



EUROPEAN COMMISSION

DIRECTORATE-GENERAL INFORMATICS

# **Domibus 3.1 RC2**

## **Quick Start Guide**

Author(s) : CEF Support  
Reviewed by :

Approved by :  
Version : 1.00

Date: 18/04/2016

# CONTENTS

<b>INTRODUCTION.....</b>	<b>3</b>
<b>PURPOSE OF THIS GUIDE .....</b>	<b>4</b>
<b>PREREQUISITES .....</b>	<b>6</b>
<b>CONFIGURE YOUR ENVIRONMENT .....</b>	<b>7</b>
1.1. Package Overview .....	7
1.1.1. domibus-distribution-3.1-RC2-tomcat-full.zip .....	7
1.1.2. domibus-distribution-3.1-RC2-sample-configuration-and-testing.zip .....	10
1.2. Tomcat Standalone Access Point .....	10
<b>TESTING .....</b>	<b>17</b>
<b>DEFAULT PLUGINS.....</b>	<b>20</b>
<b>ANNEX 1 PARAMETERS.....</b>	<b>21</b>
<b>ANNEX 2 FIREWALL SETTINGS.....</b>	<b>22</b>
<b>ANNEX 3 PROCESSING MODE .....</b>	<b>25</b>
<b>ANNEX 4 DOMIBUS PCONF TO EBMS3 PMODE MAPPING .....</b>	<b>29</b>
<b>ANNEX 5 INTRODUCTION TO AS4 SECURITY .....</b>	<b>35</b>

## INTRODUCTION

CEF e-Delivery provides a set of components to exchange messages over the internet using B2B protocols. See the document concerning the [Introduction to the Connecting Europe Facility eDelivery building block](#) available on the CEF Single Web Portal for more information.

In this particular *static* deployment context, the full set of components (e.g. dynamic discovery, connector) is not required. Participants cannot communicate directly with one another. Participants must always communicate using B2B, AS4 protocol.

Therefore, this specific release contains two archives:

- **domibus-distribution-3.1-RC2-tomcat-full.zip** containing the full Tomcat distribution. Default Web Service plugin is also included in this archive and deployed as the default plugin.
- **domibus-distribution-3.1-RC2-sample-configuration-and-testing.zip** containing a sample of certificates, pMode configuration files and test SoapUI project.

Two additional archives containing default JMS plugin and default Web Service plugin are also available:

- **domibus-distribution-3.1-RC2-default-jms-plugin.zip** containing the binaries and configuration file for the JMS plugin
- **domibus-distribution-3.1-RC2-default-ws-plugin.zip** containing the binaries and configuration file for the Web Service plugin

The release provides an AS4 Access Point (CEF eDelivery component called Domibus) running on a Tomcat application server and using a MySQL database for data persistence.

## PURPOSE OF THIS GUIDE

This release contains the AS4 Access Point of the CEF eDelivery Digital Service Infrastructure (DSI). It is important to note that this release of the AS4 Access Point contains 'Domibus 3.1 Release Candidate 2'. For more information about this release, please refer to the accompanying release note.

The source code of the 'Domibus' is a release candidate code and the Web Services Description Language (WSDL) file contained in the 'Domibus' is still being tested and is subject to possible minor change in the future. Overall, therefore, this release should not be used for production purposes as performance/ load/ scalability testing is still ongoing. Final release is planned in May 2016, which will complete the current work on the AS4 Access Point.

This release of the CEF eDelivery Access Point is the result of significant collaboration among different EU policy projects, IT delivery teams and the CEF eDelivery DSI. Nevertheless, this eDelivery release is fully reusable by any other policy domain of the EU.

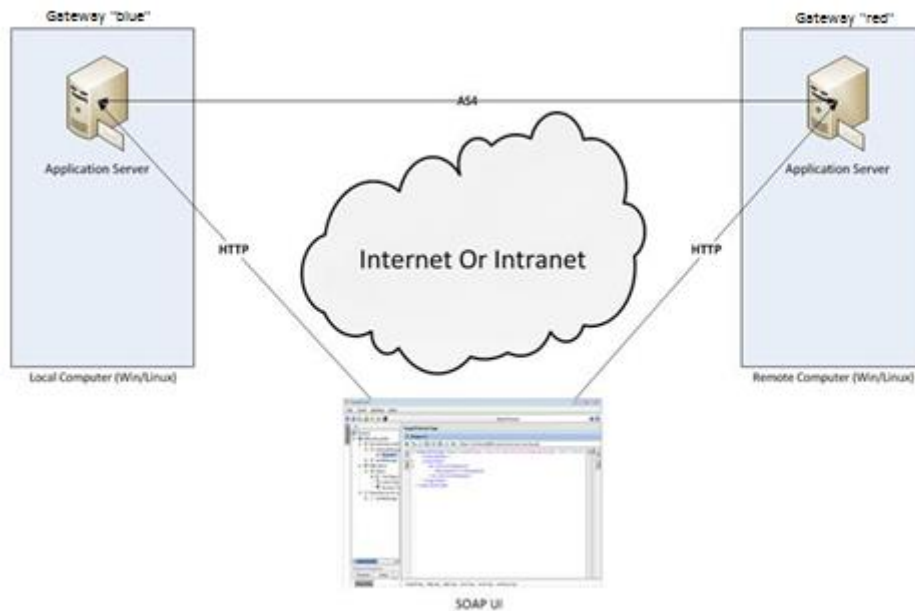
This release Candidate supports Tomcat 8 and WebLogic 12c, and is compatible with Oracle 10g+ and MySQL 5.5+. In this guide, we are covering Tomcat/MySQL configuration.

