</user-name>
                      <password>edelivery_password</password>
                  </security>
<validation>
<valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnectionChecker"/>
<background-validation>true</background-validation>
<exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLExceptionSorter"/>
            </validation>
      </xa-datasource>
……..
</datasources>
</subsystem>
```

3. Configure the Oracle Database (option 2)
   - Drivers:

Create the directory *cef_edelivery_path***/domibus/modules/system/layers/base/com/oracle/main** if it does not exist. Under this directory:

- Download and copy the Oracle JDBC driver. (e.g. ***ojdbc7.jar***) (Available on the Oracle official web site cf.[REF3][REF4]) in the folder.
- Copy the file *cef_edelivery_path***/domibus/modules/system/layers/base/com/mysql/main/module.xml** then copy it in the folder recently created.

  Edit **module.xml** by copying the following module configuration. Make sure to put the name of the driver you are using as an argument of **resource-root** element. e.g. ***ojdbc7.jar***:

```
<module xmlns="urn:jboss:module:1.1" name="com.oracle">
	<resources>
		<resource-root path="ojdbc7.jar"/>
	</resources>
	<dependencies>
		<module name="javax.api"/>
		<module name="javax.transaction.api"/>
	</dependencies>
</module>
```

- Add your DBMS driver metadata to the Drivers section in *cef_edelivery_path***/domibus/standalone/configuration/standalone-full.xml** (Only change the items described below while replacing MYSQL configuration in the process).

  - Datasources
    - Add the datasources as indicated below to *cef_edelivery_path***/domibus/standalone/configuration/standalone-full.xml**.

Remark:

Please make sure you modify the connection details for the ***eDeliveryOracleXADS*** *datasource for Oracle according to your environment.*

xe is the Oracle Database SID

```
<xa-datasource jndi-name="java:/jdbc/cipaeDeliveryDs" pool-
name="eDeliveryOracleXADS" enabled="true" use-ccm="true">
.
.
<xa-datasource jndi-name="java:/jdbc/cipaeDeliveryDs" pool-
name="eDeliveryOracleXADS" enabled="true" use-ccm="true">
.
.
<xa-datasource-property name="URL">
jdbc:oracle:thin:username/user_password@//localhost:1521/xe

<xa-datasource-property name="DatabaseName">
domibus
</xa-datasource-property>.
.
.
<driver>com.oracle</driver>
.
.
<valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectionChecker"/
>
.
```

```
.
<exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorter"/>
.
.
<driver name="com.oracle" module="com.oracle">
<xa-datasource-class>oracle.jdbc.xa.client.OracleXADataSource</xa-datasource-
class>
```

- Edit the configuration file *cef_edelivery_path***/domibus/conf/domibus/domibus-datasources.xml** and configure the datasources as indicated below.

Remark:

Configure the database dialect as it is by default pre-configured for MySQL.

```
<property name="databasePlatform" value="org.hibernate.dialect.Oracle10gDialect"/>
.
.
<prop
key="hibernate.connection.driver_class">oracle.jdbc.driver.OracleDriver</prop>

.
<prop key="hibernate.dialect">org.hibernate.dialect.Oracle10gDialect</prop>
```

4. Configure your Keystore based on §5.1.2 – "*Certificates*".
5. Run the standalone server:
   - For Windows under *cef_edelivery_path***\domibus\bin**
   - **standalone.bat --server-config=standalone-full.xml**
   - For Linux under *cef_edelivery_path***/domibus/bin/**

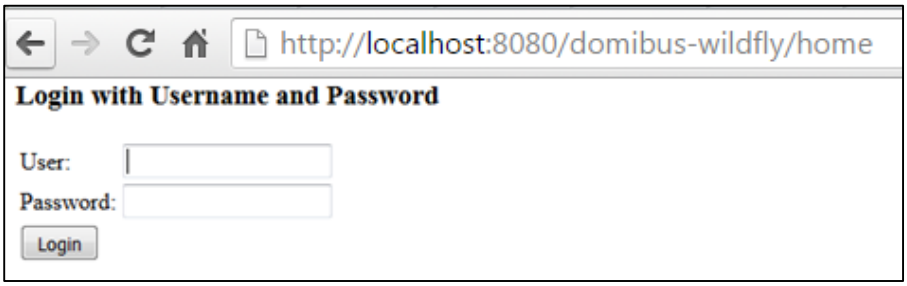
     standalone.sh --server-config=standalone-full.xml
6. Display the Domibus home page on your browser: http://*localhost:8080*/domibus-wildfly/home
   (by default: User = ***admin***; Password = ***123456)***

   Remark:

   It is recommended to change the passwords for the default users. See §6.4.1 – "Administration Dashboard" for further information.

   If you can access the page it means the deployment was successful.

Expected result:



4.4.2. Single Server Deployment

In this section we assume that WildFly is installed at the location *cef_edelivery_path*/**domibus**

For this step, you will have to use the following resources (see §3.1 – "*Binaries repository*" for the download location):

- **domibus-MSH-X.Y.Z-wildfly.war**
- **domibus-MSH-X.Y.Z-wildfly-configuration.zip**

1. Follow steps **2** (MySQL) or **3** (Oracle) from the §4.4.1 – "*Pre-Configured Single Server Deployment*"

2. Configure the environment variables under *cef_edelivery_path*/**domibus/bin/standalone.conf**:

```
……….
JAVA_OPTS="-Xms128m –Xmx1024m
 java.net.preferIPv4Stack=true"
JAVA_OPTS="$JAVA_OPTS -Ddomibus.config.location=$JBOSS_HOME/conf/domibus
……….
```

3. Download and unzip **domibus-MSH-X.Y.Z-wildfly-configuration.zip** in the directory *cef_edelivery_path*/**domibus/conf/domibus**
4. Configure your Keystore based on §5.1.2 – "*Certificates*".
5. Configure the JMS resources

Configure the JMS resources in the configuration file *cef_edelivery_path*/**domibus/standalone/configuration/standalone-full.xml** by adding the **jms-connection-factories** and **jms-queues**.

```
<address-settings>
   <!--default for catch all-->
   <address-setting match="#">
       <dead-letter-address>jms.queue.DLQ</dead-letter-address>
       <expiry-address>jms.queue.ExpiryQueue</expiry-address>
       <max-size-bytes>10485760</max-size-bytes>
       <page-size-bytes>2097152</page-size-bytes>
       <message-counter-history-day-limit>10</message-counter-history-day-limit>
   </address-setting>
   <address-setting match="jms.queue.DomibusSendMessageQueue">
       <dead-letter-address>jms.queue.DomibusDLQ</dead-letter-address>
       <expiry-address>jms.queue.ExpiryQueue</expiry-address>
       <redelivery-delay>1000</redelivery-delay>
       <max-delivery-attempts>1</max-delivery-attempts>
   </address-setting>
   <address-setting match="jms.queue.DomibusBusinessMessageOutQueue">
       <dead-letter-address>jms.queue.DomibusDLQ</dead-letter-address>
       <expiry-address>jms.queue.ExpiryQueue</expiry-address>
       <redelivery-delay>300000</redelivery-delay>
       <max-delivery-attempts>10</max-delivery-attempts>
   </address-setting>
   <address-setting match="jms.queue.DomibusNotifyBackendJmsQueue">
       <dead-letter-address>jms.queue.DomibusDLQ</dead-letter-address>
       <expiry-address>jms.queue.ExpiryQueue</expiry-address>
       <redelivery-delay>300000</redelivery-delay>
       <max-delivery-attempts>10</max-delivery-attempts>
   </address-setting>
```

```
   <address-setting match="jms.queue.DomibusErrorNotifyConsumerQueue">
       <dead-letter-address>jms.queue.DomibusDLQ</dead-letter-address>
       <expiry-address>jms.queue.ExpiryQueue</expiry-address>
       <redelivery-delay>300000</redelivery-delay>
       <max-delivery-attempts>10</max-delivery-attempts>
   </address-setting>
   <address-setting match="jms.queue.DomibusErrorNotifyProducerQueue">
       <dead-letter-address>jms.queue.DomibusDLQ</dead-letter-address>
       <expiry-address>jms.queue.ExpiryQueue</expiry-address>
       <redelivery-delay>300000</redelivery-delay>
       <max-delivery-attempts>10</max-delivery-attempts>
   </address-setting>
       <address-setting match="jms.queue.DomibusBusinessMessageInQueue">
       <dead-letter-address>jms.queue.DomibusDLQ</dead-letter-address>
       <expiry-address>jms.queue.ExpiryQueue</expiry-address>
       <redelivery-delay>300000</redelivery-delay>
       <max-delivery-attempts>10</max-delivery-attempts>
   </address-setting>
   <address-setting match="jms.queue.DomibusPluginToBackendQueue">
       <dead-letter-address>jms.queue.DomibusDLQ</dead-letter-address>
       <expiry-address>jms.queue.ExpiryQueue</expiry-address>
       <redelivery-delay>300000</redelivery-delay>
       <max-delivery-attempts>10</max-delivery-attempts>
   </address-setting>
   <address-setting match="jms.queue.DomibusNotifyBackendWebServiceQueue">
       <dead-letter-address>jms.queue.DomibusDLQ</dead-letter-address>
       <expiry-address>jms.queue.ExpiryQueue</expiry-address>
       <redelivery-delay>300000</redelivery-delay>
       <max-delivery-attempts>10</max-delivery-attempts>
   </address-setting>
   <address-setting match="jms.queue.DomibusUnknownReceiverQueue">
       <dead-letter-address>jms.queue.DomibusDLQ</dead-letter-address>
       <expiry-address>jms.queue.ExpiryQueue</expiry-address>
       <redelivery-delay>300000</redelivery-delay>
       <max-delivery-attempts>10</max-delivery-attempts>
   </address-setting>
   <address-setting match="jms.queue.DomibusNotifyBackendQueue">
       <dead-letter-address>jms.queue.DomibusDLQ</dead-letter-address>
       <expiry-address>jms.queue.ExpiryQueue</expiry-address>
       <redelivery-delay>300000</redelivery-delay>
       <max-delivery-attempts>10</max-delivery-attempts>
   </address-setting>
   <address-setting match="jms.queue.DomibusClusterCommandTopic">
       <dead-letter-address>jms.queue.DomibusDLQ</dead-letter-address>
       <expiry-address>jms.queue.ExpiryQueue</expiry-address>
       <redelivery-delay>10000</redelivery-delay>
       <max-delivery-attempts>3</max-delivery-attempts>
   </address-setting>
</address-settings>
……
<subsystem xmlns="urn:jboss:domain:messaging:3.0">
   <hornetq-server>
       <jmx-management-enabled>true</jmx-management-enabled>
       <jms-connection-factories>
……
           <connection-factory name="edeliveryConnectionFactory">
               <connectors>
                  <connector-ref connector-name="in-vm"/>
               </connectors>
                <entries>
```

```
                    <entry name="java:/jms/ConnectionFactory"/>
                </entries>
                 <compress-large-messages>false
                 </compress-large-messages>
                  <failover-on-initial-connection>false
                  </failover-on-initial-connection>
                   <use-global-pools>true</use-global-pools>
          </connection-factory>
        ..........................
      </jms-connection-factories>
        <jms-destinations>
        ..........................
        <jms-queue name="DomibusBusinessMessageOutQueue">
         <entry name="java:/jms/domibus.backend.jms.outQueue"/>
         <entry name="java:/jms/queue/DomibusBusinessMessageOutQueue"/>
            <durable>true</durable>
        </jms-queue>
        <jms-queue name="DomibusNotifyBackendJmsQueue">
         <entry name="java:/jms/domibus.notification.jms"/>
         <entry name="java:/jms/queue/DomibusNotifyBackendJmsQueue"/>
        <durable>true</durable>
        </jms-queue>
        <jms-queue name="DomibusErrorNotifyConsumerQueue">
         <entry name="java:/jms/domibus.backend.jms.errorNotifyConsumer"/>
         <entry name="java:/jms/queue/DomibusErrorNotifyConsumerQueue"/>
            <durable>true</durable>
        </jms-queue>
        <jms-queue name="DomibusErrorNotifyProducerQueue">
         <entry name="java:/jms/domibus.backend.jms.errorNotifyProducer"/>
         <entry name="java:/jms/queue/DomibusErrorNotifyProducerQueue"/>
            <durable>true</durable>
        </jms-queue>
        <jms-queue name="DomibusBusinessMessageInQueue">
         <entry name="java:/jms/domibus.backend.jms.inQueue"/>
         <entry name="java:/jms/queue/DomibusBusinessMessageInQueue"/>
            <durable>true</durable>
        </jms-queue>
        <jms-queue name="DomibusPluginToBackendQueue">
         <entry name="java:/jms/domibus.backend.jms.replyQueue"/>
         <entry name="java:/jms/queue/DomibusPluginToBackendQueue"/>
            <durable>true</durable>
        </jms-queue>
        <jms-queue name="DomibusSendMessageQueue">
         <entry name="java:/jms/domibus.internal.dispatch.queue"/>
         <entry name="java:/jms/queue/DomibusSendMessageQueue"/>
            <durable>true</durable>
        </jms-queue>
        <jms-queue name="DomibusNotifyBackendWebServiceQueue">
         <entry name="java:/jms/domibus.notification.webservice"/>
         <entry name="java:/jms/queue/DomibusNotifyBackendWebServiceQueue"/>
            <durable>true</durable>
        </jms-queue>
        <jms-queue name="DomibusUnknownReceiverQueue">
         <entry name="java:/jms/domibus.internal.notification.unknown"/>
         <entry name="java:/jms/queue/DomibusUnknownReceiverQueue"/>
            <durable>true</durable>
        </jms-queue>
        <jms-queue name="DomibusNotifyBackendQueue">
         <entry name="java:/jms/domibus.internal.notification.queue"/>
         <entry name="java:/jms/queue/DomibusNotifyBackendQueue"/>
```

```
            <durable>true</durable>
         </jms-queue>
         <jms-queue name="DLQ">
          <entry name="java:/jms/domibus/ DLQ"/>
          <entry name="java:/jms/queue/DLQ"/>
            <durable>true</durable>
         </jms-queue>
         <jms-topic name="DomibusClusterCommandTopic">
          <entry name="java:/jms/domibus.internal.command"/>
          <entry name="java:/jms/topic/DomibusClusterCommandTopic"/>
         </jms-topic>
         ……….
         </jms-destinations>
    </hornetq-server>
</subsystem>
```

Remark:

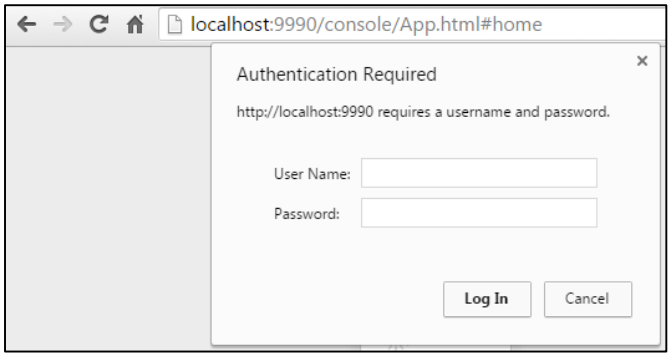
Please note also the JMX management has to be enabled so the JMS resources can be monitored in the JMS Monitoring screen.

6. Configure the executor services

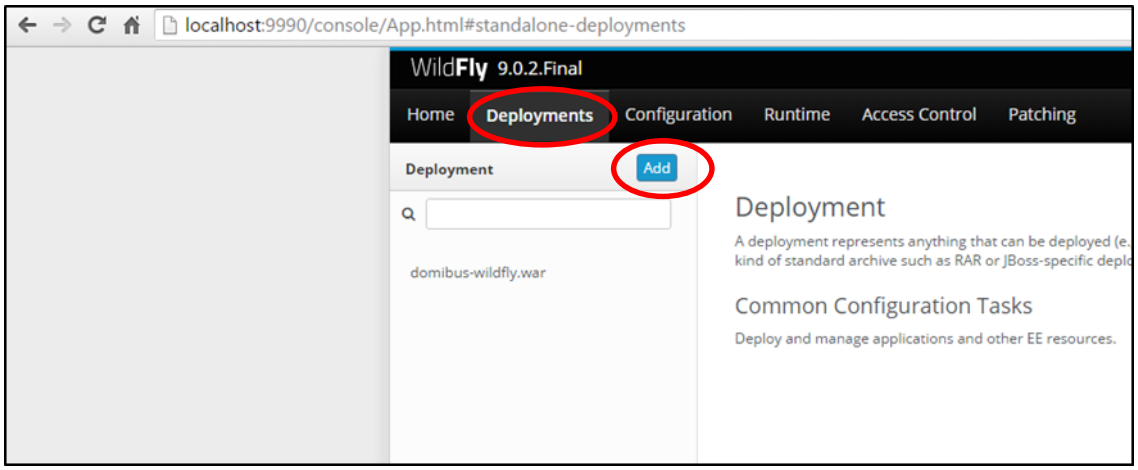
Configure the executor's services in the configuration file *cef_edelivery_path***/domibus/standalone/configuration/standalone-full.xml**