*Remark:*

*PostgreSQL is not officially supported.*

In other words, we will guide you to setup two Tomcat standalone Access Points, deployed on different machines, to exchange B2B documents securely over AS4 by:

- Deploying and configuring both Access Points (blue and red)
- Configuring processing mode files for both AS4 Access Points
- Using the provided AS4 Access Points certificates
- Setup the Access Points blue and red for running test cases (see [Testing section](#))



Installation on two different machines

*Remarks:*

- *The same procedure can be extended to a third (or more) Access Point.*
- *This guide does not cover the preliminary network configuration allowing communication between separate networks (i.e. infrastructure firewall/Proxy setup).*

## PREREQUISITES

- Java runtime environment (JRE), version 7:  
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- JCE Unlimited Strength Policy files, for JRE7:  
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>  
Copy the jar files from the extracted zip to <JRE\_HOME>\lib\security.
- MySQL database server listening on the default port 3306:  
<http://dev.mysql.com/downloads/>

Please install the above software on your host machine. For further information and installation details, we kindly advise you to refer to the manufacturers' websites.

*Remark:*

*Please ensure that environment variable `JAVA_HOME` is set to JRE7 but also that the path for JRE7 and MySQL are set to their respective bin directory.*

## CONFIGURE YOUR ENVIRONMENT







### 1.1. Package Overview

#### 1.1.1. domibus-distribution-3.1-RC2-tomcat-full.zip

Download the Domibus3.1-RC2 Distribution from the Single Web Portal:

<https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/AP+-+v3.1+RC2>

This package has the following structure and contains a naked version of Domibus:

Name	Type	Compressed size
 domibus	File folder	
 sql-scripts	File folder	
 changelog.txt	Text Document	1 KB
 Installation_Instructions.pdf	Adobe Acrobat Document	932 KB
 Quick_Start_Guide.pdf	Adobe Acrobat Document	78 KB
 upgrade-info.txt	Text Document	1 KB

#### Package content

- **<CEF-eDelivery path>/domibus/bin** contains the executable batch file (Windows) and shell script (Linux) which are required to launch the Access Point.

*Remark:*

*<CEF-eDelivery path> is the location where you extracted the downloaded package.*

- <CEF-eDelivery path>/domibus contains:

Name	Type	Compressed size
bin	File folder	
common	File folder	
conf	File folder	
lib	File folder	
logs	File folder	
shared	File folder	
temp	File folder	
webapps	File folder	
.cargo	CARGO File	1 KB
LICENSE	File	14 KB
NOTICE	File	1 KB
RELEASE-NOTES	File	3 KB
RUNNING.txt	Text Document	6 KB

- **conf** folder where you will find the *configuration files* (.xml used to administer your Tomcat and the default domibus configuration files)
- **logs** folder where the logs are stored
- **webapps** folder where the WAR files are stored

Name	Type	Compressed size
host-manager	File folder	
manager	File folder	
domibus.war	WAR File	51,144 KB

- <CEF-eDelivery path>/domibus/conf/domibus contains domibus configuration files.

Name	Type	Compressed size
internal	File folder	
plugins	File folder	
policies	File folder	
domibus-configuration.xml	XML File	2 KB
domibus-datasources.xml	XML File	2 KB
domibus-plugins.xml	XML File	1 KB
domibus-security.xml	XML File	2 KB
domibus-transactions.xml	XML File	2 KB
log4j.properties	notepad++	1 KB
persistence.xml	XML File	1 KB



- **<CEF-eDelivery path>/sql-scripts** contains the required application SQL code that needs to be executed on the MySQL database (and scripts for Oracle DB).

### 1.1.2. [domibus-distribution-3.1-RC2-sample-configuration-and-testing.zip](#)

Download the Domibus3.1-RC2 configuration files sample from the CEF Digital Single Web Portal:

<https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/AP++v3.1+RC2>

This package has the following structure and contains pre-configured files for Domibus:

Name	Type
 conf	File folder
 test	File folder

- **<CEF-eDelivery path>/test** contains a SOAP UI test project.
- **<CEF-eDelivery path>/domibus/conf /pmodes** contains two AS4 processing mode (xml file, one for blue and one for red Access Point) pre-configured to use compression, payload encryption, message signing and non-repudiation, according to the [eSENS AS4 profile](#).
- **<CEF-eDelivery path>/domibus/conf/domibus/keystores** contains a keystore (with the private keys of Access Point blue and Access Point red) and a truststore (with the public keys of Access Point *blue* and Access Point *red*) that can be used by both Access Points. Note that the keystore contains the private keys of both Access Points blue and red. This setup is not secured and is only proposed for convenience purpose. In production, the private key is only known by one participant and only deployed in his keystore. For this test release, each Access Point uses self-signed certificates. Please refer to [Annex 5](#) for more information about AS4 security.

*Remark:*

*The /conf folder in the sample archive should be unzipped in "<CEF-eDelivery path>/domibus".*

## 1.2. Tomcat Standalone Access Point

As described in the purpose of this guide, we need to configure two Access Points running on two separate machines. Therefore, the procedure below would need to be applied on both machines *Hostname "blue"* (**<blue\_hostname>:8080**) and *Hostname "red"* (**<red\_hostname>:8080**).