```
<subsystem xmlns="urn:jboss:domain:ee:3.0">
    ……................
    <concurrent>
        ……................
        <managed-executor-services>
           <managed-executor-service name="domibusExecutorService" jndi-
name="java:jboss/ee/concurrency/executor/DomibusExecutorService" context-
service="default" hung-task-threshold="60000" core-threads="5" max-threads="25"
keepalive-time="5000"/>
       </managed-executor-services>
       <managed-executor-services>
           <managed-executor-service name="quartzExecutorService" jndi-
name="java:jboss/ee/concurrency/executor/QuartzExecutorService" context-
service="default" hung-task-threshold="0" long-running-tasks="true" core-
threads="5" max-threads="25" keepalive-time="5000"/>
       </managed-executor-services>
         ……................
     </concurrent>
     ……................
<subsystem xmlns="urn:jboss:domain:ee:3.0">
```

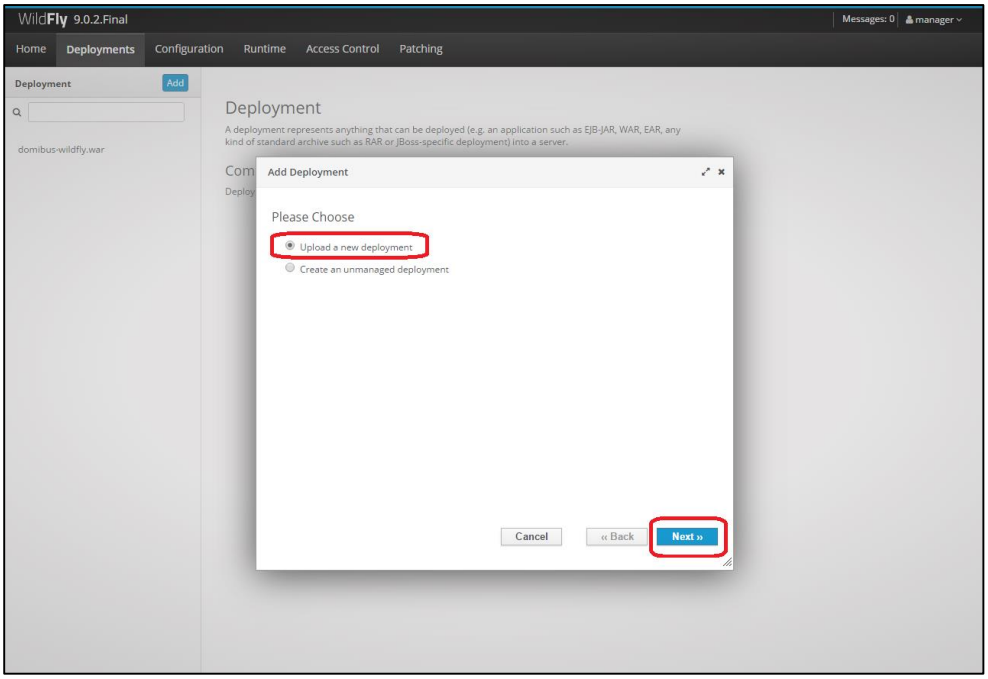
7. Connect to the Admin Console of WildFly at **http://*localhost*:9990/*console***



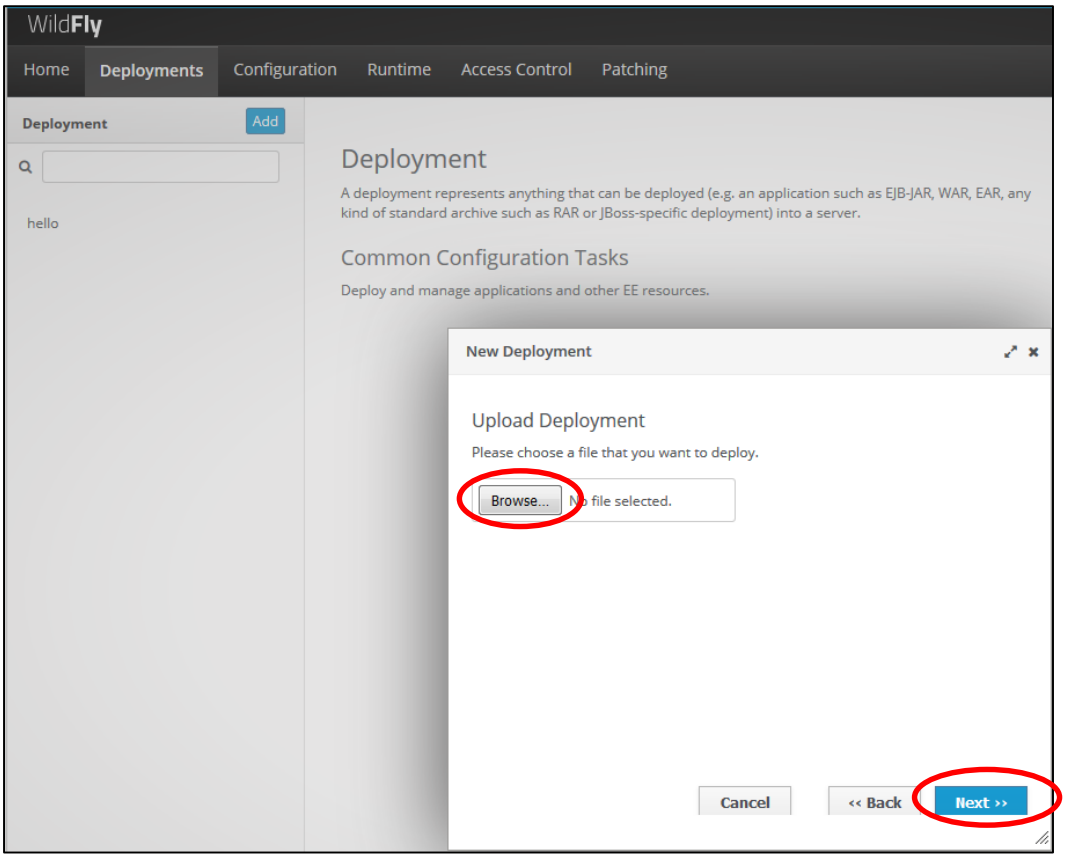
8. Click on **Deployments** in the console menu then click on **Add**



9. Select **Upload a new deployment** then click **Next**

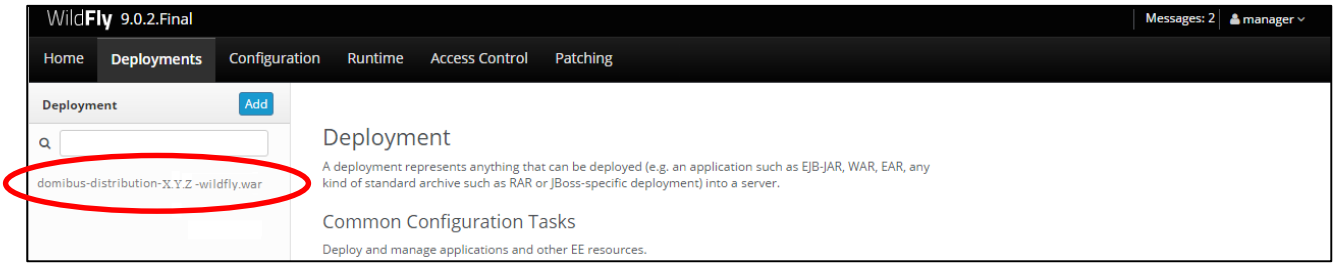


10. Browse to the location of the **domibus-distribution-X.Y.Z-wildfly.war** file, select it and click **Next**



11. The deployment is successful when the name of the .war file appears in the Deployment column.

Expected Result:



4.4.3. Clustered Deployment

For this step, you will have to use the following resources (see §3.1 – "*Binaries repository*" for the download location):

- **domibus-MSH-X.Y.Z-wildfly-configuration.zip**
- **domibus-MSH-X.Y.Z-wildfly.war**

In this section we assume that the setup of WildFly 9 in domain mode has already been done and that the cluster has been enabled as described in the official documentation. For more details on how to perform an installation of Wildfly 9 in domain mode please refer to the official documentation cf.[REF4].

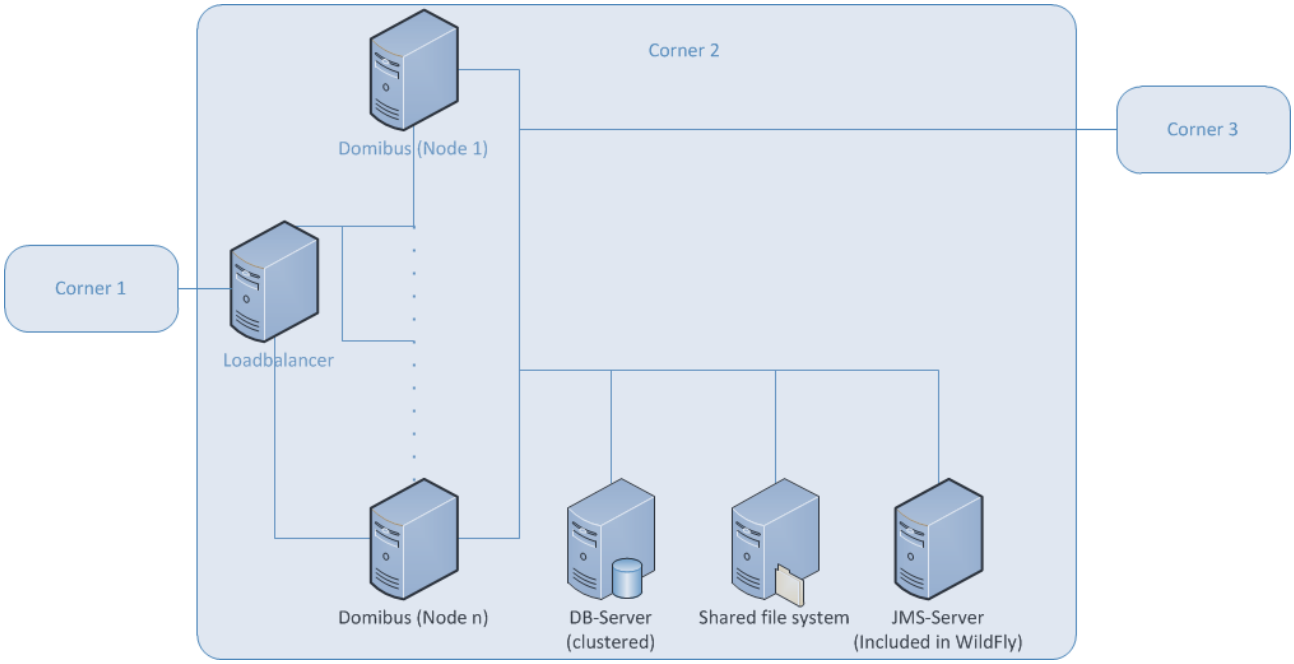


Figure 3 - Diagram representing the Deployment of Domibus in a Cluster on WildFly

In order to install Domibus in a WildFly cluster please follow the steps below:

1. Download and unzip **domibus-MSH-X.Y.Z-wildfly-configuration.zip** in a shared location that is accessible by all the nodes from the cluster. We will refer to this directory as *cef_shared_edelivery_path***/Domibus**

2. Follow steps **2** (MySQL) or **3** (Oracle) from the §4.4.1 – "*Pre-Configured Single Server Deployment*"

   Remarks:

   - *This step needs to be performed on all the nodes from the cluster*
   - *In the following 2 steps we will edit the profile **full-ha** from the configuration file **domain/configuration/domain.xml** located in the master node*

3. Configure the JMS queues and topics as indicated in §4.4.2 point 5 – "*Configure the JMS resources*"

4. Configure the database dialect as indicated in §4.4.1 point 3 – "*Edit the configuration file cef_edelivery_path**/domibus/conf/domibus/domibus-datasources.xml***"

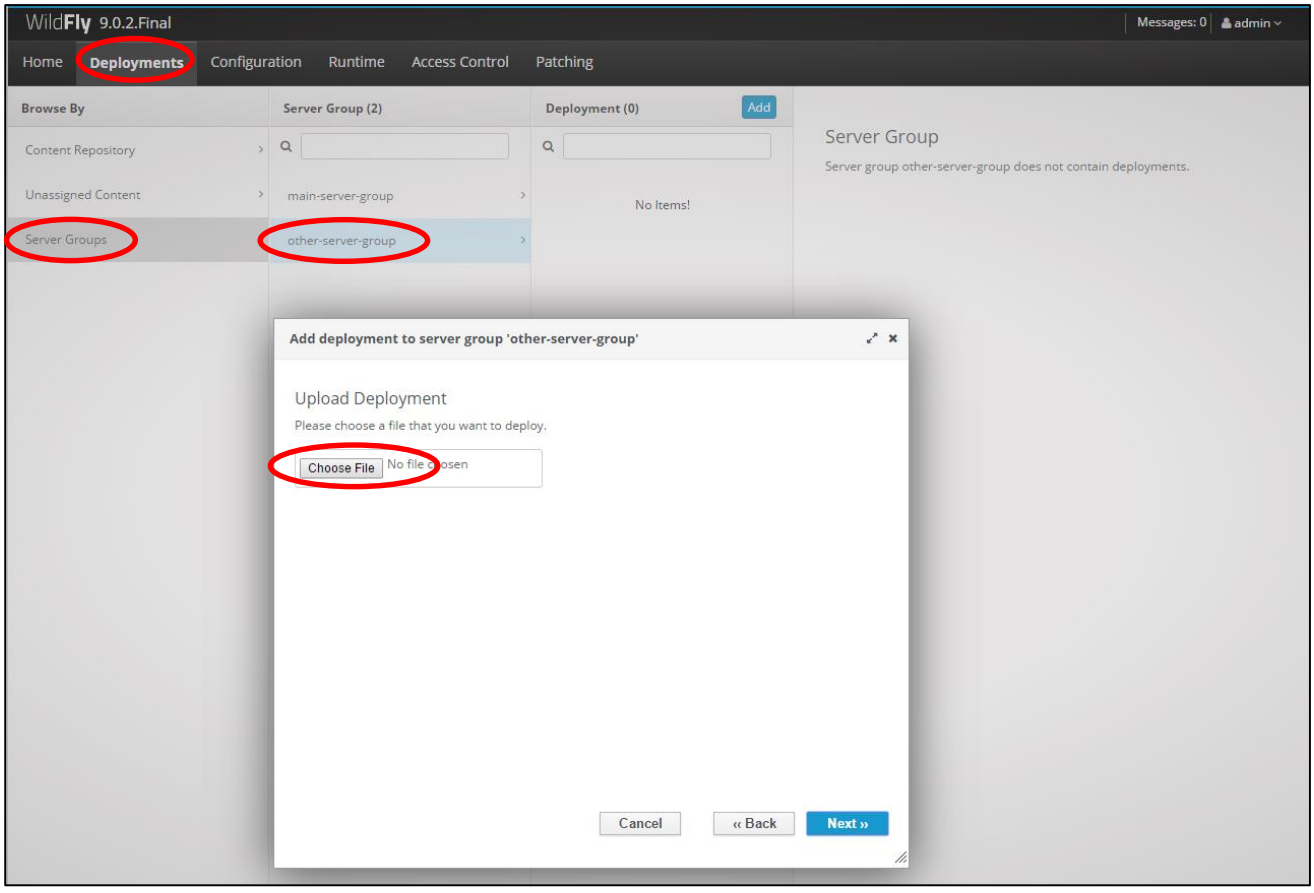
5. Configure the environment variables in the file **bin/domain.conf**

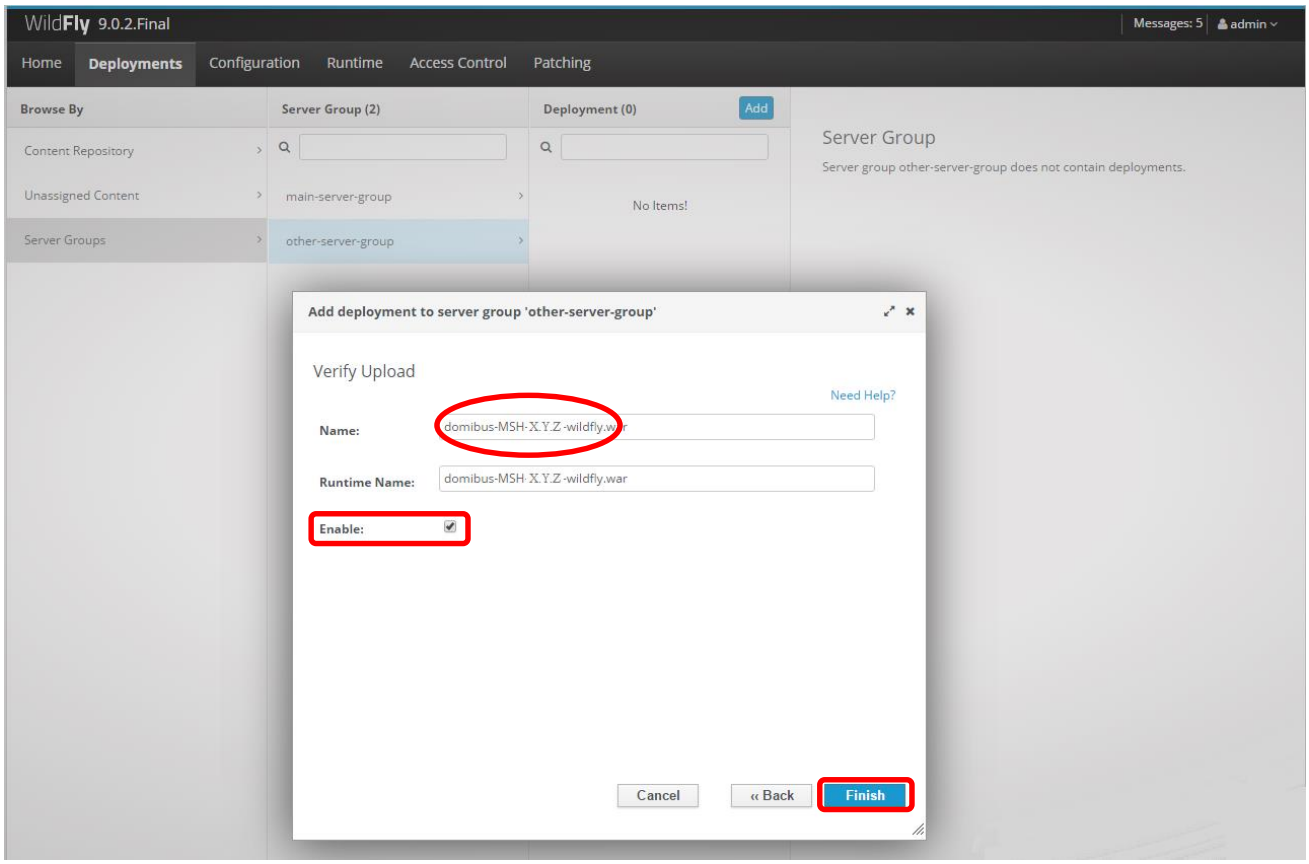
Remark:

bin/domain.conf is located in each WildFly node. The environment variable setting needs to be performed in every node from the cluster.

```
……
JAVA_OPTS="-Xms128m –Xmx1024m
-java.net.preferIPv4Stack=true"
JAVA_OPTS="$JAVA_OPTS -
Ddomibus.config.location=cef_shared_edelivery_path/conf/domibus
……
```

6. Deploy the **domibus-MSH-X.Y.Z-wildfly.war** to the cluster. We will use the WildFly Administration console for performing the deployment. We will deploy the application on the **other-server-group** cluster which is configured step by step in the official documentation cf.[REF4].




WildFly 9.0.2.Final
Messages: 5
admin
Home
Deployments
Configuration
Runtime
Access Control
Patching
Browse By
Content Repository
Unassigned Content
Server Groups
Server Group (2)
main-server-group
other-server-group
Deployment (0)
Add
No Items!
Server Group
Server group other-server-group does not contain deployments.
Add deployment to server group 'other-server-group'
Verify Upload
Need Help?
Name:
domibus-MSH-X.Y.Z-wildfly.war
Runtime Name:
domibus-MSH-X.Y.Z-wildfly.war
Enable:
Cancel
« Back
Finish

5. DOMIBUS CONFIGURATION

Domibus application has one main webservice:

- services/msh: The Message Service Handler is the URL of your AS4 Access Point endpoint. This interface has to be exposed on the internet and should be reachable by your correspondent Access Point(s).

Domibus has also one optional webservice:

- services/backend: The URL of the backend webservice. This interface should ONLY be exposed to your backend client(s) within your internal network. This uses the default WS plugin (§6.1.2 – "*WS Plugin*")

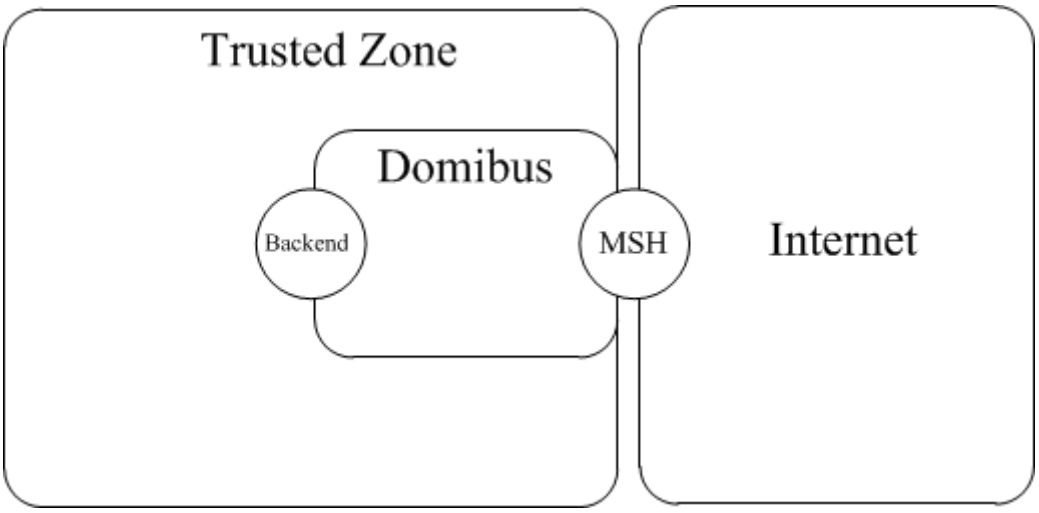


Figure 4 - Message Service Handler diagram

5.1. Security Configuration

5.1.1. Policies

Domibus uses a security policy which mandatory to be fully conformant with the e-SENS AS4 profile. It is used for the configuration of WS-Security. The policy is referenced in the PMode configuration file in §6.3 – "*PMode Configuration*". The policy definition can be found under *cef_edelivery_path***/domibus/conf/domibus/policies/eDeliveryPolicy.xml**. For more information related to the AS4 e-SENS profile, please refer to [REF10].

5.1.2. Certificates

Domibus Access Point uses certificates for the encryption, for the signature of the AS4 messages and for establishing trust with other Access Points.
The certificates need to be configured in the keystore and truststore JKS files defined in **domibus-security.xml** configuration file.
The keystore contains the certificate of the Access Point which includes its private and public keys. The truststore contains the public keys of the trusted Access Points.

Modify the file *cef_edelivery_path***/domibus/conf/domibus*/domibus-security.xml*** as defined in the box below:

```
<property name="password_store">
<util:map><entry value="your_privatekey_password" key="your_keystore_alias"/>
</util:map>
</property>
…
<!-- The password used to load the keystore -->
<prop
key="org.apache.ws.security.crypto.merlin.keystore.password">your_keystore_passw
ord</prop>
<!-- The keystore alias to use for decryption and signing. -->
<prop
key="org.apache.ws.security.crypto.merlin.keystore.alias">your_keystore_alias</p
rop>
…
<!-- The password used to load the truststore -->
<prop key="org.apache.ws.security.crypto.merlin.truststore.password">
your_trustore_password</prop>
…
<!-- The location of the keystore -->
        <prop
key="org.apache.ws.security.crypto.merlin.file">${domibus.config.location}/keys
tores/your_keystore.jks</prop>
…
<!-- The location of the truststore -->
        <prop
key="org.apache.ws.security.crypto.merlin.truststore.file">${domibus.config.loca
tion}/keystores/your_truststore.jks</prop>
```

1. Create, if not present, a folder *cef_edelivery_path***/domibus/conf/domibus/keystores**

2. Get your key pair from an external provider. (Self-signed certificates should only be used for testing purposes, not production). If you are interested in using the CEF Public Key Infrastructure Solution, cf.[REF5].
3. Create, if not present, the public and private keys containers (e.g. *truststore***.jks** *and keystore***.jks**)
4. Import your private key into your keystore

Remarks:

- *Your private key and your keystore should always stay secret. Please never share them.*
- *The keystore alias has to be the same as the party ID defined in the §6.3 – "PMode Configuration". It is strongly recommended to put your key pair (private and public key) and the public key of the other participants you trust in two separate containers.*

5.2. Domibus Properties

Edit *cef_edelivery_path***/conf/domibus/domibus-configuration.xml** and set:

Configuration Property	Default value	Purpose
domibus.msh.messageid.suffix	domibus.eu	This Property is used to generate the random Message id with a fixed suffix which is set by default to "domibus.eu". The resulting format will be UUID@$domibus.msh.messageid.suffix. This property is mandatory.
domibus.msh.retry.cron	0/5 * * * *	It is the retry cron job to send the messages. It is set by default to every 5 seconds. This property is mandatory
domibus.msh.retry.tolerance	10000	Timeout tolerance for retry messages (ms). Should be set to double of the retry worker execution interval. This property is mandatory
domibus.dispatch.ebms.error.unrecoverable.retry	true	This property should be set to true if Domibus needs to retry sending the failed messages. This property is mandatory
domibus.smlzone	acc.edelivery.tech.ec.europa.eu	Set the SMLZone if Domibus needs to be used under Dynamic discovery model. This property is only mandatory if an SML is used.
domibus.backend.jmsInQueue	domibus.backend.jms.inQueue	This queue is the entry point for messages to be sent by the sending MSH. This property is only mandatory if the JMS plugin is used.
domibus.deployment.clustered	false	If true the quartz scheduler jobs are clustered. This property is mandatory, it should be set to true if the deployment of Domibus is done in a cluster.
message.retention.downloaded.max.delete	50	This property is used to tweak the maximum downloaded messages to be deleted by the retention worker
message.retention.not_downloaded.max.delete	50	This property is used to tweak the maximum not-downloaded messages to be deleted by the retention worker

domibus.attachment.storage.location	-	It is possible to configure Domibus to save the message payloads on the file system instead of the database. This setting is recommended when exchanging payloads bigger than 30MB. In order to enable the file system storage please add the following property to : *cef_edelivery_path***/conf/domibus/domibus-configuration.xml** domibus.attachment.storage.location= *your_file_system_location* where *your_file_system_location* is the location on the file system where the payloads will be saved. Remark: In a cluster configuration the file system storage needs to be accessible by all the nodes from the cluster.
domibus.jmx.user	jmsManager	WebLogic specific: The user that will be used to access the queues via JMX
domibus.jmx.password	jmsManager1	WebLogic specific: The associated password of the configured domibus.jmx.user
domibus.sendMessage.messageIdPattern	^[\x20-\x7E]*$	When an initiator backend client submits messages to Dombus for transmission, with the message id field populated, then the message id should be RFC2822 compliant. The pattern specified here ensures this validation. This field is optional. In case the existing client does not match this message id pattern during submission, then this property can be omitted to skip the validation.
domibus.listPendingMessages.maxCount	500	This property specifies the maximum number of messages that would be served when the 'listPendingMessages' operation is invoked. Setting this property is expected to avoid timeouts due to huge resultsets being served. A value of 0 would return all the pending messages. This property is optional. Omitting this property would default the resultset size to 500.
domibus.dispatcher.connectionTimeout	240000	For connection between the access points – C2 & C3. Specifies the amount of time, in milliseconds, that the consumer will attempt to establish a connection before it times out. 0 is infinite.
domibus.dispatcher.receiveTimeout	240000	For connection between the access points – C2 & C3. Specifies the amount of time, in milliseconds, that the consumer will wait for a response before it times out. 0 is infinite.

Configuration Property	Default value	Purpose
Proxy Settings		In case your Access Point has to use a proxy server you can configure it with these properties.
domibus.proxy.enabled	false	true/false depending on whether you need to use proxy or not.
domibus.proxy.http.host	-	Host name of the proxy server
domibus.proxy.http.port	-	Port of Proxy server
domibus.proxy.user	-	Username for authentication on the proxy server
domibus.proxy.password	-	Password
domibus.proxy.nonProxyHosts	-	Indicates the hosts that should be accessed without going through the proxy.

Table 1 - Domibus Properties

6. PLUGIN MANAGEMENT

This section describes the different types of plugins and their registration process.

6.1. Default Plugins

Domibus comes with two default plugins. The two Interface Control Documents (ICD) describe these two plugins (JMS and WS) cf.[REF6].

6.1.1. JMS Plugin

For the JMS plugin, you will have to use the following resource (see section *3.1 Binaries repository* for the download location):

- **domibus-MSH-X.Y.Z-default-jms-plugin.zip**

6.1.2. WS Plugin

For the WS plugin, you will have to use the following resource (see section *3.1 Binaries repository* for the download location):

- **domibus-MSH-X.Y.Z-default-ws-plugin.zip**

6.1.2.1. Domibus authentication

The default web service plugin for Domibus 3.2.5 includes an example of how to implement authentication and authorization. By default this feature is disabled to insure backwards compatibility with older versions of Domibus.

The documentation below answers the question "*how to enable and use the authentication in the WS plugin?*"

The default WS plugin supports:

- Basic Authentication
- X509Certificates Authentication
- Blue Coat Authentication

Remark:

> *Blue Coat is the name of the reverse proxy at the commission. It forwards the request in HTTP with the certificate details inside the request ("Client-Cert" header key).*

When more than one authentication method is used, the Basic Authentication takes precedence on both http and https.
When no Basic Authentication is provided, X509 Certificates are expected on https requests.
When no Basic Authentication is provided, Blue Coat certificates are expected on http requests.

6.1.2.2. Domibus Authorization

For convenience reasons, the WS plugin uses exactly the same database as configured for Domibus core to store the users/passwords and certificate ids. To learn more about authorization (and authentication), read the plugin cookbook cf.[REF6].

There are two default users already inserted in the database (make sure you already ran the migration scripts),

- *admin* and *user* both with **123456** as password.

- *admin* has the role ROLE_ADMIN and *user* has the role ROLE_USER

Roles:

ROLE_ADMIN has the right to call:

- sendMessage with any value for originalSender property
- downloadMessage (any message among messages notified to this plugin)
- listPendingMessages will list all pending messages for this plugin
- getMessageStatus and getMessageErrors

ROLE_USER has the right to call:

- sendMessage with originalSender equal to the originalUser
- downloadMessage, only if finalRecipient equals the originalUser
- listPendingMessages, only messages with finalRecipient equal to the origi-nalUser

6.1.2.3. Enable the authentication in Domibus

To enable the authentication at Domibus level the following steps must be configured:

1. In **conf/domibus/domibus-configuration.xml** and set the property "**domibus.auth.unsecureLoginAllowed**" to false.

```
<util:properties>
       ……
         <!-- To disable unsecureLogin, set this to false -->
         <prop key="domibus.auth.unsecureLoginAllowed">false</prop>
</util:properties>
```

2. The application server must be configured to allow https requests and pass the authentication credentials to Domibus.

6.2. Custom Plugin

Users can develop their own plugins. Please refer to the Plugin Cookbook cf.[REF6] for more details.

6.2.1. Plugin registration

Remark:

Please refer to section 6.4.5 Message Filtering for the routing of the specific plugin after registering the plugin on your specific Application Server.

6.2.1.1. Tomcat

Remark:

CATALINA_HOME is the folder where the Tomcat is installed.

1. Stop Tomcat server
2. Copy the custom plugin jar file to the plugins folder *CATALINA_HOME***/conf/domibus/plugins/lib**
3. Copy the custom plugin XML configuration file to *CATALINA_HOME***/conf/domibus/plugins/config**
4. Start Tomcat server

6.2.1.2. WebLogic

Remark:

DOMAIN_HOME is the folder corresponding to the WebLogic domain.

1. Stop the WebLogic server
2. Copy the custom plugin jar file to the plugins folder *DOMAIN_HOME***/conf/domibus/plugins/lib**
3. Copy the custom plugin XML configuration file to *DOMAIN_HOME***/conf/domibus/plugins/config**
4. Start the WebLogic server

6.2.1.3. WildFly

In order to install a custom plugin please follow the steps**:**

1. Stop the WildFly server
2. Copy the custom plugin jar file to the plugins folder *cef_edelivery_path* **/conf/domibus/plugins/lib**
3. Copy the custom plugin XML configuration file to *cef_edelivery_path* **/conf/domibus/plugins/config**
4. Start the WildFly server

6.3. PMode Configuration

Processing Modes (PModes) are used to configure Access Points. The PMode parameters are loaded into the Access Point via an XML file.

The following features described in the PMode file are, Security, Reliability, Transport, Business Collaborations, Error Reporting, Message Exchange Patterns (MEPs) and Message Partition Channels (MPCs).

As different messages maybe subject to various types of processing or, as different business domains may have several requirements, Access Points commonly support several PModes. Some PMode parameters are mandatory, others are optional. For more information please refer to the Access Point Component Offering Document.

6.3.1. Configuration

In Domibus, PMode are XML files, you can create one or edit the existing PMode files: *cef_edelivery_path***/domibus/conf/pmodes/domibus-gw-sample-pmode-***party_id_name1***.xml** and *cef_edelivery_path***/domibus/conf/pmodes/domibus-gw-sample-pmode-***party_id_name2*.xml,** by replacing *party_id_name1* with your party name and *party_id_name2* with your correspondent's party name in the file's names and in the files themselves as shown below. The partyID must match the alias of the certificate in the keystore and the endpoint must be the external access link to your own instance.

Remark:

> *This step could be managed by a PMode Configuration Manager known by your Business Owner.*

```
<party name="party_id_name2"
       endpoint="http:// party_id_name2_hostname:8080/domibus/services/msh"
       allowChunking="false">
  <identifier partyId="party_id_name2_1"
              partyIdType="partyTypeUrn"/>
</party>
<party name="party_id_name1"
       endpoint="http:// party_id_name1_hostname:8080/domibus/services/msh"
       allowChunking="false">
  <identifier partyId="party_id_name1_1" partyIdType="partyTypeUrn"/>
</party>
```

Figure 5 - PMode view

6.3.2. Adding a new participant

If a new participant's Access Point is joining your network, you need to edit your PMode accordingly and to re-upload it like mentioned in §6.3.5 – "*Upload new Configuration*".

- Add a "new_party" element

```
<party name="new_party_name"
                    endpoint="http://new_party_msh"
                    allowChunking="false">
      <identifier partyId="new_party_id" partyIdType="partyTypeUrn"/>
</party>
```

- Add your "new_party_name" as initiator

The party with the role of initiator will be the sender of the messages

```
<initiatorParties>
            …
            <initiatorParty name="new_party_name"/>
      </initiatorParties>
```

- Add your "new_party_name" as responder

The party with the role of responder will be the receiver of the messages

```
<responderParties>
            …
            <responderParty name="new_party_name"/>
      </responderParties>
```

6.3.3. Sample PMode file

Processing modes (PModes) describe how messages are exchanged between AS4 partners (in this case *Access Points blue_gw and red_gw*). These files contain the identifiers of each AS4 Access Point (identified as *parties* in the PMode file below).

Sender Identifier and Receiver Identifier represent the organizations that send and receive the business documents. They are both used in the authorization process (PMode). Therefore, adding, modifying or deleting a participant implies modifying the corresponding PMode files.

Here is an example of the content of a PMode XML file:

Remark:

> *In this setup we have allowed each party (blue_gw or red_gw) to initiate the process. If only blue_gw is supposed to send messages, we need to put only blue_gw in <initiatorParties> and red_gw in <responderParties>.*

```
<?xml version="1.0" encoding="UTF-8"?>
<db:configuration xmlns:db="http://domibus.eu/configuration" party="blue_gw">
      <mpcs>
            <mpc name="defaultMpc"
                  qualifiedName="http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/ns/core/200704/defaultMPC"
                  enabled="true"
                  default="true"
                  retention_downloaded="0"
                  retention_undownloaded="14400"/>
      </mpcs>
```

```
      <businessProcesses>
            <roles>
                  <role name="defaultInitiatorRole"
                              value="http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/ns/core/200704/initiator"/>
                  <role name="defaultResponderRole"
                              value="http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/ns/core/200704/responder"/>
            </roles>
            <parties>
                  <partyIdTypes>
                        <partyIdType name="partyTypeUrn"
value="urn:oasis:names:tc:ebcore:partyid-type:unregistered"/>
                  </partyIdTypes>
                  <party name="red_gw"

endpoint="http://<red_hostname>:8080/domibus/services/msh"
                              allowChunking="false">
                        <identifier partyId="domibus-red"
partyIdType="partyTypeUrn"/>
                  </party>
                  <party name="blue_gw"

      endpoint="http://<blue_hostname>:8080/domibus/services/msh"
                              allowChunking="false">
                        <identifier partyId="domibus-blue"
partyIdType="partyTypeUrn"/>
                  </party>
            </parties>
            <meps>
                  <mep name="oneway" value="http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/ns/core/200704/oneWay"/>
                  <mep name="twoway" value="http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/ns/core/200704/twoWay"/>
                  <binding name="push" value="http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/ns/core/200704/push"/>
                  <binding name="pushAndPush" value="http://docs.oasis-
open.org/ebxml-msg/ebms/v3.0/ns/core/200704/push-and-push"/>
            </meps>
            <properties>
                  <property name="originalSenderProperty"
                              key="originalSender"
                              datatype="string"
                              required="true"/>
                  <property name="finalRecipientProperty"
                              key="finalRecipient"
                              datatype="string"
                              required="true"/>
                  <propertySet name="ecodexPropertySet">
                        <propertyRef property="finalRecipientProperty"/>
                        <propertyRef property="originalSenderProperty"/>
                  </propertySet>
            </properties>
            <payloadProfiles>
                  <payload name="businessContentPayload"
                              cid="cid:message"
                              required="true"
                              mimeType="text/xml"/>
                  <payload name="businessContentAttachment"
                              cid="cid:attachment"
```

```
                                    required="false"
                                    mimeType="application/octet-stream"/>
                        <payloadProfile name="MessageProfile"
                                    maxSize="40894464">
                              <attachment name="businessContentPayload"/>
                              <attachment name="businessContentAttachment"/>
                        </payloadProfile>
                  </payloadProfiles>
                  <securities>
                        <security name="eDeliveryPolicy"
                                    policy="eDeliveryPolicy.xml"
                                    signatureMethod="RSA_SHA256" />
                        <security name="noSigNoEnc"
                                      policy="doNothingPolicy.xml"
                                      signatureMethod="RSA_SHA256"/>
                        <security name="eSensPolicy"
                                      policy="eSensPolicy.xml"
                                      signatureMethod="RSA_SHA256"/>
                  </securities>
                  <errorHandlings>
                        <errorHandling name="demoErrorHandling"
                                    errorAsResponse="true"
                                    businessErrorNotifyProducer="false"
                                    businessErrorNotifyConsumer="false"
                                    deliveryFailureNotifyProducer="false"/>
                  </errorHandlings>
                  <agreements>
                 <agreement name="agreement1" value="A1" type=""/>
                 <agreement name="agreement2" value="A2" type=""/>
                 <agreement name="agreement3" value="A3" type=""/>
                  </agreements>
                  <services>
                   <service name="testService1" value="bdx:noprocess" type="tc1"/>
                  </services>
                  <actions>
                        <action name="tc1Action" value="TC1Leg1"/>
                        <action name="tc2Action" value="TC2Leg1"/>
                  </actions>
                  <as4>
                        <receptionAwareness name="receptionAwareness"
retry="12;4;CONSTANT" duplicateDetection="true"/>
                        <reliability name="AS4Reliability" nonRepudiation="true"
replyPattern="response"/>
                        <reliability name="noReliability" nonRepudiation="false"
replyPattern="response"/>
                  </as4>
                  <legConfigurations>
                        <legConfiguration name="pushTestcase1tc1Action"
                                    service="testService1"
                                    action="tc1Action"
                                    defaultMpc="defaultMpc"
                                    reliability="AS4Reliability"
                                    security="eDeliveryPolicy"
                                    receptionAwareness="receptionAwareness"
                                    propertySet="ecodexPropertySet"
                                    payloadProfile="MessageProfile"
                                    errorHandling="demoErrorHandling"
                                    compressPayloads="true"/>
                        <legConfiguration name="pushTestcase1tc2Action"
                                          service="testService1"
```