1. Unzip the archives:
  - a. Unzip **domibus-distribution-3.1-RC2-tomcat-full.zip** to a location on your physical machine, which we will refer to in this document as your **< CEF-eDelivery path >**.
  - b. The **/conf** folder in the **domibus-distribution-3.1-RC2-sample-configuration-and-testing.zip** should be unzipped in "**<CEF-eDelivery path>/domibus**".
2. Prepare the mysql database:
  - a. Open a command prompt and navigate to this directory:  
**< CEF-eDelivery path >/sql-scripts**.
  - b. Execute the following commands in the command prompt :

```
mysql -h localhost -u root --password=root -e "drop schema if exists
domibus;create schema domibus;alter database domibus charset=utf8; create user
edelivery identified by 'edelivery';grant all on domibus.* to edelivery;"
```

```
mysql -h localhost -u root --password=root domibus < mysql5innoDB-initial.ddl
mysql -h localhost -u root --password=root domibus < mysql5innoDB-quartz.ddl
```

*Remarks:*

*If you are using Windows, make sure to have mysql.exe added to your PATH variable.*

*If you are using a different schema, please adapt your commands but also edit the **conf/domibus/domibus-datasources.xml** file*

```
<prop key="user">edelivery</prop>
<prop key="password">edelivery</prop>
<prop key="url">
jdbc:mysql://localhost:3306/domibus?pinGlobalTxToPhysicalConnection=true
</prop>
```

- c. Add MySQL JDBC driver (.jar file available on MySQL official web site) in the folder /domibus/lib.

d. 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):

MSSQLServerADHelper100		SQL Active...	Stopped	N/A
MySQL56	2708	MySQL56	Running	N/A
napagent		Network A...	Stopped	NetworkSe...

MySQL service

### 3. Set \$domibus.config.location

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 **/domibus/bin/setenv.bat** (Windows) and **/domibus/bin/setenv.sh** (Linux), uncomment lines two and three and adapt CATALINA\_HOME value to the absolute path of the installation **<CEF-eDelivery path>/domibus**

#### Windows

```
REM Please change CATALINA_HOME to the right folder (below line only works if you start from current folder)
REM set CATALINA_HOME=<YOUR_INSTALLATION_PATH>
REM set JAVA_OPTS=%JAVA_OPTS% -Ddomibus.config.location=%CATALINA_HOME%\conf\domibus -Xmx768m -XX:MaxPermSize=256m
```

#### Linux

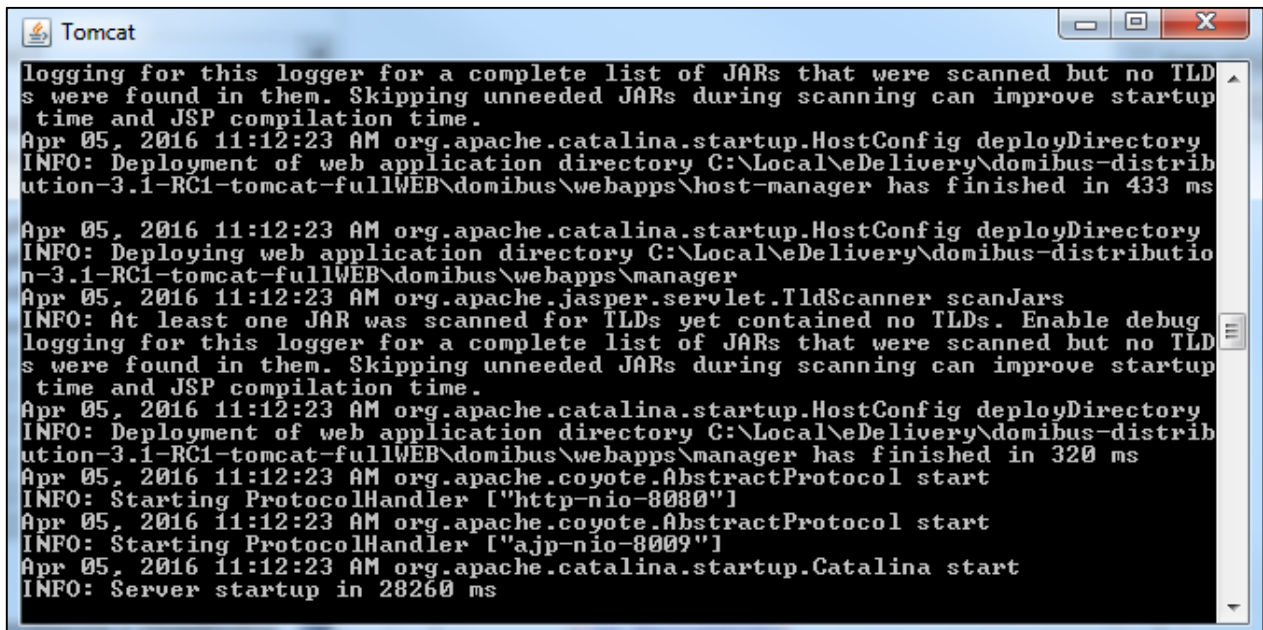
```
#Please change CATALINA_HOME to the right folder
#export CATALINA_HOME=<YOUR_INSTALLATION_PATH>
#JAVA_OPTS="$JAVA_OPTS -XX:MaxPermSize=4096m -Ddomibus.config.location=$CATALINA_HOME/conf/domibus"
```

### 4. You can now start the Tomcat standalone Access Point on your computer.

Execute:

- <CEF-eDelivery path>/domibus/bin/startup.sh** (for Linux)
- <CEF-eDelivery path>/domibus/bin/startup.bat** (for Windows)

Expected result:

A screenshot of a Windows command window titled "Tomcat". The window displays a series of log messages from the Apache Catalina startup process. The logs indicate the deployment of web application directories, the scanning of JARs for TLDs, and the starting of the Coyote AbstractProtocol and the Catalina server. The final message states "Server startup in 28260 ms".