```
                                                    action="tc2Action"
                                                    defaultMpc="defaultMpc"
                                                    reliability="AS4Reliability"
                                                    security="eSensPolicy"
receptionAwareness="receptionAwareness"
                                                    propertySet="ecodexPropertySet"
                                                    payloadProfile="MessageProfile"
errorHandling="demoErrorHandling"
                                                    compressPayloads="true"/>
            </legConfigurations>
<process name="tc1Process"
         agreement=""
         mep="oneway"
         binding="push"
         initiatorRole="defaultInitiatorRole"
         responderRole="defaultResponderRole">
      <initiatorParties>
            <initiatorParty name="blue_gw"/>
            <initiatorParty name="red_gw"/>
      </initiatorParties>
      <responderParties>
            <responderParty name="blue_gw"/>
            <responderParty name="red_gw"/>
      </responderParties>
      <legs>
            <leg name="pushTestcase1tc1Action"/>
            <leg name="pushTestcase1tc2Action"/>
      </legs>
</process>
      </businessProcesses>
</db:configuration>
```

6.3.4. Domibus PMode configuration to ebMS3 PMode Mapping

The following table provides additional information concerning the Domibus PMode configuration files.

Domibus PMode Configuration	EbMS3 Specification [ebMS3CORE] [AS4-Profile]	Description
MPCs	-	Container which defines the different MPCs (Message Partition Channels).
MPC	PMode[1].BusinessInfo.MPC: The value of this parameter is the identifier of the MPC (Message Partition Channel) to which the message is assigned. It maps to the attribute **Messaging** / **UserMessage**	Message Partition Channel allows the partition of the flow of messages from a *Sending MSH* to a *Receiving MSH* into several flows, each of which is controlled separately. An MPC also allows merging flows from several *Sending MSHs* into a unique flow that will be treated as such by a *Receiving MSH*. The value of this parameter is the identifier of the MPC to which the message is assigned.
MessageRetentionDownloaded	-	Retention interval for messages already delivered to the backend.
MessageRetentionUnDownloaded	-	Retention interval for messages not yet delivered to the backend.
Parties	-	Container which defines the different PartyIdTypes, Party and Endpoint.
PartyIdTypes	maps to the attribute **Messaging/UserMessage/ PartyInfo**	Message Unit bundling happens when the Messaging element contains multiple child elements or Units (either User Message Units or Signal Message Units).
Party ID	maps to the element **Messaging/UserMessage/ PartyInfo**	The ebCore Party ID type can simply be used as an identifier format and therefore as a convention for values to be used in configuration and – as such – does not require any specific solution building block.

Endpoint	maps to **PMode[1].Protocol.Address**	The endpoint is a party attribute that contains the link to the MSH. The value of this parameter represents the address (endpoint URL) of the *Receiver MSH* (or *Receiver Party*) to which Messages under this PMode leg are to be sent. Note that a URL generally determines the transport protocol (e.g. if the endpoint is an email address, then the transport protocol must be SMTP; if the address scheme is "http", then the transport protocol must be HTTP).
AS4	-	Container
Reliability [@Nonrepudiation] [@ReplyPattern]	Nonrepudiation maps to **PMode[1].Security.SendReceipt.NonRepudiation** ReplyPattern maps to **PMode[1].Security.SendReceipt.ReplyPattern**	PMode[1].Security.SendReceipt.NonRepudiation : value = 'true' (to be used for non-repudiation of receipt), value = 'false' (to be used simply for reception awareness). PMode[1].Security.SendReceipt.ReplyPattern: value = 'Response' (sending receipts on the HTTP response or back-channel). PMode[1].Security.SendReceipt.ReplyPattern: value = 'Callback' (sending receipts use a separate connection.)
ReceptionAwareness [@retryTimeout] [@retryCount] [@strategy] [@duplicateDetection]	retryTimeout maps to **PMode[1].ReceptionAwareness.Retry=true** PMode[1].ReceptionAwareness.Retry.Parameters retryCount maps to **PMode[1].ReceptionAwareness.Retry.Parameters** strategy maps to **PMode[1].ReceptionAwareness.Retry.Parameters** duplicateDetection maps to **PMode[1].ReceptionAwareness.DuplicateDetection**	These parameters are stored in a composite string. • *retryTimeout* defines timeout in seconds. • *retryCount* is the total number of retries. • *strategy* defines the frequency of retries. The only *strategy* available as of now is *CONSTANT*. • *duplicateDetection* allows to check duplicates when receiving twice the same message. The only *duplicateDetection* available as of now is *TRUE*.
Securities	-	Container
Security	-	Container
Policy	PMode[1].Security.* NOT including PMode[1].Security.X509.Signature.Algorithm	The parameter in the pconf file defines the name of a WS-SecurityPolicy file.
SignatureMethod	PMode[1].Security.X509.Signature.Algorithm	This parameter is not supported by WS-SecurityPolicy and therefore it is defined separately.
BusinessProcessConfiguration	-	Container

Agreements	maps to eb:Messaging/ UserMessage/ CollaborationInfo/ AgreementRef	This OPTIONAL element occurs zero times or once. The *AgreementRef* element is a string that identifies the entity or artifact governing the exchange of messages between the parties.
Actions	-	Container
Action	maps to **Messaging/ UserMessage/ CollaborationInfo/Action**	This REQUIRED element occurs once. The element is a string identifying an operation or an activity within a Service that may support several of these
Services	-	Container
ServiceTypes Type	maps to **Messaging/ UserMessage/ CollaborationInfo/ Service[@type]**	This REQUIRED element occurs once. It is a string identifying the service that acts on the message and it is specified by the designer of the service.
MEP [@Legs]	-	An ebMS MEP defines a typical choreography of ebMS User Messages which are all related through the use of the referencing feature (RefToMessageId). Each message of an MEP Access Point refers to a previous message of the same Access Point, unless it is the first one to occur. Messages are associated with a label (e.g. *request*, *reply*) that precisely identifies their direction between the parties involved and their role in the choreography.
Bindings	-	Container
Binding	-	The previous definition of ebMS MEP is quite abstract and ignores any binding consideration to the transport protocol. This is intentional, so that application level MEPs can be mapped to ebMS MEPs independently from the transport protocol to be used.
Roles	-	Container

Role	maps to **PMode.Initiator.Role or PMode.Responder.Role** depending on where this is used. In ebMS3 message this defines the content of the following element: • For Initiator: **Messaging/UserMessage/PartyInfo/From/Role** • For Responder: **Messaging/UserMessage/PartyInfo/To/Role**	The required role element occurs once, and identifies the authorized role (*fromAuthorizedRole* or *toAuthorizedRole*) of the Party sending the message (when present as a child of the *From* element), or receiving the message (when present as a child of the *To* element). The value of the role element is a non-empty string, with a default value of *http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/defaultRole* Other possible values are subject to partner agreement.
Processes	-	Container
PayloadProfiles	-	Container
Payloads	-	Container
Payload	maps to **PMode[1].BusinessInfo.PayloadProfile**	This parameter allows specifying some constraint or profile on the payload. It specifies a list of payload parts. A payload part is a data structure that consists of five properties: 1. **name** (or Content-ID) that is the **part identifier**, and can be used as an index in the notation PayloadProfile; 2. **MIME data type** (text/xml, application/pdf, etc.); 3. **name of the applicable XML Schema file** if the MIME data type is text/xml; 4. **maximum size in kilobytes**; 5. **Boolean** string indicating whether the part is **expected** or **optional**, within the User message. The message payload(s) must match this profile.
ErrorHandlings	-	Container
ErrorHandling	-	Container

ErrorAsResponse	maps to **PMode[1].ErrorHandling.Report.AsResponse**	This Boolean parameter indicates (if *true*) that errors generated from receiving a message in error are sent over the back-channel of the underlying protocol associated with the message in error. If *false*, such errors are not sent over the back-channel.
ProcessErrorNotifyProducer	maps to **PMode[1].ErrorHandling.Report.ProcessErrorNotifyProducer**	This Boolean parameter indicates whether (if *true*) the Producer (application/party) of a User Message matching this PMode should be notified when an error occurs in the Sending MSH, during processing of the *User Message to be sent*.
ProcessErrorNotifyConsumer	maps to **PMode[1].ErrorHandling.Report.ProcessErrorNotifyProducer**	This Boolean parameter indicates whether (if *true*) the Consumer (application/party) of a User Message matching this PMode should be notified when an error occurs in the Receiving MSH, during processing of the *received User message*.
DeliveryFailureNotifyProducer	maps to **PMode[1].ErrorHandling.Report.DeliveryFailuresNotifyProducer**	When sending a message with this reliability requirement (*Submit* invocation), one of the two following outcomes shall occur: - The Receiving MSH successfully delivers (*Deliver* invocation) the message to the Consumer. - The Sending MSH notifies (*Notify* invocation) the Producer of a delivery failure.
Legs	-	Container

Leg	-	Because messages in the same MEP may be subject to different requirements - e.g. the reliability, security and error reporting of a response may not be the same as for a request – the PMode will be divided into *legs*. Each user message label in an ebMS MEP is associated with a PMode leg. Each PMode leg has a full set of parameters for the six categories above (except for *General Parameters*), even though in many cases parameters will have the same value across the MEP legs. Signal messages that implement transport channel bindings (such as PullRequest) are also controlled by the same categories of parameters, except for *BusinessInfo group*.
Process	-	In *Process* everything is plugged together.

Table 2 - Domibus PMode configuration to ebMS3 mapping

6.3.5. Upload new Configuration

Upload the PMode file on both Access Points:

Remark:

In case the configuration is updated on one Access Point, all access points are informed about this change (via JMS topic).

1. To update the PMode configuration and/or Truststore, connect to the administration dashboard using your credentials (by default: User = ***admin***; Password = ***123456***) to http://*localhost*:*8080*/domibus/home

Remark:

It is recommended to change the passwords for the default users.
See §6.4.1 – "Administration Dashboard" for further information.

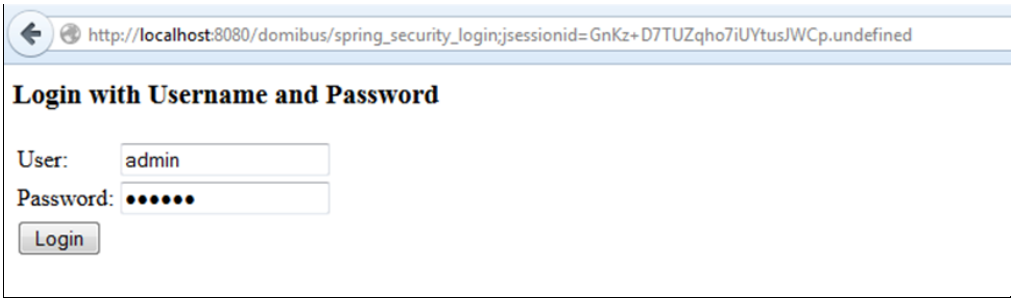


Figure 6 - Login to administration dashboard

2. Click on the **Configuration upload** tab:

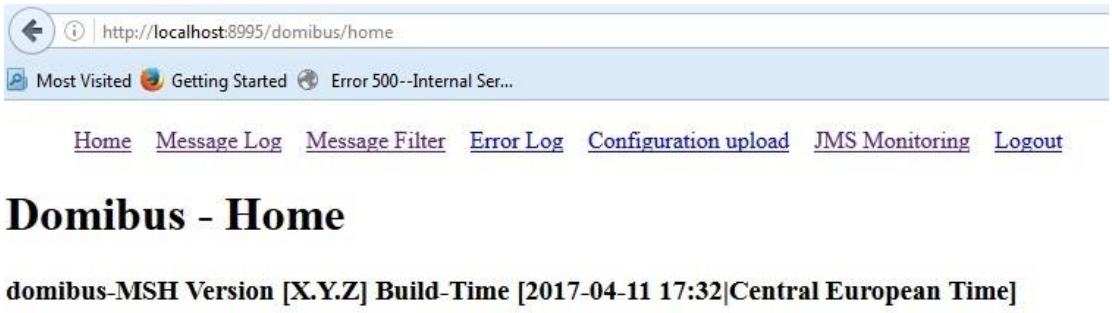


Figure 7 - Configuration upload

3. Select the PMode file that has been edited by pressing **Browse…** then **Press here to upload the Pmode xml file**:

Remark:

Each time a PMode is updated, the truststore is reloaded into the access point from the filesystem.

4. Select the Truststore file that needs to be uploaded by pressing **Browse**… then **Press here to upload the truststore jks file**

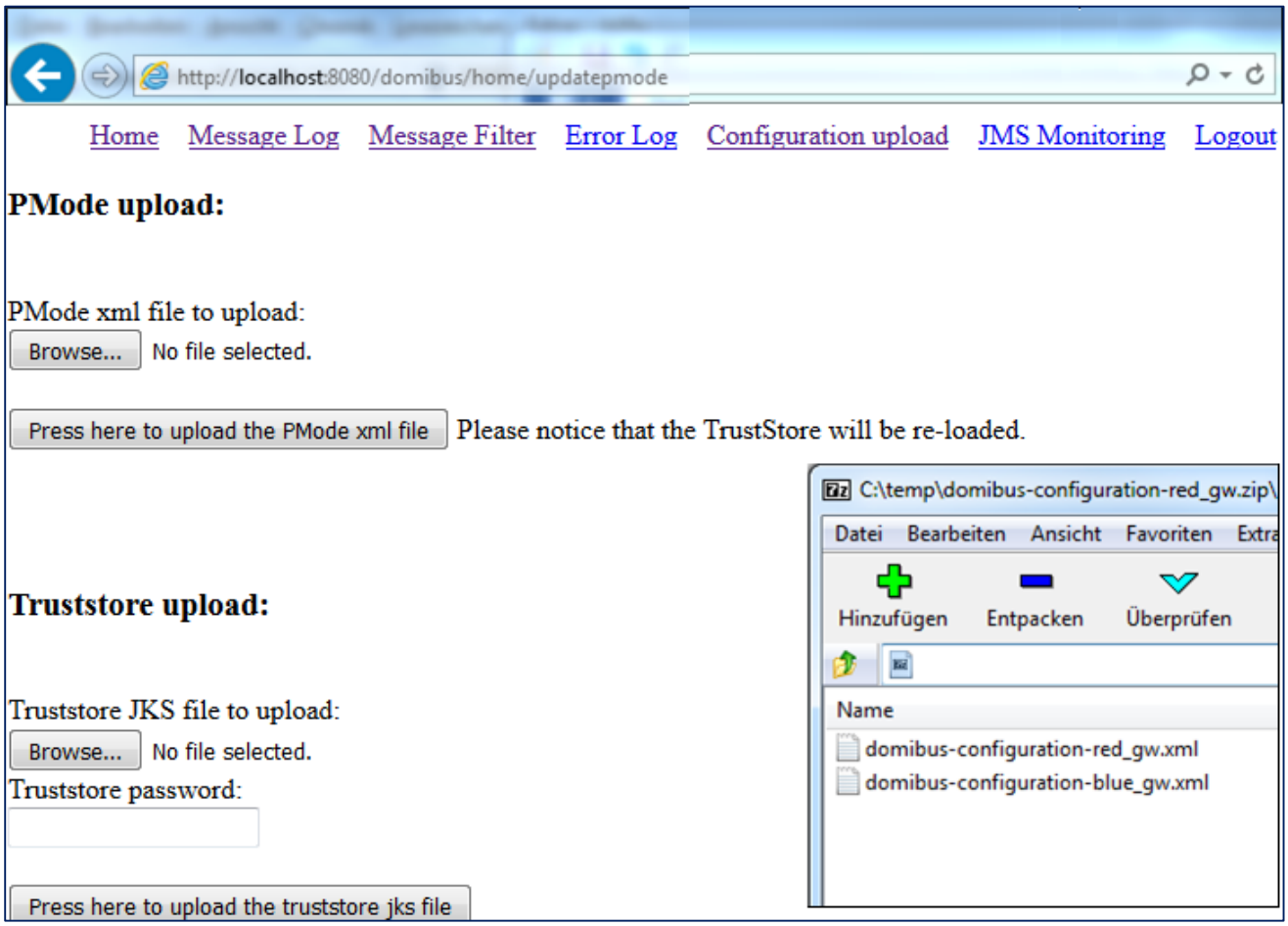


Figure 8 - PMode uploading

6.4. Administration Tools

6.4.1. Administration Dashboard

It is recommended to change the passwords for the default users which have access to the Domibus Administration page: ***admin*** and ***user***.

In order to change the password, please use a BCrypt strong hashing algorithm to generate your custom password. You can use an online BCrypt password generator (e.g. https://www.bcrypt-generator.com/)

Once you have the hashed password please modify the passwords for the default users (***admin*** and ***user***) in the file *cef_edelivery_path***/domibus/conf/domibus/domibus-security.xml**:

```
<sec:authentication-manager>
    <sec:authentication-provider>
        <sec:password-encoder ref="bcryptEncoder"/>
        <sec:user-service>
            <sec:user name="user" password="your_custom_user_password"
                      authorities="ROLE_USER"/>
            <sec:user name="admin" password="your_custom_admin_password"
                      authorities="ROLE_USER,ROLE_ADMIN"/>
        </sec:user-service>
    </sec:authentication-provider>
</sec:authentication-manager>
```

Remark:

> *In February 2014 a new revision of the bcrypt algorithm was published which now generates bcrypt hashes that are not yet supported by the Spring Security version used by Domibus. (https://github.com/spring-projects/spring-security/issues/3320).*
> *You need then to take the generated hash and change the prefix of the generated value from $2**y**$10$ to $2**a**$10$.*

Other user credentials than **admin** and **user** can be added. In some big companies, security policies disallow to use share or generic accounts. It could be useful to clearly identify who as updated that PMode file.

e.g:

We can access the Domibus Web Console using the username **user2** by adding the credentials as follow:

```
<sec:user-service>
   <sec:user name="user"
           password="$2a$10$uhZqXZdgBKMzwLaMI.rFGeOIwpTOkt4CfZ9rYQPC12rGKWe3Q8MKe"
           authorities="ROLE_USER"/>
   <sec:user name="admin"
           password="$2a$10$ZfMUsy7BD49l81xUHN0bbOl8yvw8/FWt/NqGTdNffB9z7t8mqlSxq"
           authorities="ROLE_USER, ROLE_ADMIN"/>
   <sec:user name="user2"
           password="$2a$10$ZfMUsy7BD49l81xUHN0bbOl8yvw8/FWt/NqGTdNffB9z7t8mqlSxq"
           authorities="ROLE_USER, ROLE_ADMIN"/>
</sec:user-service>
```

Domibus administration dashboard includes a message logging page that gives the administrator information related to send messages, received messages and their status (SENT, RECEIVED, FAILED, ACKNOWLEDGE…)

The following state machines illustrate the evolution of the processing of messages according to the encountered events:

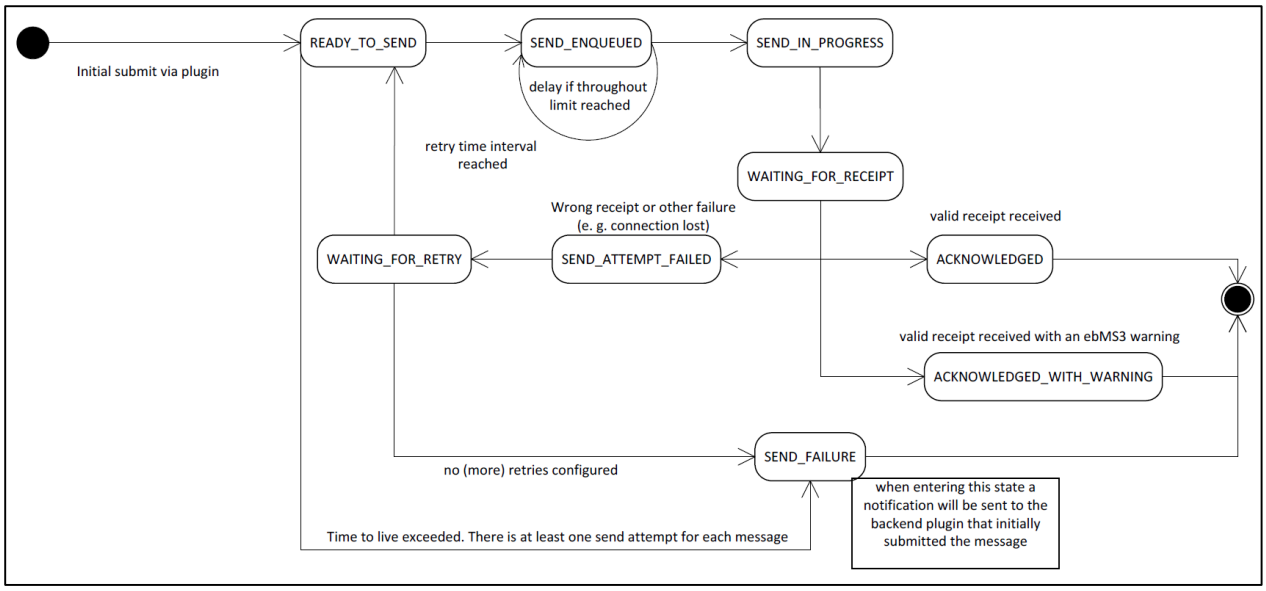


Figure 9 - State machine of Corner 2 (sending access point)

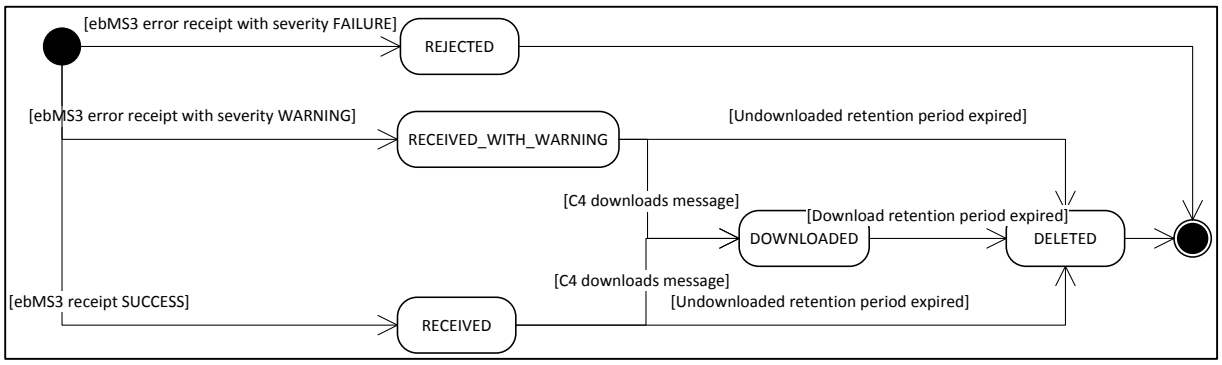


Figure 10 - State machine of Corner 3 (receiving access point)

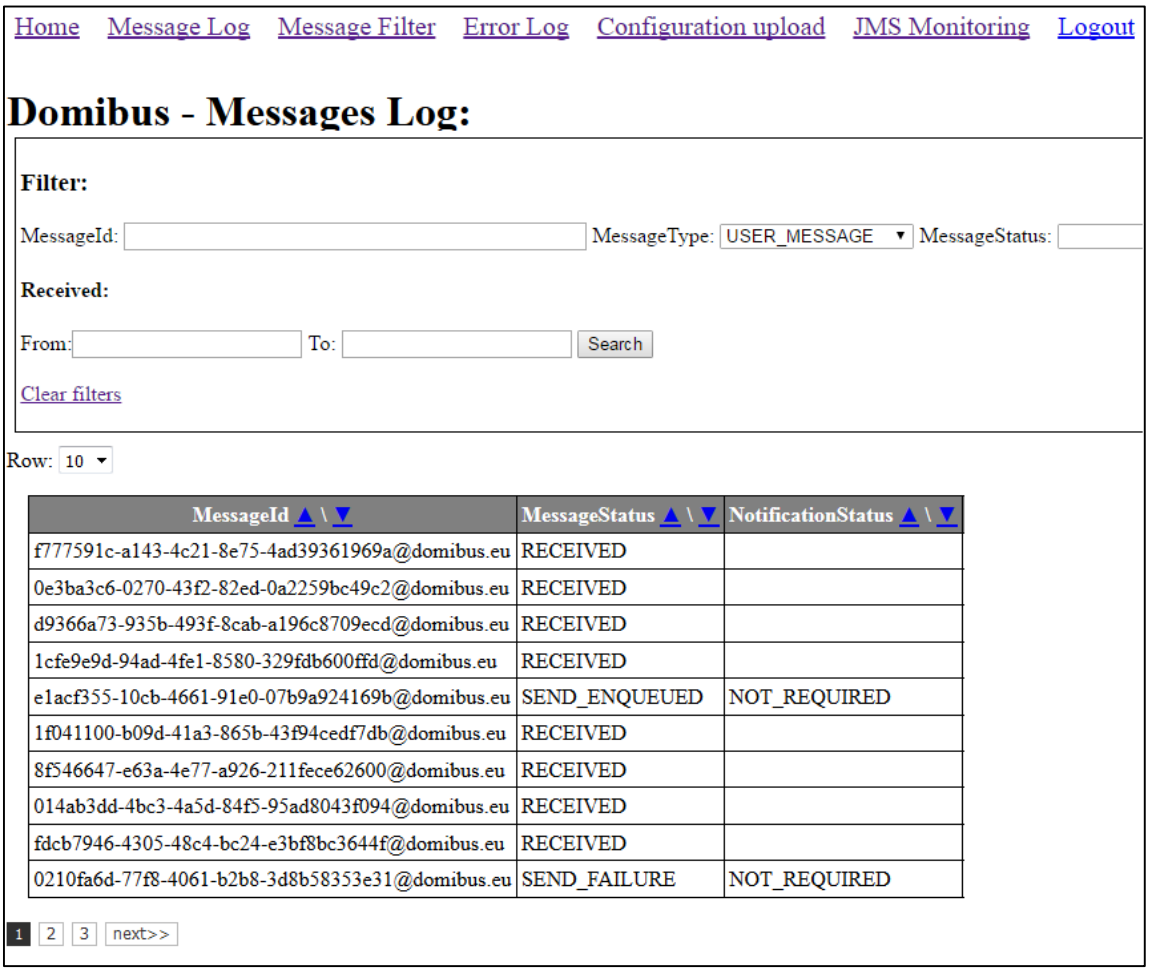


Figure 11 - Domibus Message Log

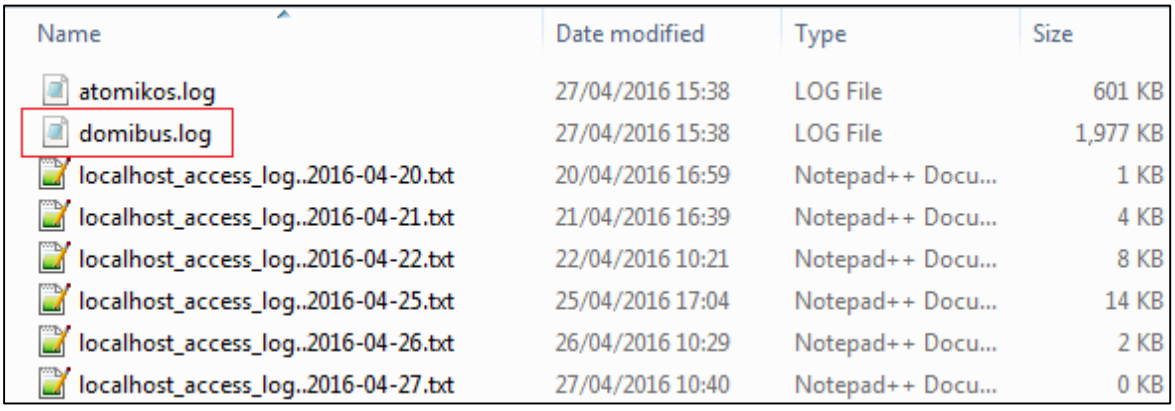
Remark:

The administration dashboard is reachable via the URL:
http**://***your_server*:*your_port_number*/domibus/home (Tomcat)
http**://***your_server*:*your_port_number*/domibus-wildfly/home (WildFly)
http**://***your_server*:*your_port_number*/domibus-weblogic/home (WebLogic)

6.4.2. Application Logging

6.4.2.1. Domibus log file

The file *cef_edelivery_path***/domibus/logs/domibus.log** contains errors encountered by the application. The file contains information related to internal errors thrown if the application fails to process an incoming or an outgoing message.

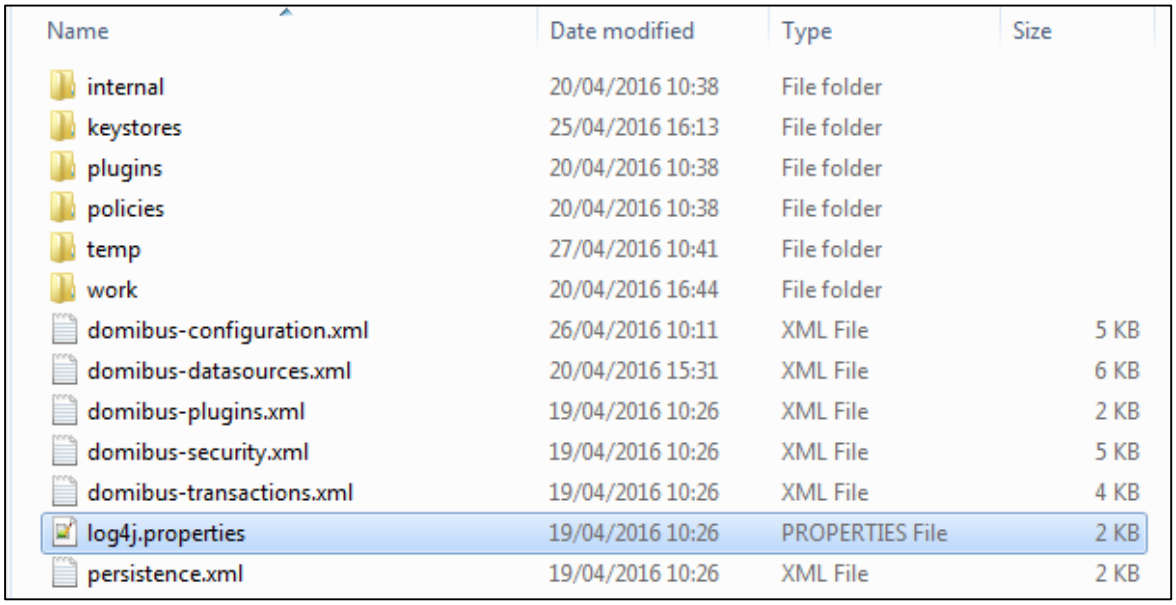


Remark:

> *The response of the application to your client's request might also contain information about the errors encountered, depending on the root cause of the issue (e.g. if the header of the request is not compliant with your PMode the message error will be included in the soap response. On the other hand if the error is related to the security protocol, the information will be included in Domibus.log only).*

6.4.2.2. Logging properties

It is possible to modify the configuration of the logs by editing the logging properties in the file *cef_edelivery_path***/domibus/conf/domibus/log4j.properties**:



Name	Date modified	Type	Size
internal	20/04/2016 10:38	File folder	
keystores	25/04/2016 16:13	File folder	
plugins	20/04/2016 10:38	File folder	
policies	20/04/2016 10:38	File folder	
temp	27/04/2016 10:41	File folder	
work	20/04/2016 16:44	File folder	
domibus-configuration.xml	26/04/2016 10:11	XML File	5 KB
domibus-datasources.xml	20/04/2016 15:31	XML File	6 KB
domibus-plugins.xml	19/04/2016 10:26	XML File	2 KB
domibus-security.xml	19/04/2016 10:26	XML File	5 KB
domibus-transactions.xml	19/04/2016 10:26	XML File	4 KB
log4j.properties	19/04/2016 10:26	PROPERTIES File	2 KB
persistence.xml	19/04/2016 10:26	XML File	2 KB

In the example below, you can see the contents of the **log4j.properties** file:

```
# Direct log messages to stdout
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target=System.out
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L - %m%n

log4j.appender.file=org.apache.log4j.FileAppender
log4j.appender.file.file=${catalina.home}/logs/domibus.log
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L - %m%n


log4j.appender.atomikos=org.apache.log4j.FileAppender
log4j.appender.atomikos.file=${catalina.home}/logs/atomikos.log
log4j.appender.atomikos.layout=org.apache.log4j.PatternLayout
log4j.appender.atomikos.layout.ConversionPattern=%d{ABSOLUTE} %5p %c{1}:%L - %m%n

# In order to enable logging of request/responses please change the loglevel to INFO
log4j.logger.org.apache.cxf=WARN
# Root logger option
log4j.rootLogger=INFO, file, stdout

log4j.logger.com.atomikos=WARN, atomikos
```

Figure 12 - log4j.poperties

- In red: these parameters can be edited to modify the location of the log file, and the layout.
- In green: these parameters can be edited to change the level of logging (3 levels definied: INFO, WARN, and ERROR).

6.4.2.3. Error Log page

This option lists all the error logs related to Message Transfers and includes the **ErrorSignalMessageId**, **ErrorDetail** and **Timestamp**. The messages can be sorted by clicking on the up and down arrows which helps to search for specific messages.

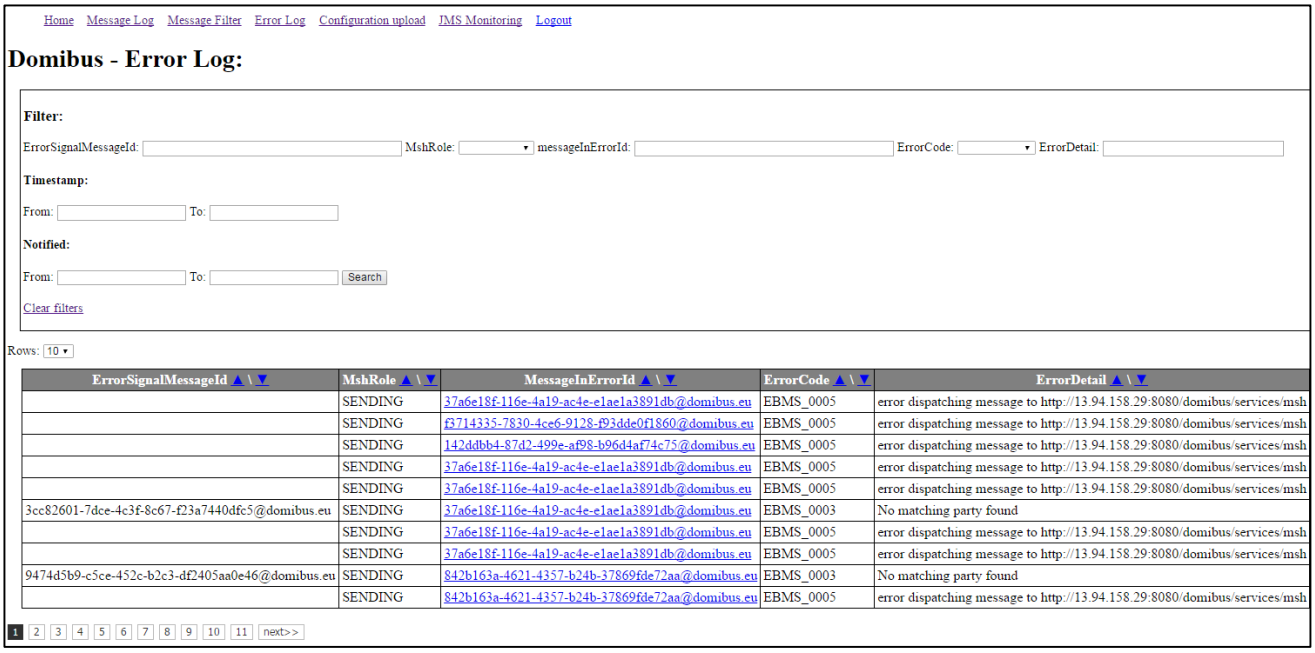


Figure 13 - Domibus – Error Log page

6.4.3. Queue Monitoring

Domibus uses JMS queues to handle the messages:

Destination type	JNDI name	Comment	Description
Queue	jms/domibus.internal.dispatch.queue	No redelivery because redelivery of MSH messages is handled via ebMS3/AS4	This queue is used for scheduling messages for sending via the MSH
Queue	jms/domibus.internal.notification.unknown		Notifications about received messages (by the MSH) that do not match any backend routing criteria will be sent to this queue. In production environment this queue should be monitored in order to handle those messages manually
Topic	jms/domibus.internal.command		This topic is used for sending commands to all nodes in a cluster. For example, it is used after a PMode was uploaded in order to notify all nodes to update their PMode cache (in case caching is enabled)
Queue	jms/domibus.backend.jms.replyQueue		This queue is used for sending replies back to the sender of a message. Replies contain: a correlationId, ebMS3 messageId (if possible), error messages (if available)
Queue	jms/domibus.backend.jms.outQueue		Messages received by the MSH (that match the routing criteria for the JMS plugin) will be sent to this queue
Queue	jms/domibus.backend.jms.inQueue		This queue is the entry point for messages to be sent by the sending MSH
Queue	jms/domibus.backend.jms.errorNotifyConsumer		This queue is used to inform the receiver of a message that an error occurred during the processing of a received message
Queue	jms/domibus.backend.jms.errorNotifyProducer		This queue is used to inform the sender of a message that an error occurred during the processing of a message to be sent

Queue	jms/domibus.notification.jms		Used for sending notifications to the configured JMS plugin
Queue	jms/domibus.internal.notification.queue		This queue is used to notify the configured plugin about the status of the message to be sent
Queue	jms/domibus.notification.webservice		Used for sending notifications to the configured WS plugin
Queue	jms/domibus.DLQ		This is the Dead Letter Queue of the application. The messages from other queues that reached the retry limit are redirected to this queue

Table 3 - Queue Monitoring

All these queues can be monitored and managed using the **JMS Monitoring** page.

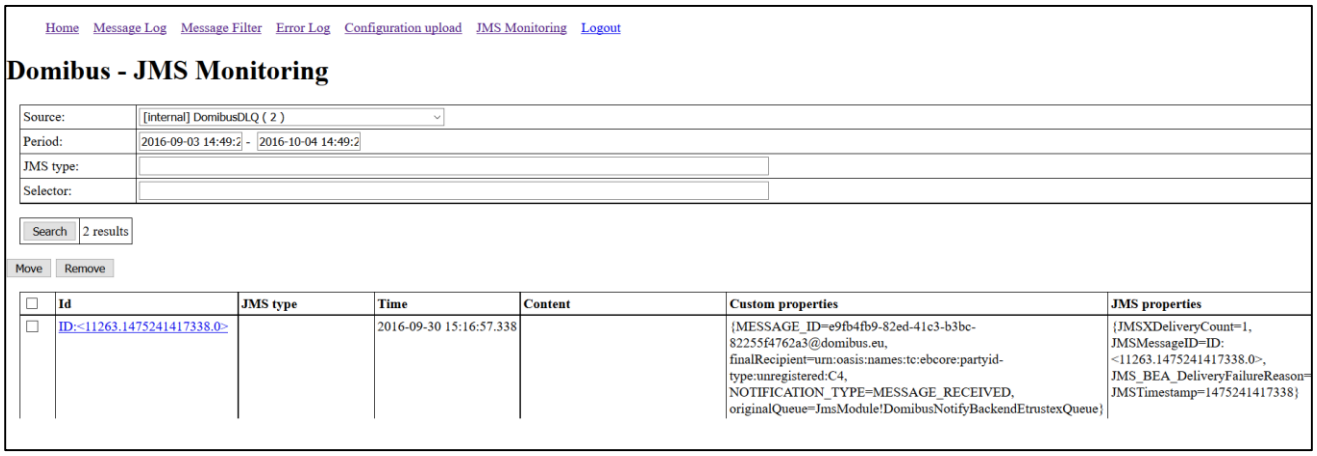


Figure 14 - Domibus – JMS Monitoring page

In the **Source** field, we have all the queues listed along with the number of messages pending in each queue.
If a queue is used internally by the application core, its name will start with **[internal].** A regular expression is used to identify all the internal queues. The value for this regular expression can be adapted in the property **domibus.jms.internalQueue.expression** from the file *cef_edelivery_path***/conf/domibus/domibus-configuration.xml**

In the **JMS Monitoring** page the following operations can be performed:

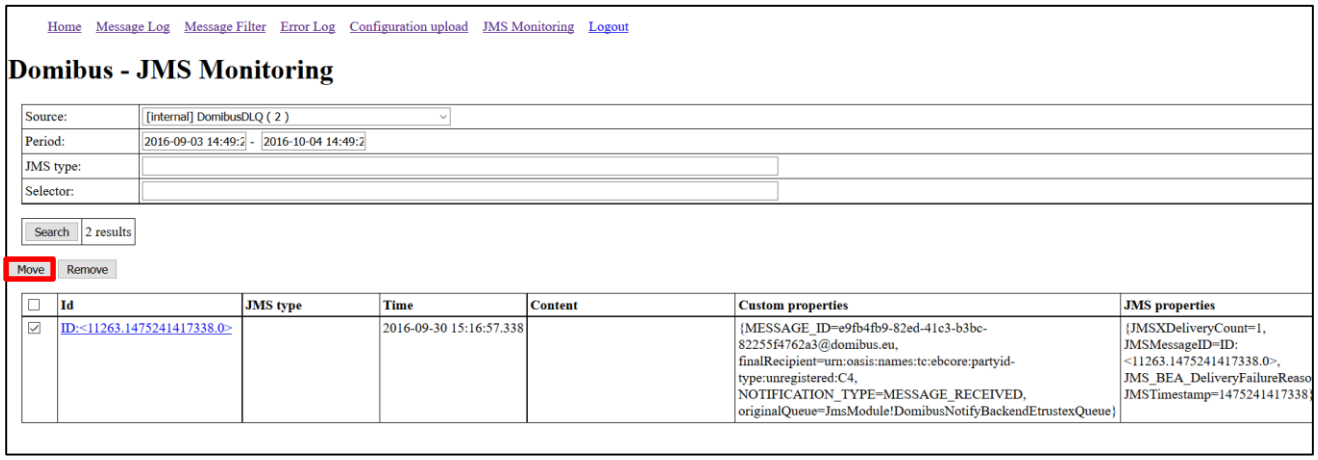
1. Inspecting and filtering the messages from a queue based on the following fields:
    a. Source: the source queue of the messages
    b. Period: time interval that will filter the messages based on the send date
    c. JMS type: the JMS header **JMSType**
    d. Selector: the JMS message selector expression

Remark:

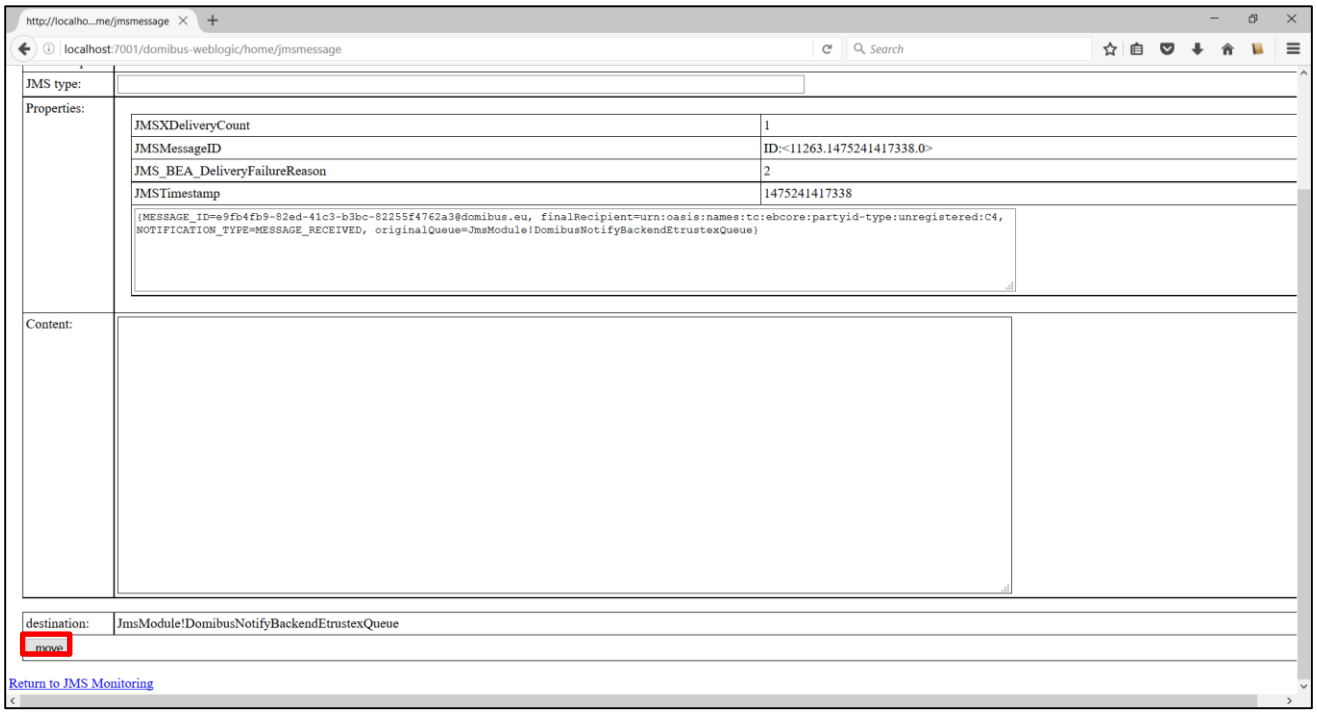
For more info on the JMS message headers and on the JMS message selector, please check the official documentation https://docs.oracle.com/cd/E19798-01/821-1841/bnces/index.html

2. Move message

a. Move a message from the DLQ to the original queue:
   - Select a JMS message from the DLQ and press the **Move** button

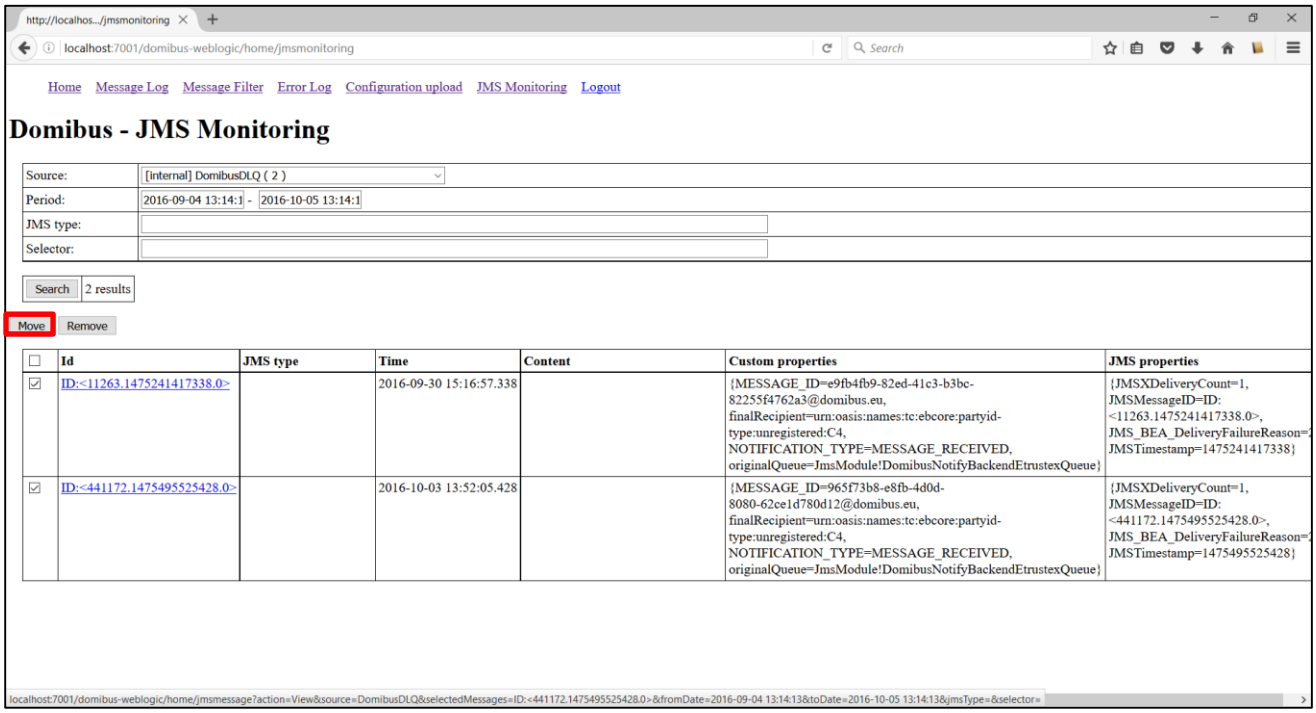


   - The message details are displayed and the original queue where the message came from is pre-selected
   - Press the **Move** button and the message will be move to the original queue



b. Move multiple messages from the DLQ to the original queue

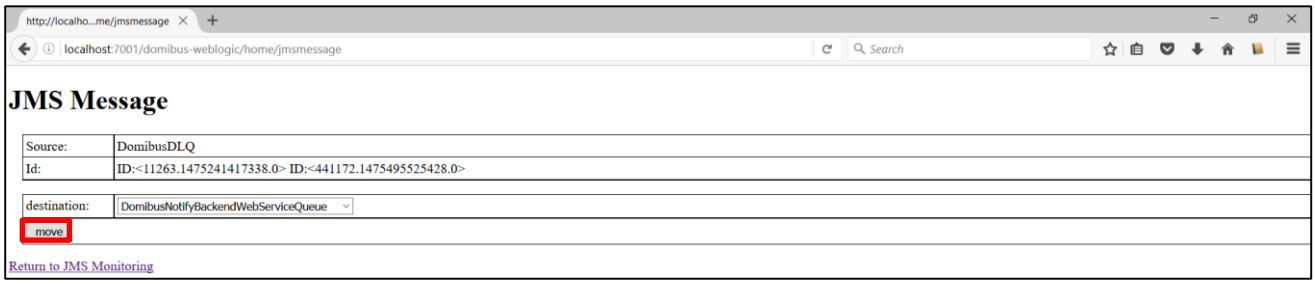
- Select multiple JMS message from the DLQ and press the **Move** button



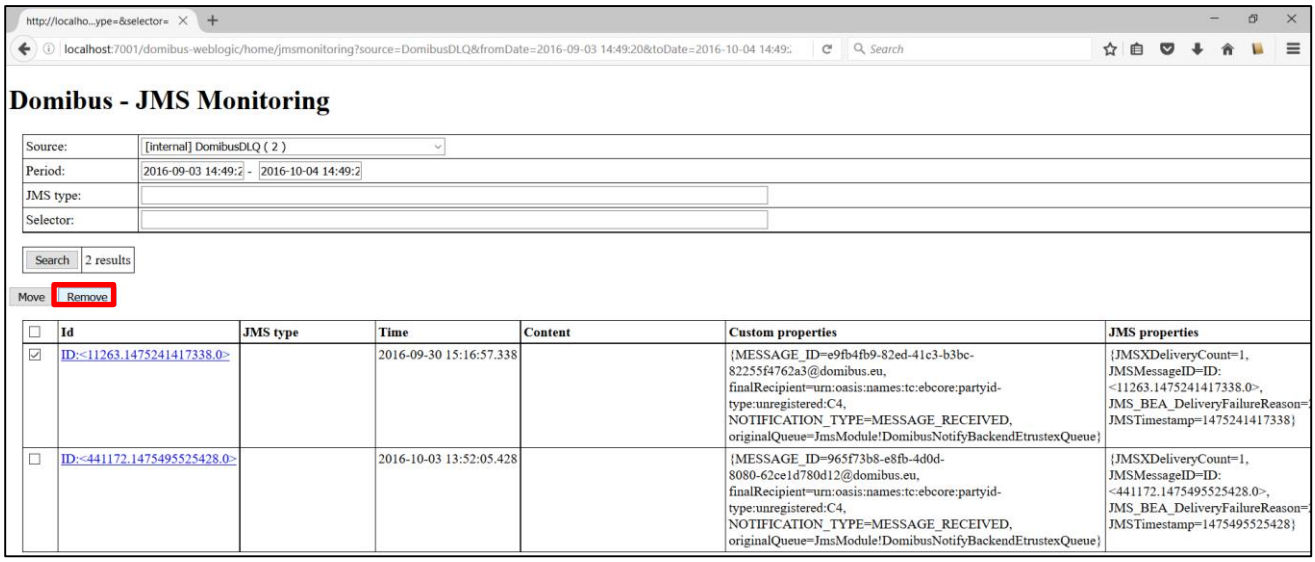
- The messages ID are displayed in the "Id:" field and all the available queues are displayed for selection in a drop down list in the "destination" field;
- Select the destination queue from the dropdown list and press **move.** All the previously selected JMS messages will be moved to the selected destination queue

Remark:

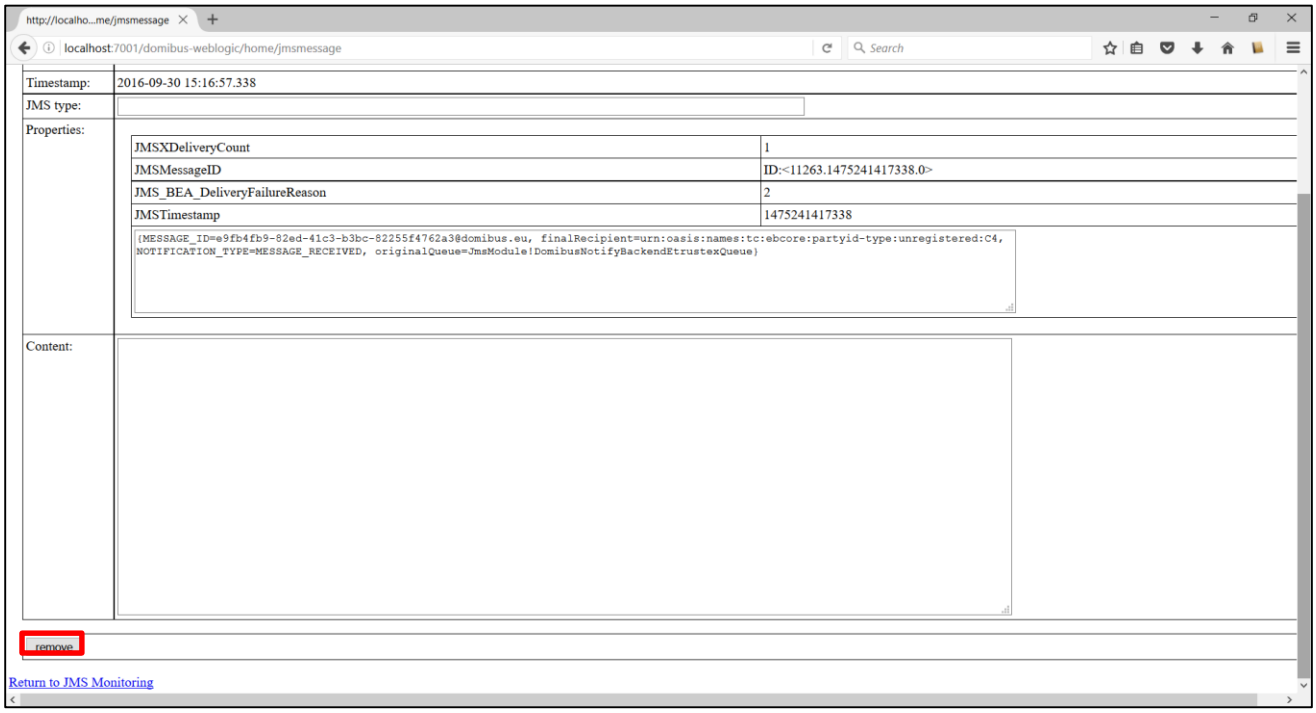
Please make sure that all the selected messages came from the same source queue. Use the filtering capabilities to ensure this.



3. Delete message
    a. Delete a message from a queue:
        - Select a JMS message from the source queue and press the **Remove** button

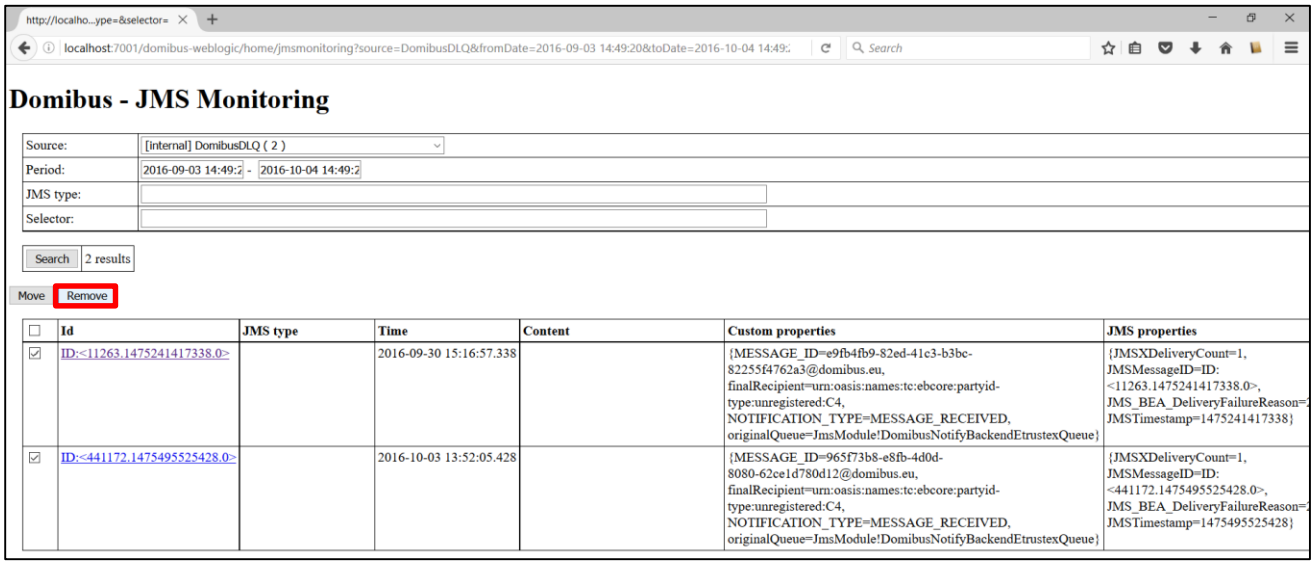


        - The message details are displayed
        - Press the **Remove** button to remove it

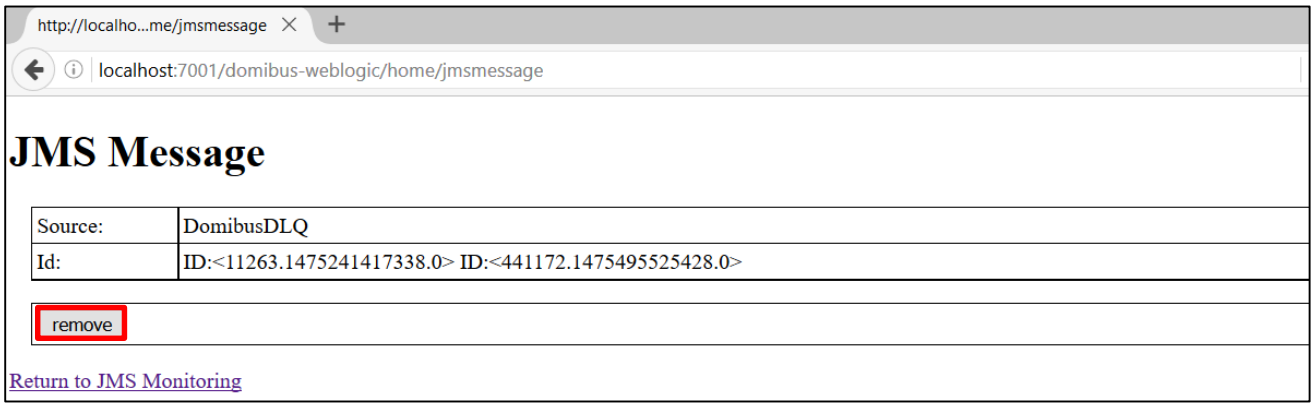


b. Delete multiple messages from a queue:

- Select multiple JMS message from the source queue and press the **Remove** button



- The messages ID to be removed are displayed
- Press the **Remove** button to remove them



6.4.4. Configuration of the queues

Queues should be configured appropriately and according to the backend system needs and re-delivery policy.

6.4.4.1. Tomcat

Domibus uses ActiveMQ as JMS broker. The various queues are configured in the *cef_edelivery_path***/domibus/conf/domibus/internal/activemq.xml** file.

Please see ActiveMQ redelivery policy and configure the parameters below:

```
<redeliveryPlugin fallbackToDeadLetter="true"
                  sendToDlqIfMaxRetriesExceeded="true">
    <redeliveryPolicyMap>
        <redeliveryPolicyMap>
            <redeliveryPolicyEntries>
                <redeliveryPolicy queue="sendMessageQueue"
                                  maximumRedeliveries="0"/>
     <redeliveryPolicy queue="*"
                  maximumRedeliveries="10" redeliveryDelay="300000"/>
            </redeliveryPolicyEntries>
        </redeliveryPolicyMap>
    </redeliveryPolicyMap>
</redeliveryPlugin>
    </redeliveryPolicyMap>
</redeliveryPlugin>
```

Access to the JMS messaging subsystem is protected by a username and password in clear text in the *cef_edelivery_path***/domibus/conf/domibus/internal/activemq.xml'** and *cef_edelivery_path***/domibus/conf/domibus/domibus-datasources.xml'** files.

It is recommended to change the password for the default user:

Remark:

> *If you change the default password, it needs to be changed in both activemq.xml and Domibus-datasources.xml.*

In the *cef_edelivery_path***/domibus/conf/domibus/internal/activemq.xml:**

```
<simpleAuthenticationPlugin anonymousAccessAllowed="false">
  <users>
    <authenticationUser username="domibus" password="changeit" groups="users"/>
    <authenticationUser username="admin" password="123456" groups="admins,users"/>
  </users>
</simpleAuthenticationPlugin>
```

In the *cef_edelivery_path***/domibus/conf/domibus/domibus-datasources.xml:**

```
<amq:xaConnectionFactory id="xaJmsConnectionFactory"
brokerURL="tcp://localhost:63616" userName="domibus" password="changeit">
<!-- do not remove this! otherwise the redeliveryPolicy configured in activemq.xml
will be ignored -->
  <amq:redeliveryPolicy>
    <amq:redeliveryPolicy />
  </amq:redeliveryPolicy>
</amq:xaConnectionFactory>
```

6.4.4.2. WebLogic

Please use the admin console of WebLogic to configure the re-delivery limit and delay.

6.4.4.3. WildFly

Please use the admin console of WildFly to configure the re-delivery limit and delay.

6.4.5. Message Filtering

In case there are multiple plugins registered, Domibus will route the incoming message to the first plugin in the list, by default.

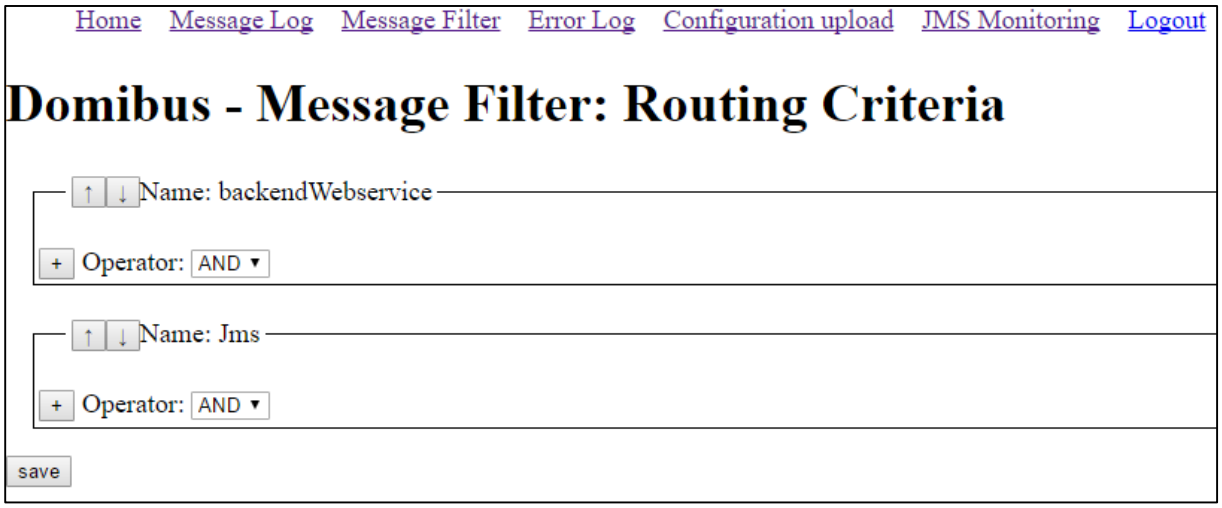


Figure 15 - Domibus – Message filter page

In the configuration above:

- The incoming messages will be routed to the **backendWebservice** (name of the default WS Plugin).
- A plugin can be configured to treat a subset of incoming messages according to 4 criterias: **action**, **service**, **to** and **from**.
- All the messages will normally be routed to the **backendWebservice** plugin, with no messages routed to **Jms**, unless filtered out by some criteria (at **backendWebservice**).
- Messages that do match any of the plugins criteria's are then routed to the **unknownReceiverQueue.**
- In any case, plugins should be configured with the correct filtering criteria to ensure that all messages are treated (**unknownReceiverQueue** should be 0) and each message only consumed by a single plugin.

Four fields are available to use in the filtering criteria: **Action**, **Service**, **From** and **To** as well as the logical operators **AND** and **OR**.

An example of the routing fields is shown below:

- **From** : *domibus-party_id_name1:urn:oasis:names:tc:ebcore:partyid-type:unregistered*
- **To** : *domibus-party_id_name2:urn:oasis:names:tc:ebcore:partyid-type:unregistered*
- **Service** : *bdx:noprocess:tc1*
- **Action** : *TC1Leg1.*

The values above will be matched against the corresponding calculated values from the incoming message received by Domibus (e.g. Soap request extract, below)

```
<ns:PartyInfo>
                <ns:From>
                   <ns:PartyId type="urn:oasis:names:tc:ebcore:partyid-
type:unregistered">party_id_name1</ns:PartyId>
                   <ns:Role>http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/ns/core/200704/initiator</ns:Role>
                </ns:From>
                <ns:To>
                   <ns:PartyId type="urn:oasis:names:tc:ebcore:partyid-
type:unregistered">party_id_name2</ns:PartyId>
                   <ns:Role>http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/ns/core/200704/responder</ns:Role>
                </ns:To>
             </ns:PartyInfo>
             <ns:CollaborationInfo>
                <ns:Service type="tc1">bdx:noprocess</ns:Service>
                <ns:Action>TC1Leg1</ns:Action>
             </ns:CollaborationInfo>
```

7. DATA ARCHIVING

7.1. What's archiving?

Data archiving is the method of moving message that have been processed successfully or unsuccessfully by the access point to an external storage location for long-term retention.

Archiving data involves older data that have been processed at the communication level by the access points but that is still significant to the business and may be needed for future reference, or data that must be retained for legal constraints.

Data archives are indexed and searchable to allow easy retrieval,

It is not recommended to use Domibus as an archiving solution. Nevertheless, if it is really needed to keep the data, it is possible to set the Data Retention Policy so the data can be extracted from the database through the webservices or by an external archiving tool.

7.2. Data Retention Policy

A data retention policy is a business's established procedure for continuous information storage for operational, legal or compliance reasons

The data retention policy needs to be defined based on the business needs and constraints.

In Domibus, the data retention policy can be found here in the PMode file:

```
<mpcs>
   <mpc name="defaultMpc"
        qualifiedName="http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/ns/core/200704/defaultMPC"
        enabled="true"
        default="true"
        retention_downloaded="0"
        retention_undownloaded="14400"/>
</mpcs>
```

In the sample PMode configuration of Domibus, the data retention policy is set to **14400 seconds** (4 hours) if the message is not downloaded. This means that if the message is not downloaded, it will be deleted then only the metadata containing the information of the receiver and the acknowledgement.

The data retention policy is also set to **0 seconds** if the message is downloaded. This means that the message will be instantaneously deleted as soon as it is downloaded. Those two parameters can be configured to meet the needs of the business.

7.3. Data Extraction

In order to keep the metadata and the payload of the message for a defined amount of time, that exceeds the one set in the PMode, it is recommended to extract it to an external storage. As long as the retention worker does not delete it, data can be extracted through the webservices or through an external archiving tool.

For more information, please refer to the Data Model provided in the Domibus Software Architecture Document.

8. NON REPUDIATION

In order to guarantee non-repudiation, the sending Access Point (C2) is storing the full **SignalMessage**, including the **MessageInfo** and the Receipt (that contains the **NonRepudiationInformation** for each part) and the signature of the receipt by the receiver Access Point (C3).

This will guarantee that the receiver Access Point (C3) of the message cannot deny having received a message from the sender Access Point (C2) at the moment of the sending. However if the initial sender (C1) wants to be sure that the final recipient (C4) cannot deny having received a specific content inside this message the sender must be able to show the specific content that was used to produce the receiver Access Point's (C3) signature.

Domibus, as a sending Access Point (C2), keeps track of the metadata of the sent messages but does not store the message payloads itself. Therefore, it is recommended that the initial sender (C1) stores the message payloads safely for the time needed to guarantee non-repudiation of the sent messages.

In order to guarantee non-repudiation, the receiving Access Point (C3) is storing the full UserMessage and the associated signature of the sender (C2).

This will guarantee that the sender Access Point (C2) of the message cannot deny having sent a message to the receiver at the moment of the sending. However if the final recipient (C4) wants to be sure that the sender cannot deny having sent a specific content inside this message the final recipient (C4) must be able to show the specific content that was used to produce the sender Access Point 's signature (C2).

Domibus, as a receiving Access Point (C3), keeps track of the metadata of the received messages and stores the message payloads itself only for the (limited) duration configured in the retention period (specified in the PMode). Therefore, it is recommended that the final recipient (C4) either stores the message payloads safely or aligns the retention period on the receiving Access Point (C3) with the time needed to guarantee non-repudiation of the received messages.

9. TROUBLESHOOTING

9.1. Failed to obtain DB connection from datasource