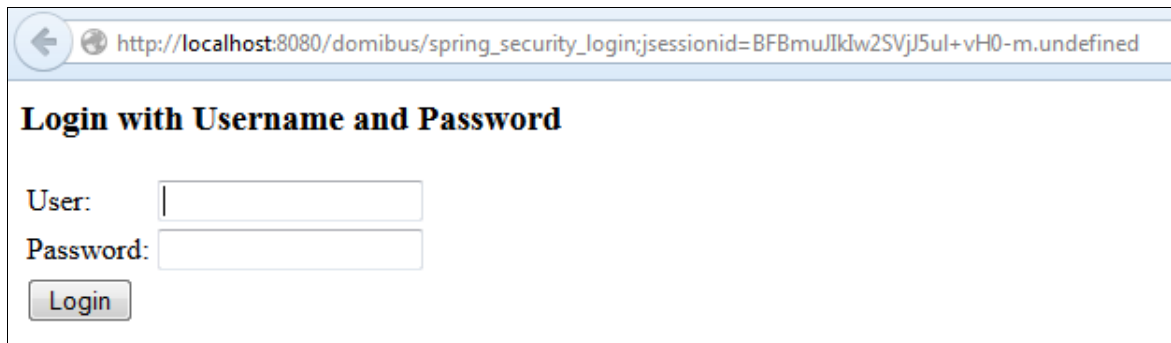
```
logging for this logger for a complete list of JARs that were scanned but no TLD
s were found in them. Skipping unneeded JARs during scanning can improve startup
time and JSP compilation time.
Apr 05, 2016 11:12:23 AM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deployment of web application directory C:\Local\edelivery\domibus-distrib
ution-3.1-RC1-tomcat-fullWEB\domibus\webapps\manager has finished in 433 ms
Apr 05, 2016 11:12:23 AM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory C:\Local\edelivery\domibus-distributio
n-3.1-RC1-tomcat-fullWEB\domibus\webapps\manager
Apr 05, 2016 11:12:23 AM org.apache.jasper.servlet.TldScanner scanJars
INFO: At least one JAR was scanned for TLDs yet contained no TLDs. Enable debug
logging for this logger for a complete list of JARs that were scanned but no TLD
s were found in them. Skipping unneeded JARs during scanning can improve startup
time and JSP compilation time.
Apr 05, 2016 11:12:23 AM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deployment of web application directory C:\Local\edelivery\domibus-distrib
ution-3.1-RC1-tomcat-fullWEB\domibus\webapps\manager has finished in 320 ms
Apr 05, 2016 11:12:23 AM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-nio-8080"]
Apr 05, 2016 11:12:23 AM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-nio-8009"]
Apr 05, 2016 11:12:23 AM org.apache.catalina.startup.Catalina start
INFO: Server startup in 28260 ms
```

#### Tomcat Access Point up and running

*Remark:*

*If the application server does not start properly, more details about the encountered errors can be found in the log files. Refer to **<CEF-eDelivery path>/domibus/logs/***

5. Once the application server is started, you can ensure that this server is operational by displaying the administration dashboard ([http://localhost:8080/domibus/spring\\_security\\_login](http://localhost:8080/domibus/spring_security_login)) in your browser as below:



The screenshot shows a web browser window with the address bar displaying `http://localhost:8080/domibus/spring_security_login;jsessionid=BFBmuJlkdw2SVjJ5ul+vh0-m.undefi`. The page content includes the heading **Login with Username and Password**, followed by two input fields labeled 'User:' and 'Password:', and a 'Login' button.

[Domibus administration page](#)

*Remarks:*

- To allow the remote application to send a message to this machine, you would need to create a dedicated rule (to allow this port) from your local firewall ( cf. annex "[Firewall Settings](#)")
- If you intend to install both Access Points on the same server, you will need to change Access Point red ports to avoid conflicts and database schema 'domibus' before starting the server.

## 6. Upload pModes

Edit the two pmode files **<CEF-eDelivery path>/domibus/conf/domibus/pmodes/domibus-gw-sample-pmode-blue.xml** and **domibus-gw-sample-pmode-red.xml** and replace **<blue\_hostname>** and **<red\_hostname>** with their real hostnames or IPs:

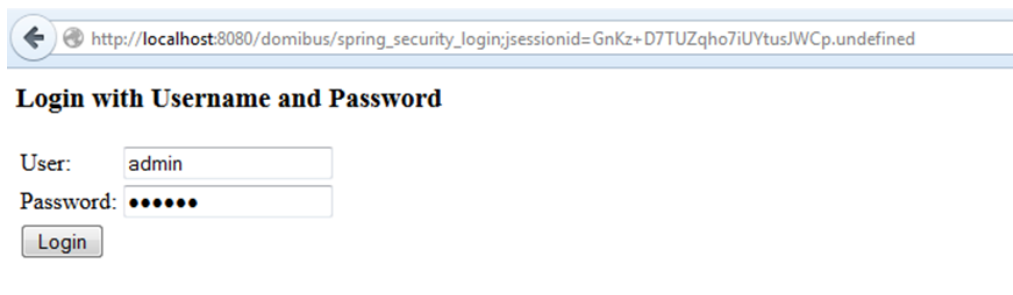
```
<party name="red_gw"
  endpoint="http://<red_hostname>:8080/domibus/services/msh"
  allowChunking="false">
  <identifier partyId="urn:oasis:names:tc:ebcore:partyid-type:unregistered:domibus-red" partyIdType="partyTypeEmpty"/>
</party>
<party name="blue_gw"
  endpoint="http://<blue_hostname>:8080/domibus/services/msh"
  allowChunking="false">
  <identifier partyId="urn:oasis:names:tc:ebcore:partyid-type:unregistered:domibus-blue" partyIdType="partyTypeEmpty"/>
</party>
```

### PMode view

For more details about the provided PMode, please [see Annex 4](#).

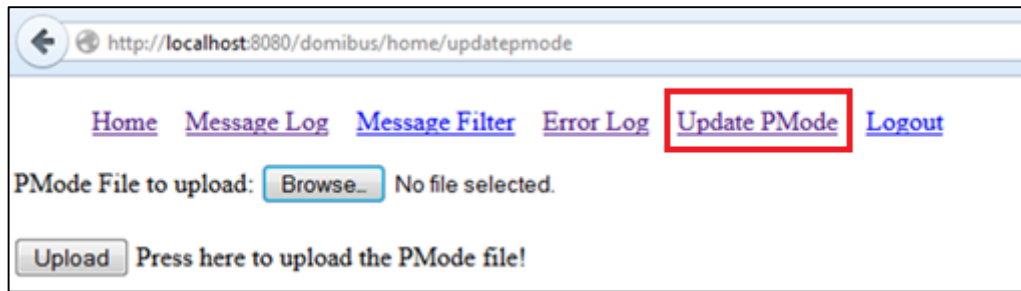
Upload the PMode file on both Access Points:

- To upload a PMode XML file, connect to the administration dashboard using your credentials (by default: login = **admin**; password = **123456**) to <http://localhost:8080/domibus/home>



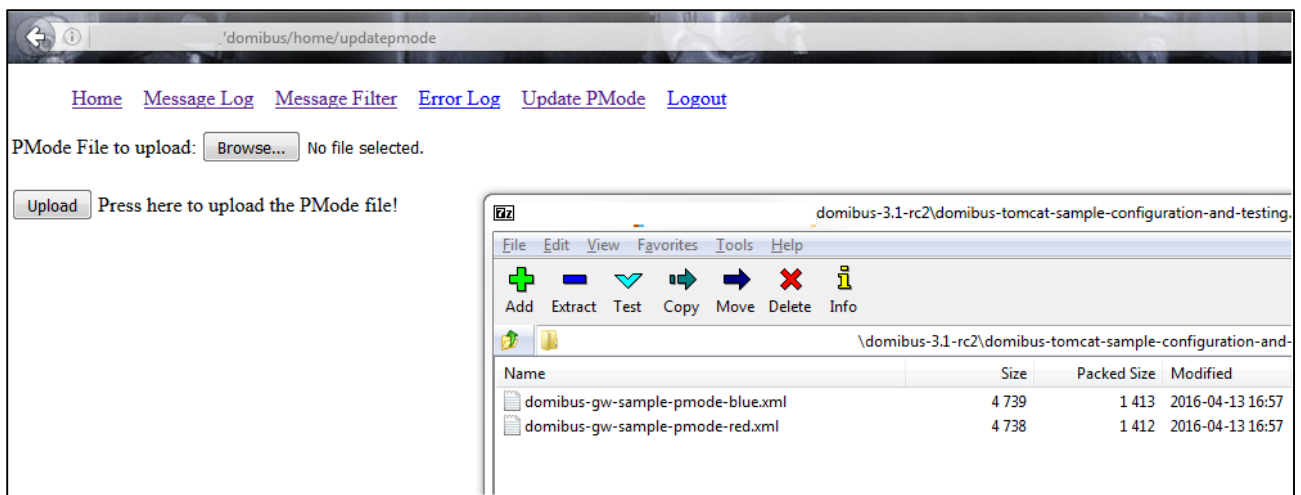
### Login to administration dashboard

- Click on the *Upload PMode* tab:



PMode update

- c. Select your PMode from "< CEF-eDelivery path >domibus/conf/domibus/pmodes/" and click on **Upload**:



PMode uploading

Now your Tomcat Access Points are running and ready to send or receive messages.



## TESTING

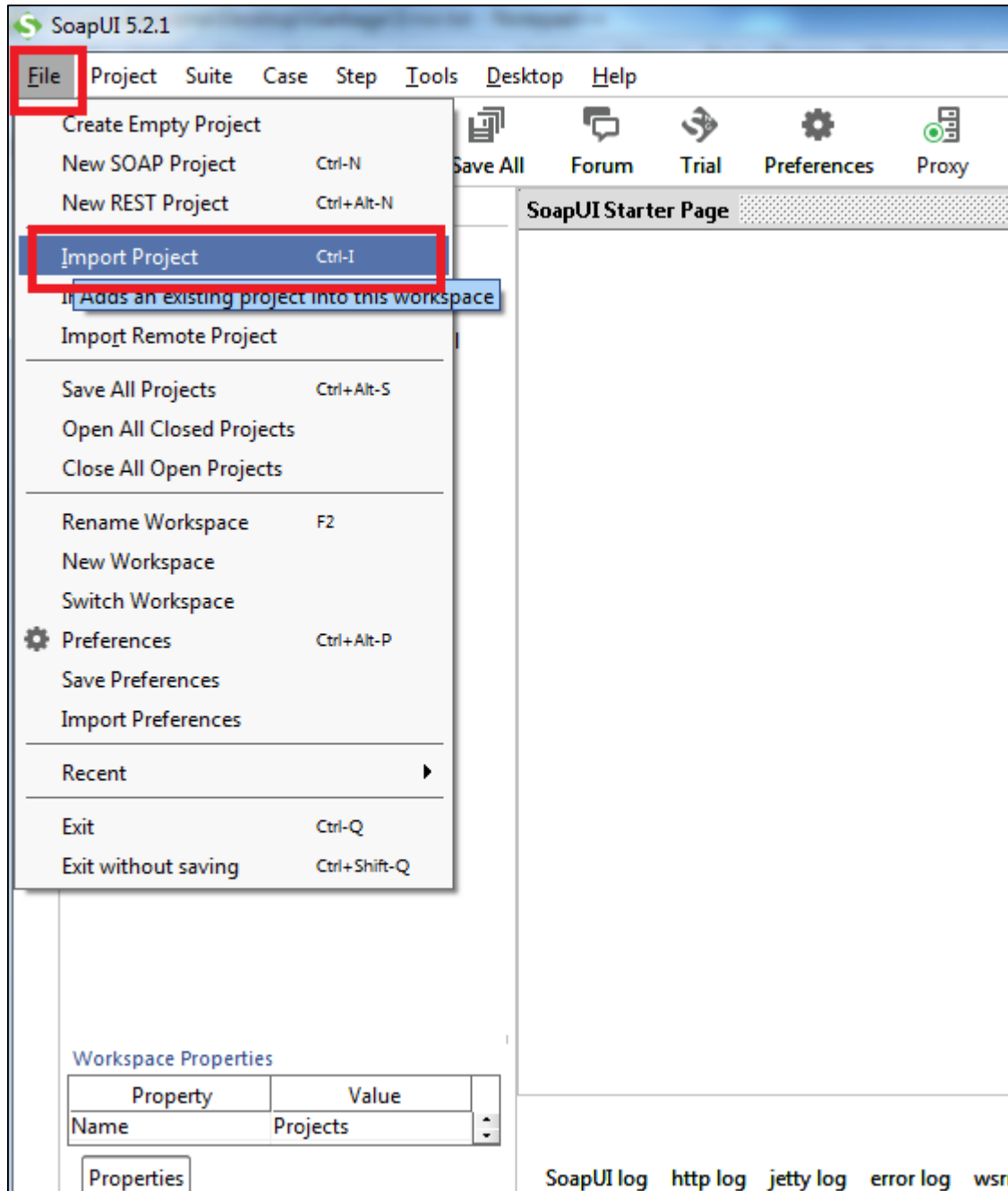
As explained in the Release Notes document and to facilitate testing, we have developed a Reference Web Service endpoint to illustrate how participants can connect and interact with the AS4 Access Point to send messages.