```
SEVERE: Exception sending context initialized event to listener instance of class
org.springframework.web.context.ContextLoaderListener
org.springframework.beans.factory.BeanCreationException: Error creating bean with
name 'org.springframework.scheduling.quartz.SchedulerFactoryBean#0' defined in
ServletContext resource [/WEB-INF/msh-config.xml]: Invocation of init method
failed; nested exception is org.quartz.JobPersistenceException: Failed to obtain
DB connection from datasource
'springTxDataSource.org.springframework.scheduling.quartz.SchedulerFactoryBean#0':
com.atomikos.jdbc.AtomikosSQLException: Failed to grow the connection pool [See
nested exception: com.atomikos.jdbc.AtomikosSQLException: Failed to grow the
connection pool]
        at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.initi
alizeBean(AbstractAutowireCapableBeanFactory.java:1578)
        at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.doCre
ateBean(AbstractAutowireCapableBeanFactory.java:545)
        at
org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory.creat
eBean(AbstractAutowireCapableBeanFactory.java:482)
        at
org.springframework.beans.factory.support.AbstractBeanFactory$1.getObject(Abstract
BeanFactory.java:305)
        at
org.springframework.beans.factory.support.DefaultSingletonBeanRegistry.getSingleto
n(DefaultSingletonBeanRegistry.java:230)
        at
org.springframework.beans.factory.support.AbstractBeanFactory.doGetBean(AbstractBe
anFactory.java:301)
SEVERE: One or more listeners failed to start. Full details will be found in the
appropriate container log file
May 11, 2016 10:12:43 AM org.apache.catalina.util.SessionIdGeneratorBase
createSecureRandom
INFO: Creation of SecureRandom instance for session ID generation using [SHA1PRNG]
took [13,256] milliseconds.
May 11, 2016 10:12:43 AM org.apache.catalina.core.StandardContext startInternal
SEVERE: Context [/domibus] startup failed due to previous errors
May 11, 2016 10:12:43 AM org.apache.catalina.core.ApplicationContext log
INFO: Closing Spring root WebApplicationContext
May 11, 2016 10:12:43 AM org.apache.catalina.core.ApplicationContext log
INFO: Shutting down log4j
```

Solution: *Setup the password properly in the* ***domibus-datasources.xml***

9.2. Exception sending context initialized event to listener instance of class

```
SEVERE: Exception sending context initialized event to listener instance of class
org.springframework.web.context.ContextLoaderListener
org.springframework.beans.factory.BeanCreationException: Error creating bean with
name 'entityManagerFactory' defined in URL
[file:////home/edelivery/domibusf1/conf/domibus/domibus-datasources.xml]: Cannot
resolve reference to bean 'domibusJDBC-XADataSource' while setting bean property
'dataSource'; nested exception is
org.springframework.beans.factory.BeanCreationException: Error creating bean with
name 'domibusJDBC-XADataSource' defined in URL
[file:////home/edelivery/domibusf1/conf/domibus/domibus-datasources.xml]:
Invocation of init method failed; nested exception is
com.atomikos.jdbc.AtomikosSQLException: The class
'com.mysql.jdbc.jdbc2.optional.MysqlXADataSource' specified by property
'xaDataSourceClassName' could not be found in the classpath. Please make sure the
spelling is correct, and that the required jar(s) are in the classpath.
```

Solution: *Add MySQL connector in domibus/lib folder*

9.3. Neither the JAVA_HOME nor the JRE_HOME environment variable is defined

```
Neither the JAVA_HOME nor the JRE_HOME environment variable is defined
At least one of these environment variable is needed to run this program
```

Solution: *Set JAVA_HOME variable or/and JRE_HOME*

9.4. Cannot access Admin Console

```
http://your_server:your_port_number/domibus/home
No SEVER errors in logs but no admin option in browser under
```

Solution: *Check if the firewall is open for port_no (e.g. 8080).*

9.5. Handshake Failure

Full stack trace below:

```
org.apache.cxf.interceptor.Fault: Could not write attachments.
at
org.apache.cxf.interceptor.AttachmentOutInterceptor.handleMessage(AttachmentOutInt
erceptor.java:74)
at
org.apache.cxf.phase.PhaseInterceptorChain.doIntercept(PhaseInterceptorChain.java:
308)
at org.apache.cxf.endpoint.ClientImpl.doInvoke(ClientImpl.java:514)
at org.apache.cxf.endpoint.ClientImpl.invoke(ClientImpl.java:423)
at org.apache.cxf.endpoint.ClientImpl.invoke(ClientImpl.java:324)
```

```
at org.apache.cxf.endpoint.ClientImpl.invoke(ClientImpl.java:277)
at org.apache.cxf.endpoint.ClientImpl.invokeWrapped(ClientImpl.java:312)
at org.apache.cxf.jaxws.DispatchImpl.invoke(DispatchImpl.java:327)
at org.apache.cxf.jaxws.DispatchImpl.invoke(DispatchImpl.java:246)
at eu.domibus.ebms3.sender.MSHDispatcher.dispatch(MSHDispatcher.java:126)
at
eu.domibus.ebms3.sender.MSHDispatcher$$FastClassBySpringCGLIB$$105974a1.invoke(<ge
nerated>)
at org.springframework.cglib.proxy.MethodProxy.invoke(MethodProxy.java:204)
at
org.springframework.aop.framework.CglibAopProxy$CglibMethodInvocation.invokeJoinpo
int(CglibAopProxy.java:717)
at
org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMet
hodInvocation.java:157)
at
org.springframework.transaction.interceptor.TransactionInterceptor$1.proceedWithIn
vocation(TransactionInterceptor.java:99)
at
org.springframework.transaction.interceptor.TransactionAspectSupport.invokeWithinT
ransaction(TransactionAspectSupport.java:281)
at
org.springframework.transaction.interceptor.TransactionInterceptor.invoke(Transact
ionInterceptor.java:96)
at
org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMet
hodInvocation.java:179)
at
org.springframework.aop.framework.CglibAopProxy$DynamicAdvisedInterceptor.intercep
t(CglibAopProxy.java:653)
at
eu.domibus.ebms3.sender.MSHDispatcher$$EnhancerBySpringCGLIB$$da53e95a.dispatch(<g
enerated>)
at eu.domibus.ebms3.sender.MessageSender.sendUserMessage(MessageSender.java:116)
at eu.domibus.ebms3.sender.MessageSender.onMessage(MessageSender.java:195)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:
43)
at java.lang.reflect.Method.invoke(Method.java:606)
at
org.springframework.aop.support.AopUtils.invokeJoinpointUsingReflection(AopUtils.j
ava:302)
at
org.springframework.aop.framework.ReflectiveMethodInvocation.invokeJoinpoint(Refle
ctiveMethodInvocation.java:190)
at
org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMet
hodInvocation.java:157)
at
org.springframework.transaction.interceptor.TransactionInterceptor$1.proceedWithIn
vocation(TransactionInterceptor.java:99)
at
org.springframework.transaction.interceptor.TransactionAspectSupport.invokeWithinT
ransaction(TransactionAspectSupport.java:281)
at
org.springframework.transaction.interceptor.TransactionInterceptor.invoke(Transact
ionInterceptor.java:96)
```

```
at
org.springframework.aop.framework.ReflectiveMethodInvocation.proceed(ReflectiveMet
hodInvocation.java:179)
at
org.springframework.aop.framework.JdkDynamicAopProxy.invoke(JdkDynamicAopProxy.jav
a:207)
at com.sun.proxy.$Proxy163.onMessage(Unknown Source)
at
org.springframework.jms.listener.AbstractMessageListenerContainer.doInvokeListener
(AbstractMessageListenerContainer.java:746)
at
org.springframework.jms.listener.AbstractMessageListenerContainer.invokeListener(A
bstractMessageListenerContainer.java:684)
at
org.springframework.jms.listener.AbstractMessageListenerContainer.doExecuteListene
r(AbstractMessageListenerContainer.java:651)
at
org.springframework.jms.listener.AbstractPollingMessageListenerContainer.doReceive
AndExecute(AbstractPollingMessageListenerContainer.java:315)
at
org.springframework.jms.listener.AbstractPollingMessageListenerContainer.receiveAn
dExecute(AbstractPollingMessageListenerContainer.java:233)
at
org.springframework.jms.listener.DefaultMessageListenerContainer$AsyncMessageListe
nerInvoker.invokeListener(DefaultMessageListenerContainer.java:1150)
at
org.springframework.jms.listener.DefaultMessageListenerContainer$AsyncMessageListe
nerInvoker.executeOngoingLoop(DefaultMessageListenerContainer.java:1142)
at
org.springframework.jms.listener.DefaultMessageListenerContainer$AsyncMessageListe
nerInvoker.run(DefaultMessageListenerContainer.java:1039)
at java.lang.Thread.run(Thread.java:745)
Caused by: javax.net.ssl.SSLHandshakeException: Received fatal alert:
handshake_failure
at sun.security.ssl.Alerts.getSSLException(Alerts.java:192)
at sun.security.ssl.Alerts.getSSLException(Alerts.java:154)
at sun.security.ssl.SSLSocketImpl.recvAlert(SSLSocketImpl.java:1979)
at sun.security.ssl.SSLSocketImpl.readRecord(SSLSocketImpl.java:1086)
at sun.security.ssl.SSLSocketImpl.performInitialHandshake(SSLSocketImpl.java:1332)
at sun.security.ssl.SSLSocketImpl.startHandshake(SSLSocketImpl.java:1359)
at sun.security.ssl.SSLSocketImpl.startHandshake(SSLSocketImpl.java:1343)
at sun.net.www.protocol.https.HttpsClient.afterConnect(HttpsClient.java:563)
at
sun.net.www.protocol.https.AbstractDelegateHttpsURLConnection.connect(AbstractDele
gateHttpsURLConnection.java:185)
at
sun.net.www.protocol.http.HttpURLConnection.getOutputStream(HttpURLConnection.java
:1092)
at
sun.net.www.protocol.https.HttpsURLConnectionImpl.getOutputStream(HttpsURLConnecti
onImpl.java:250)
at
org.apache.cxf.transport.http.URLConnectionHTTPConduit$URLConnectionWrappedOutputS
tream.setupWrappedStream(URLConnectionHTTPConduit.java:236)
at
org.apache.cxf.transport.http.HTTPConduit$WrappedOutputStream.handleHeadersTrustCa
ching(HTTPConduit.java:1302)
at
org.apache.cxf.transport.http.HTTPConduit$WrappedOutputStream.onFirstWrite(HTTPCon
duit.java:1262)
```

```
at
org.apache.cxf.transport.http.URLConnectionHTTPConduit$URLConnectionWrappedOutputS
tream.onFirstWrite(URLConnectionHTTPConduit.java:267)
at
org.apache.cxf.io.AbstractWrappedOutputStream.write(AbstractWrappedOutputStream.ja
va:47)
at
org.apache.cxf.io.AbstractThresholdOutputStream.write(AbstractThresholdOutputStrea
m.java:69)
at
org.apache.cxf.io.AbstractWrappedOutputStream.write(AbstractWrappedOutputStream.ja
va:60)
at
org.apache.cxf.io.CacheAndWriteOutputStream.write(CacheAndWriteOutputStream.java:8
9)
at
org.apache.cxf.attachment.AttachmentSerializer.writeProlog(AttachmentSerializer.ja
va:172)
at
org.apache.cxf.interceptor.AttachmentOutInterceptor.handleMessage(AttachmentOutInt
erceptor.java:72)
... 43 more
```

Solution: If you receive this error, then it's likely that you configured the client with TLSv1.1 while the server only accepts TLSv1.2.

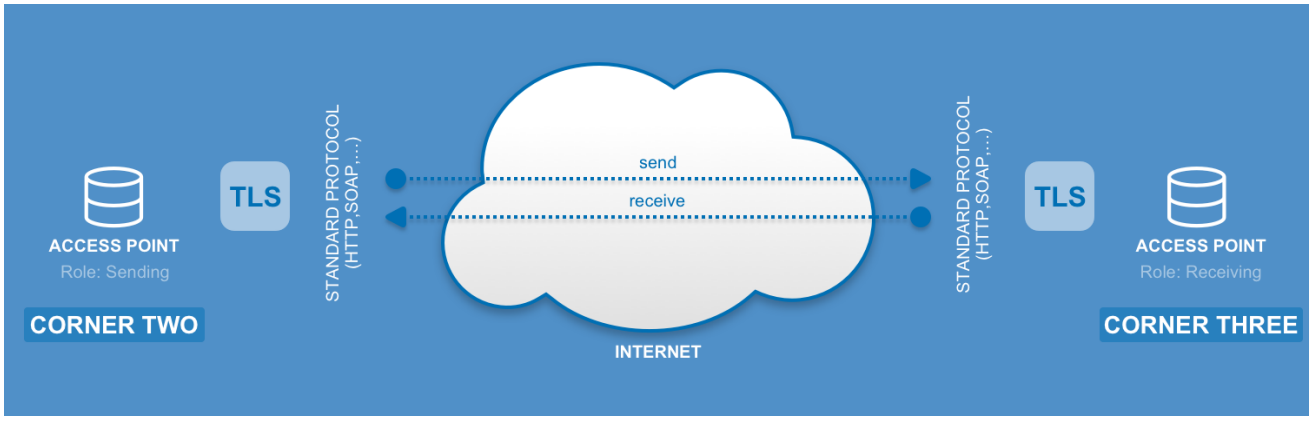
10. ANNEX 1 – TLS CONFIGURATION

10.1. TLS Configuration

10.1.1. Transport Layer Security in Domibus

One way of implementing TLS for AS4 e-Sens is to use the TLS in the Domibus Message Handler (MSH) described below, otherwise this would have to be handled at a higher level (e.g. Application Server, Proxy, etc…)

To enable secure communication at the transport layer (TLS) between a sending and a receiving MSH (Access Point), both the client and the server need to be configured accordingly.



The client is used in the initiator MSH to send the request and is therefore configured via CXF while the server is configured at container/application server level.

10.1.2. Client side configuration (One Way SSL)

The **tlsClientParameters** are configured in *cef_edelivery_path***/conf/domibus/clientauthentication.xml** file:

```
<http-conf:tlsClientParameters disableCNCheck="true"
secureSocketProtocol="TLSv1.2"
        xmlns:http-
conf="http://cxf.apache.org/transports/http/configuration"
        xmlns:security="http://cxf.apache.org/configuration/security">

<security:trustManagers>
        <security:keyStore type="JKS" password="your_trustore_password"
                           file="${domibus.config.location}/keystores/your_tr
ustore_ssl.jks"/>
</security:trustManagers>

</http-conf:tlsClientParameters>
```

Remark:

your_trustore_ssl is used at the transport layer (SSL) while your_trustore, described in §5.1.2 – "Certificates" is used by Domibus to encrypt and sign (WS-Security)

When the **clientauthentication.xml** file is present and the endpoint of the receiving MSH is *https://*, the TLS parameters are added via the CXF framework to the send request.

The version of the TLS must be specified by setting **secureSocketProtocol="TLSv1.2"**.

If you use self-signed certificates you need to set **disableCNCheck="true"**.

The attribute **disableCNCheck** specifies if JSSE should omit checking if the host name specified in the URL matches that of the Common Name (CN) on the server's certificate. Default is false; this attribute should not be set to true during production use cf.[REF7].

Remark:

TLSv1.2 is mandatory for AS4 e-Sens Profile.

10.1.3. Client side configuration (Two Way SSL)

The same as O*ne Way SSL* but the **tlsClientParameters** gets configured with both *trustManagers* and *keystoreManagers* . The **clientauthentication.xml** file should look like this:

```
<http-conf:tlsClientParameters disableCNCheck="true"
secureSocketProtocol="TLSv1.2"
        xmlns:http-conf="http://cxf.apache.org/transports/http/configuration"
        xmlns:security="http://cxf.apache.org/configuration/security">
    <security:trustManagers>
        <security:keyStore type="JKS" password="your_trustore_password"
               file="${domibus.config.location}/keystores/your_trustore_ssl.jks"/>
    </security:trustManagers>
       <security:keyManagers keyPassword="your_keystore_password">
        <security:keyStore type="JKS" password="your_keystore_password"
               file="${domibus.config.location}/keystores/your_keystore_ssl.jks"/>
    </security:keyManagers>
</http-conf:tlsClientParameters>
```

Remark:

your_trustore_ssl and your_keystore_ssl are used at the transport layer (SSL) while your_trustore and your_keystore, described in §5.1.2 - Certificates" are used by Domibus to encrypt and sign (WS-Security)

Two Way SSL is optional based on the AS4 e-Sens Profile.

10.1.4. Server side configuration

10.1.4.1. Tomcat 8

In Server.xml, add a new connector having **SSLEnabled** attribute set to true:

```
<Connector SSLEnabled="true"
           protocol="org.apache.coyote.http11.Http11NioProtocol"
           port="8443" maxThreads="200"
           scheme="https" secure="true"
           keystoreFile="${domibus.config.location}/keystores/your_keystore
_ssl.jks" keystorePass="your_keystore_password"
           clientAuth="false" sslProtocol="TLS" />
```

The keystore jks location and password must be specified, otherwise the default one will be considered.

TLS version can also be specified.

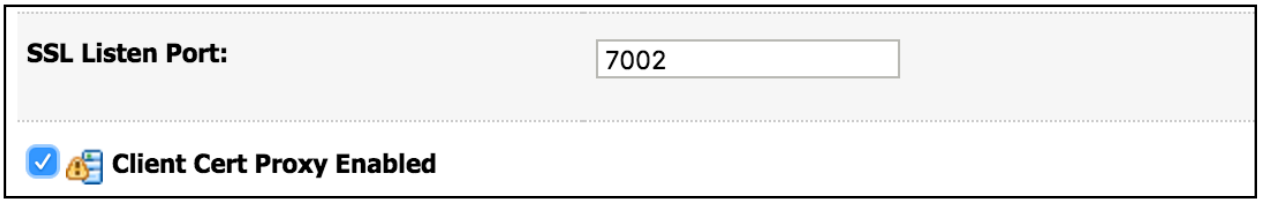
The above connector has **clientAuth="false"** this means that only the server has to authenticate himself (One Way SSL). To configure Two Way SSL, which is optional based on the *AS4 e-Sens* Profile, in Server.xml, set **clientAuth="true"** and provide the location of the *your_truststore_ssl***.jks** so that the server can verify the client:

```
<Connector SSLEnabled="true"
           protocol="org.apache.coyote.http11.Http11NioProtocol"
           port="8443" maxThreads="200"
           scheme="https" secure="true"
           keystoreFile="${domibus.config.location}/keystores/your_keystore
_ssl.jks" keystorePass="your_keystore_password"
           truststoreFile="${domibus.config.location}/keystores/your_trusts
tore_ssl.jks" truststorePass="your_trustore_password"
           clientAuth="true" sslProtocol="TLS" />
```

10.1.4.2. WebLogic

1. Specify the use of SSL on default port 7002

Go to Servers ➔ select server name ➔ Configuration ➔ General then **click** on **Client Cert Proxy Enabled**



2. Add keystore and truststore:

Go to Servers ➔ select server name ➔ Configuration ➔ Keystores and SSL tabs and use **Custom Identity and Custom Trust** then set keystore and trustore jks.

To disable basic authentication at WebLogic level:

By default WebLogic performs its own basic authentication check before passing the request to Domibus. Instead, we want basic authentication to be performed by Domibus so we disable it at application server level.

In **DOMAIN_HOME/config/config.xml** add:

```
<enforce-valid-basic-auth-credentials>false</enforce-valid-basic-auth-credentials>
```

10.1.4.3. Wildfly 9

In file *cef_edelivery_path***/domibus/standalone/configuration/standalone-full.xml***:*

- add the keystore and trustore jks to the ApplicationRealm:

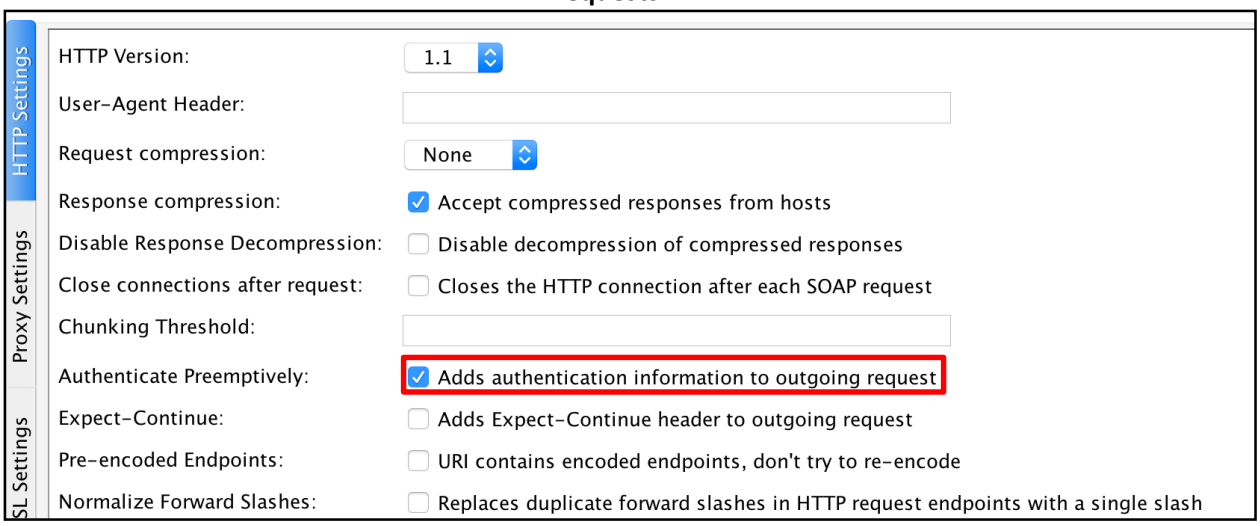
```
<security-realm name="ApplicationRealm">
  <server-identities>
    <ssl>
      <keystore path="../conf/domibus/keystores/gateway_keystore.jks"
relative-to="jboss.server.base.dir" keystore-password="test123"
alias="blue_gw" key-password="test123"/>
    </ssl>
  </server-identities>
  <authentication>
     <truststore path="../conf/domibus/keystores/gateway_truststore.jks"
relative-to="jboss.server.base.dir" keystore-password="test123" />
…
</authentication>
```

- add https-listener to default-server:

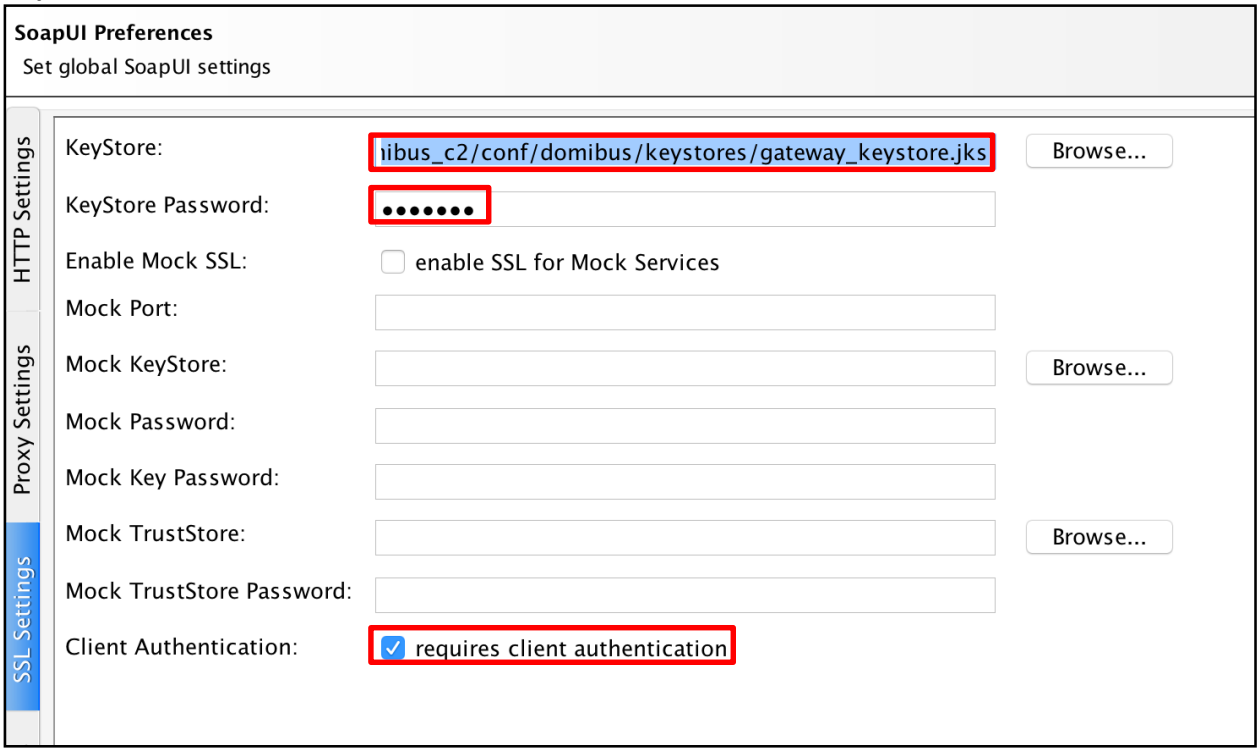
```
<subsystem xmlns="urn:jboss:domain:undertow:2.0">
            <buffer-cache name="default"/>
            <server name="default-server">
                <http-listener name="default" socket-binding="http" redirect-
socket="https"/>
                <https-listener name="default_https" socket-binding="https"
security-realm="ApplicationRealm" verify-client="REQUIRED"/>
```

10.1.4.4. Configure Basic and Certificates authentication in SoapUI

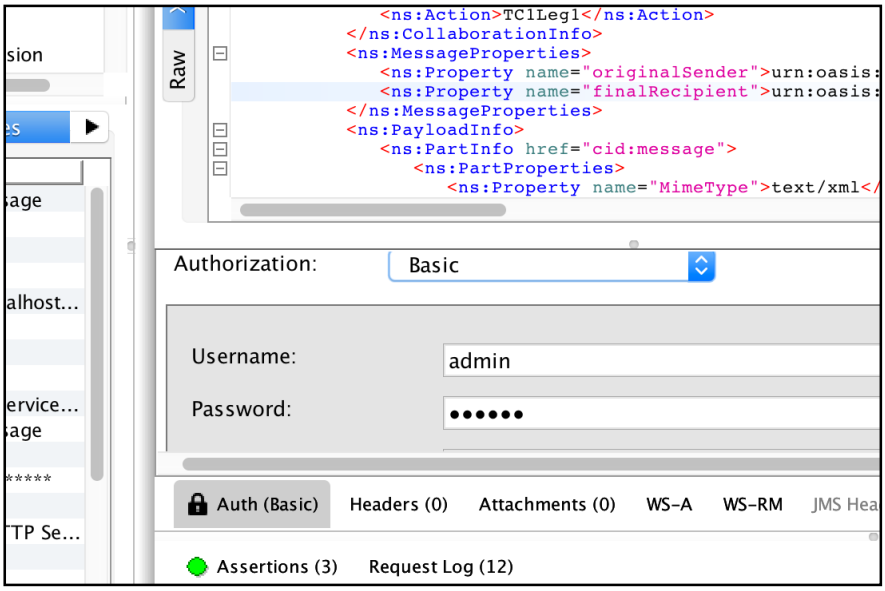
Go to File ➔ Preferences ➔ HTTP Settings and check **Adds authentication information to outgoing requests**



Go to File ➔ Preferences ➔ SSL Settings and add the **KeyStore**, **KeyStore Password** and check the **requires client authentication**



To pass Basic Authentication, in the Auth tab, click Add New Authorization and select Basic. Enter user and password (e.g: Username = ***admin***; Password = ***123456***)



10.1.4.5. PMode update

If you enable HTTPS then your PMode Configuration Manager needs to make sure that all other endpoint PModes are modified accordingly.

With the SSL connector configured as above, the MSH endpoint is now:
https://*your_domibus_host***:*8443*/domibus/services/msh**

The PMode needs to be updated accordingly and uploaded via the Admin Console:

Example:

```
<party name="party_id_name1"
endpoint=
"https://
party_id_name1_hostname:8443/domibus/services/msh" allowChunking="false">
```

11. DYNAMIC DISCOVERY OF UNKNOWN PARTICIPANTS

11.1. Overview

In a dynamic discovery setup, the sender and/or the receiver parties and their capabilities are not configured in advance.

The sending Access Point will dynamically retrieve the necessary information for setting up an interoperability process from the Service Metadata Publisher (SMP). The SMP stores the interoperability metadata which is a set of information on the end entity recipient (its identifier, supported business documents and processes in which it accepts those documents) and AP (metadata which includes technical configuration information on the receiving endpoint, such as the transport protocol and its address) cf.[REF8].

The receiving AP registers its metadata in the SMP and configures the PMode to be able to accept messages from trusted senders that are not previously configured in the PMode. The receiving AP will have to configure one process in its PMode for each SMP entry.

The mapping between the PMode process and the SMP entry is defined in §11.3 – "*Policy and* certificates *for PEPPOL*" and §11.8 - – "*Policy and* certificates *for OASIS*".

Please note that the sender does not have to register in the SMP and the receiver merely extracts its identifier from the received message.

The following sections describe how to configure Domibus AP in order to use Dynamic Discovery (§11.3 – "*PMode configuration for PEPPOL*", §11.4 – "*Policy and certificates for PEPPOL*", §11.8 – "*PMode configuration for OASIS*", §11.9 – "*Policy and certificates for OASIS*").

11.2. Domibus configuration for PEPPOL

To enable the integration with the SMP/SML components, Domibus requires some changes in the **domibus-configuration.xml** configuration file which include:

1. Add the following properties to enable the usage of the PEPPOL dynamic discovery client:

   ```
   <prop key="domibus.dynamic.discovery.client.specification">PEPPOL</prop>
   ```

2. The dynamic discovery client is using certificates to access the SMP. These certificates are different in TEST and PRODUCTION, therefore we need to specify the Mode the dynamic discovery client is using by setting the following property:

   ```
   <prop key="domibus.dynamic.discovery.peppolclient.mode">TEST</prop>
   ```

3. Set the property "**domibus.smlzone**"
4. The bean "**pModeProvider**" must be configured with "**eu.domibus.ebms3.common.dao.DynamicDiscoveryPModeProvider**"

```
<bean id="pModeProvider"
class="eu.domibus.ebms3.common.dao.DynamicDiscoveryPModeProvider"/>
```

11.3. PMode configuration for PEPPOL

11.3.1. *Sender PMode*

In a dynamic process the receiver of the messages is not known beforehand and therefore the **PMode.Responder** parameter SHOULD NOT be set.

The dynamic process must include a leg which maps the configured entry (action, service and service type – cf. 11.5 – "*Message format*") of the Receiver in the SMP.

The security policy to be used in the leg is (see §5.1.1 – "*Policies*" in for more information):

```
security="eDeliveryPolicy_CA"
```

Remark:

eSensPolicy.v2.0_CA is also supported

Sample Sender PMODE configuration extract:

```
…
<services>
   <service name="testService1"
      value="urn:www.cenbii.eu:profile:bii05:ver2.0"
      type="cenbii-procid-ubl"/>
</services>
<actions>
   <action name="tc1Action"
      value="urn:oasis:names:specification:ubl:schema:xsd:CreditNote-
2::CreditNote##urn:www.cenbii...."/>
</actions>
<securities>
   <security name="eDeliveryPolicy"
      policy="eDeliveryPolicy.xml"
      signatureMethod="RSA_SHA256"/>
   <security name="eDeliveryPolicy_CA"
      policy="eDeliveryPolicy_CA.xml"
      signatureMethod="RSA_SHA256"/>
</securities>
<legConfigurations>
   <legConfiguration name="pushTestcase1tc1Action"
   service="testService1"
   action="tc1Action"
   defaultMpc="defaultMpc"
   reliability="AS4Reliability"
   security="eDeliveryPolicy_CA"
   receptionAwareness="receptionAwareness"
   propertySet="ecodexPropertySet"
   payloadProfile="MessageProfile"
   errorHandling="demoErrorHandling"
   compressPayloads="true"/>
</legConfigurations>
<process name="tc1Process"
   agreement="agreementEmpty"
   mep="oneway"
   inding="push"
   initiatorRole="defaultInitiatorRole"
   responderRole="defaultResponderRole">
```

```
   <initiatorParties>
      <initiatorParty name="senderalias"/>
   </initiatorParties>
   <!-- no responderParties element -->
   <legs>
      <leg name="pushTestcase1tc1Action"/>
   </legs>
</process>
…
```

11.3.2. Receiver PMode

Dynamic configuration of the receiver is similar to the configuration of the sender, except that the roles are swapped: the sender of the messages is not known beforehand; consequently the ***PMode.Initiator*** parameter SHOULD NOT be set.

```
…
<process name="tc1Process"
   agreement="agreementEmpty"
   mep="oneway"
   inding="push"
   initiatorRole="defaultInitiatorRole"
   responderRole="defaultResponderRole">
   <responderParties>
      <responderParty name="receiveralias"/>
   </responderParties>
   <!-- no initiatorParties element -->
   <legs>
      <leg name="pushTestcase1tc1Action"/>
   </legs>
</process>
…
```

11.4. Policy and certificates for PEPPOL

The receiver must include the certificate of the trusted authority(ies) in its trustore. Its will accept only messages that were signed with certificates issued by these trusted authority(ies). (cf. §12 - Annex 1 - Usage of certificates in PEPPOL and OASIS for more information).

11.5. Message format for PEPPOL

When dynamic discovery is used, the "to" field should not be statically configured in the PMode (the "to" field may even be omitted in the message). The lookup is performed by C2 based on the **finalRecipient** message property.

Example of message using the **finalRecipient** for dynamic discovery:

```
<ns:UserMessage>
  <ns:PartyInfo>
    <ns:From>
      <ns:PartyId type="urn:oasis:names:tc:ebcore:partyid-
type:unregistered">senderalias</ns:PartyId>
      <ns:Role>http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/ns/core/200704/initiator</ns:Role>
    </ns:From>
    <ns:To>
    </ns:To>
  </ns:PartyInfo>
  <ns:CollaborationInfo>
    <ns:Service type="cenbii-procid-
ubl">urn:www.cenbii.eu:profile:bii05:ver2.0</ns:Service>
            <ns:Action>urn:oasis:names:specification:ubl:schema:xsd:CreditNote-
2::CreditNote##urn:www.cenbii.eu:transaction:biitrns014:ver2.0:extended:urn:www.pe
ppol.eu:bis:peppol5a:ver2.0::2.1</ns:Action>
  </ns:CollaborationInfo>
    <ns:MessageProperties>
      <ns:Property name="originalSender">urn:oasis:names:tc:ebcore:partyid-
type:unregistered:C1</ns:Property>
      <ns:Property name="finalRecipient" type="iso6523-actorid-
upis">0007:9340033829test1</ns:Property>
    </ns:MessageProperties>
</ns:UserMessage>
```

11.6. SMP entry

The following table describes the mapping between the PMode static configuration and the dynamic SMP records structure:

SMP Endpoint registration record	PMode attributes
ServiceMetadata/ServiceInformation/DocumentIdentifier	PMode[1].BusinessInfo.Service
ServiceInformation/Processlist/Process/ProcessIdentifier/@scheme	PMode[1].BusinessInfo.Service/@Type
ServiceMetadata/ServiceInformation/DocumentIdentifier	Pmode[1].BusinessInfo.Action
ServiceInformation/Processlist/Process/ServiceEndpointList/Endpoint/EndpointReference/Address	Pmode[].Protocol.Address

Table 4 - SMP Entry Mapping

The Service Metadata Record also provides the receiving end's certificate. This certificate can be used to encrypt the message to be sent to the receiver. The certificate can also provide the name of the gateway for this PMode by using the Certificate's CNAME as the PMode identifier cf.[REF10].

11.7. Domibus configuration for OASIS

To enable the integration with the SMP/SML components, Domibus requires some changes in the **domibus-configuration.xml** configuration file:

1. Add the following properties to enable the usage of the OASIS dynamic discovery client:

   `<prop key="domibus.dynamic.discovery.client.specification"> OASIS </prop>`

   Note: this property is not mandatory as it defaults to the above value.

2. Set the property "**domibus.smlzone**", e.g. "ehealth.acc.edelivery.tech.ec.europa.eu"
3. The bean "**pModeProvider**" must be configured with "**eu.domibus.ebms3.common.dao.DynamicDiscoveryPModeProvider**"

```
<bean id="pModeProvider"
class="eu.domibus.ebms3.common.dao.DynamicDiscoveryPModeProvider"/>
```

11.8. PMode configuration for OASIS

11.8.1. *Sender PMode*

In a dynamic process the receiver of the messages is not known beforehand and therefore the **PMode.Responder** parameter SHOULD NOT be set.

The dynamic process must include a leg which maps the configured entry (action, service and service type – cf. 11.5 – "*Message format*") of the Receiver in the SMP.

The security policy to be used in the leg is (see §5.1.1 – "*Policies*" in for more information):

```
security="eDeliveryPolicy_CA"
```

Remark:

eSensPolicy.v2.0_CA is also supported

Sample Sender PMODE configuration extract:

```
…
<services>
   <service name="testService1"
      value="urn:www.cenbii.eu:profile:bii05:ver2.0"
      type="cenbii-procid-ubl"/>
</services>
<actions>
   <action name="tc1Action"
      value="'your-schema-
name'::urn:oasis:names:specification:ubl:schema:xsd:CreditNote-
2::CreditNote##urn:www.cenbii...."/>
</actions>
<securities>
   <security name="eDeliveryPolicy"
      policy="eDeliveryPolicy.xml"
      signatureMethod="RSA_SHA256"/>
   <security name="eDeliveryPolicy_CA"
      policy="eDeliveryPolicy_CA.xml"
      signatureMethod="RSA_SHA256"/>
</securities>
<legConfigurations>
   <legConfiguration name="pushTestcase1tc1Action"
   service="testService1"
   action="tc1Action"
   defaultMpc="defaultMpc"
   reliability="AS4Reliability"
   security="eDeliveryPolicy_CA"
   receptionAwareness="receptionAwareness"
   propertySet="ecodexPropertySet"
   payloadProfile="MessageProfile"
   errorHandling="demoErrorHandling"
   compressPayloads="true"/>
</legConfigurations>
<process name="tc1Process"
   agreement="agreementEmpty"
   mep="oneway"
   inding="push"
   initiatorRole="defaultInitiatorRole"
```

```
   responderRole="defaultResponderRole">
   <initiatorParties>
      <initiatorParty name="senderalias"/>
   </initiatorParties>
   <!-- no responderParties element -->
   <legs>
      <leg name="pushTestcase1tc1Action"/>
   </legs>
</process>
…
```

Remark:

> *Schema name should be added to action value. e.g:* ***ehealth-actorid-qns::urn:oasis:names:specification:ubl:schema:xsd:CreditNote-2::CreditNote##urn:www.cenbii...***

11.8.2. Receiver PMode

Dynamic configuration of the receiver is similar to the configuration of the sender, except that the roles are swapped: the sender of the messages is not known beforehand; consequently the **PMode.Initiator** parameter SHOULD NOT be set.

```
…
<process name="tc1Process"
   agreement="agreementEmpty"
   mep="oneway"
   inding="push"
   initiatorRole="defaultInitiatorRole"
   responderRole="defaultResponderRole">
   <responderParties>
      <responderParty name="receiveralias"/>
   </responderParties>
   <!-- no initiatorParties element -->
   <legs>
      <leg name="pushTestcase1tc1Action"/>
   </legs>
</process>
…
```

11.9. Policy and certificates for OASIS

The receiver must include the certificate of the trusted authority(ies) in its trustore. Its will only accept messages that were signed with certificates issued by these trusted authority(ies).

The sender truststore must embed the SMP public certificate. This certificate is used by the AP to validate the identity of the used SMP (cf. §12 - Annex 1 - Usage of certificates in PEPPOL and OASIS for more information).

11.10. Message format for OASIS

When dynamic discovery is used, the "to" field should not be statically configured in the PMode (the "to" field may even be omitted in the message). The lookup is performed by C2 based on the **finalRecipient** message property.

Note: For the OASIS client, in the PMode "action" value, the document scheme must be included with the document ID (for PEPPOL client, only document ID is needed).

Example of message using the **finalRecipient** for dynamic discovery:

```
<ns:UserMessage>
  <ns:PartyInfo>
    <ns:From>
      <ns:PartyId type="urn:oasis:names:tc:ebcore:partyid-
type:unregistered">senderalias</ns:PartyId>
      <ns:Role>http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/ns/core/200704/initiator</ns:Role>
    </ns:From>
    <ns:To>
    </ns:To>
  </ns:PartyInfo>
  <ns:CollaborationInfo>
    <ns:Service type="cenbii-procid-
ubl">urn:www.cenbii.eu:profile:bii05:ver2.0</ns:Service>

      <ns:Action>'your_schema_name'::urn:oasis:names:specification:ubl:schema:xsd
:CreditNote-
2::CreditNote##urn:www.cenbii.eu:transaction:biitrns014:ver2.0:extended:urn:www.pe
ppol.eu:bis:peppol5a:ver2.0::2.1</ns:Action>
  </ns:CollaborationInfo>
    <ns:MessageProperties>
      <ns:Property name="originalSender">urn:oasis:names:tc:ebcore:partyid-
type:unregistered:C1</ns:Property>
      <ns:Property name="finalRecipient" type="iso6523-actorid-
upis">0007:9340033829test1</ns:Property>
    </ns:MessageProperties>
</ns:UserMessage>
```

12. ANNEX 1 - USAGE OF CERTIFICATES IN PEPPOL AND OASIS

		C2		C3	
		Keystore	**Truststore**	**Keystore**	**Truststore**
PEPPOL	Certificate:	Sender's (issued by CA)	Empty	Receiver's	CA's
	Note:	C2 signs the message with its private key	C2 discover C3's public certificate from the SMP	C3 signs the receipt with its private key	The receiver trusts all senders who's certificate were issue dby these CA's
OASIS	Certificate:	Sender's (issued by CA)	SMP's	Receiver's	CA's
	Note:	C2 signs the message with its private key	C2 discover C3's public certificate from the SMP To trust the SMP, the sender needs its public certificate	C3 signs the receipt with its private key	The receiver trusts all senders who's certificate were issue dby these CA's

13. ANNEX 2 – DOCUMENT PARTS

14. List of Figures

Figure 1 - Diagram representing the Deployment of Domibus in a Cluster on WebLogic 23
Figure 2 - Diagram representing the Deployment of Domibus in a Cluster on Tomcat 40
Figure 3 - Diagram representing the Deployment of Domibus in a Cluster on WildFly 52
Figure 4 - Message Service Handler diagram 55
Figure 5 - PMode view 63
Figure 6 - Login to administration dashboard 74
Figure 7 - Configuration upload 74
Figure 8 - PMode uploading 75
Figure 9 - State machine of Corner 2 (sending access point) 77
Figure 10 - State machine of Corner 3 (receiving access point) 77
Figure 11 - Domibus Message Log 78
Figure 12 - log4j.poperties 80
Figure 13 - Domibus – Error Log page 80
Figure 14 - Domibus – JMS Monitoring page 82
Figure 15 - Domibus – Message filter page 88

15. List of Tables

Table 1 - Domibus Properties 59
Table 2 - Domibus PMode configuration to ebMS3 mapping 73
Table 3 - Queue Monitoring 82
Table 4 - SMP Entry Mapping 109

16. CONTACT INFORMATION

CEF Support Team

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

By phone: +32 2 299 09 09

- Standard Service: 8am to 6pm (Normal EC working Days)
- Standby Service*: 6pm to 8am (Commission and Public Holidays, Weekends)

* *Only for critical and urgent incidents and only by phone*