In addition, it is possible for the backends to download received messages from their Access Point using a request (downloadMessage) defined in the same WSDL.

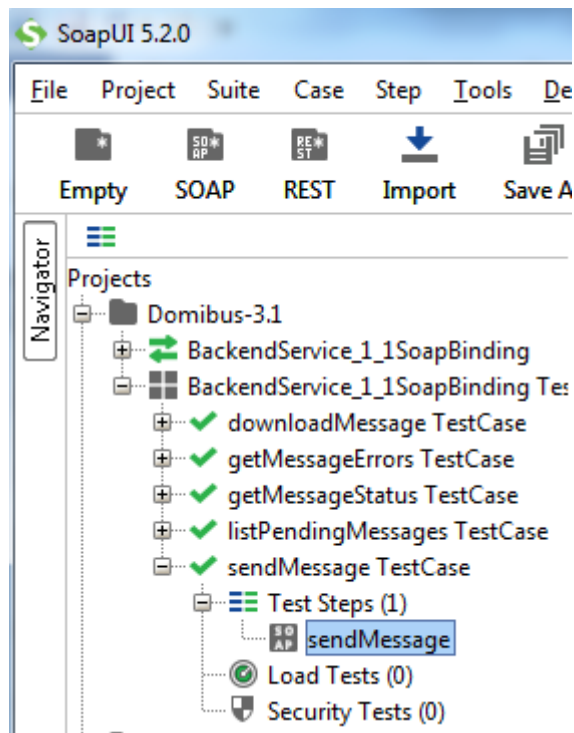
Please refer to the *Known Limitations* section in the *Release Notes* for any restrictions in the default configuration.

## SOAPUI Tutorial:

1. Once it is installed open it and click on file and import project. Browse to **/test/soapui** from **domibus-distribution-3.1-RC2-sample-configuration-and-testing.zip** and open **Domibus-MSH-soapui-project.xml**

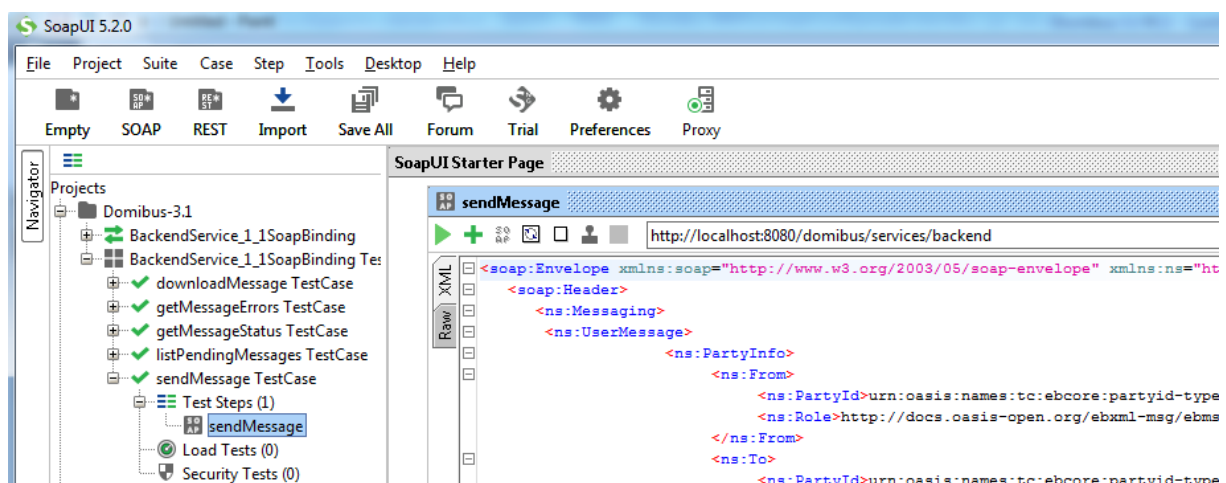


- Once the project is imported expand the BackendServices\_1\_1SoapBinding TestSuite -> sendMessage TestCase -> Test Steps and double click on the sendMessage.



The following **two** steps are only needed if you made changes to the pMode file:

- Edit the "from" and "to" PartyID according to your PMode data.
- Click on the drop down button and make sure the endpoint (hostname) is correct. If not, you need to add your backend service endpoint in here and click on 'OK'.
- Soap Client is ready to send a message from your Access Point to out Access Point. You need to click on green play button to send the message





## DEFAULT PLUGINS

By default we provide two plugins for sending and receiving/downloading messages via Domibus, a Web Service plugin and a JMS plugin.

By default, the Web Service plugin is deployed with the tomcat-full distribution.

Default JMS plugin is provided in a different archive, **domibus-distribution-3.1-RC2-default-jms-plugin.zip** including the binaries (domibus-default-jms-plugin-3.1-RC2.jar) and the configuration files (jms-plugin.xml and jms-business-defaults.properties).

Name	Type	Compr
 config	File folder	
 lib	File folder	

To use the JMS plugin, you first need to remove the Web Service plugin from **/conf/domibus/plugins** (both .jar and ws-plugin.xml) and deploy the JMS plugin from **domibus-distribution-3.1-RC2-default-jms-plugin.zip**.

Unzip **/conf/domibus/plugins** to **<CEF-eDelivery path>/domibus/conf/domibus/plugins**.

If both plugins are available within the same deploy, an additional step is required to define filters for routing the messages towards each plugin.

Connect to the administration dashboard using your credentials (by default: login = **admin**; password = **123456**) to <http://localhost:8080/domibus/home> and go to MessageFilter tab. Add the routing criteria according to your needs and click **Save**.

## ANNEX 1 PARAMETERS

Parameters	Local Access Point (Gateway "blue")	Remote Access Point (Gateway "red")
Hostname	<blue_hostname>:8080	<red_hostname>:8080
Database	MySQL database	MySQL database
Administrator Page	Username: <b>admin</b> Password: <b>123456</b> <a href="http://localhost:8080/domibus/home">http://localhost:8080/domibus/home</a>	Username: <b>admin</b> Password: <b>123456</b> <a href="http://localhost:8080/domibus/home">http://localhost:8080/domibus/home</a>
Database Schema	edelivery	edelivery
Database connector	Username: <b>edelivery</b> Password: <b>edelivery</b> <a href="jdbc:mysql://localhost:3306/domibus*">jdbc:mysql://localhost:3306/domibus*</a>	Username: <b>edelivery</b> Password: <b>edelivery</b> <a href="jdbc:mysql://localhost:3306/domibus">jdbc:mysql://localhost:3306/domibus</a>
DB username/password	edelivery/edelivery	edelivery/edelivery
PModes XML files	pmodes/domibus-gw-sample-pmode-blue.xml	pmodes/domibus-gw-sample-pmode-red.xml

\* *localhost* represents the server name that hosts the database and the application server for their respective Access Point.

## ANNEX 2 FIREWALL SETTINGS

The firewall settings may prevent you from exchanging messages between your local and remote Tomcat Access Points.

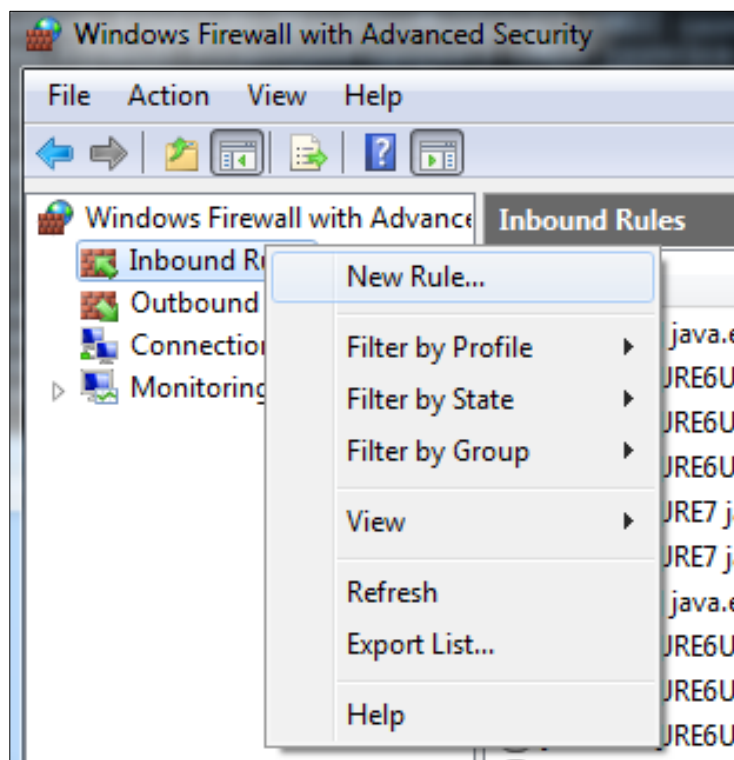
To test the status of a port, run the command `telnet <server_ip> <port>`

Tomcat uses the following ports, make sure those are opened on both machines "blue" and "red" (TCP protocol):

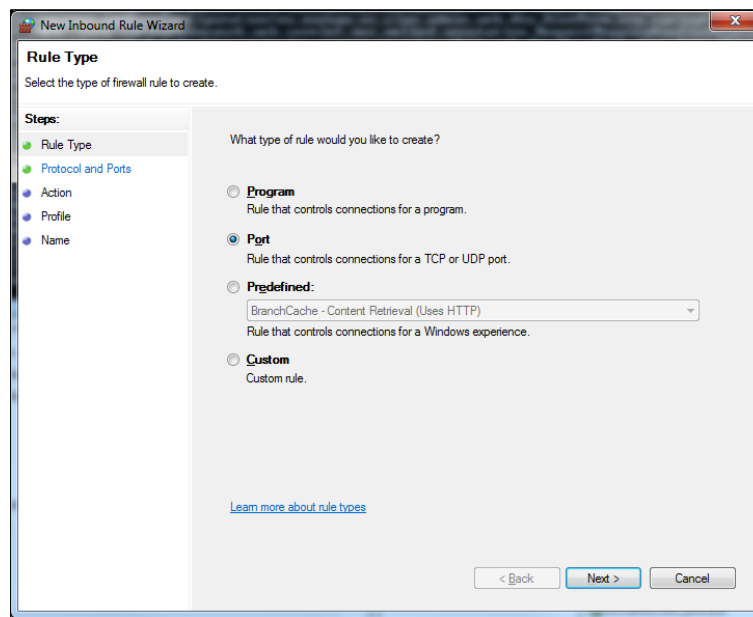
- 8080 (HTTP port)
- 3306 (MySQL port)

This is how you can open a port on the Windows Firewall

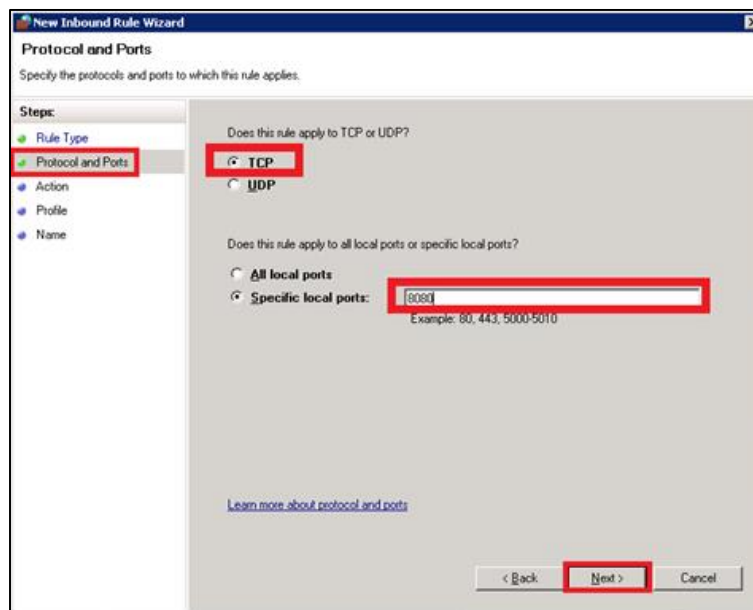
1. Click on **Start** then on **Control Panel**
2. Click on **Windows Firewall** and then click on **Advanced Settings**.
3. Right click on **Inbound Rules** then on **New Rule**:



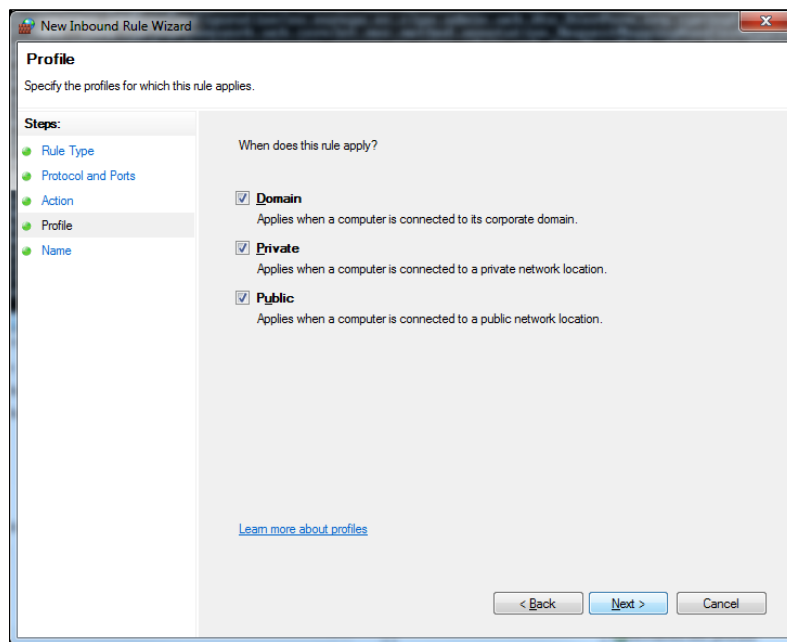
4. Select *Port* and click on *Next*:



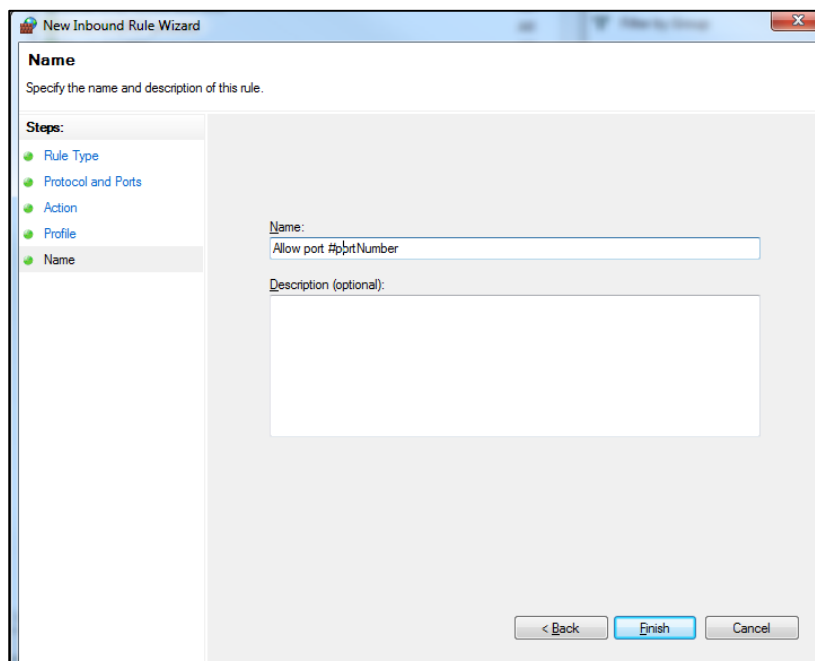
5. Enter a specific local port (e.g. 8080) and click on *Next*:



6. Click on *Next*:



7. Name the rule and click on *Finish*:





## ANNEX 3 PROCESSING MODE

Processing modes (PModes) describe how messages are exchanged between AS4 partners (*Access Point blue* and *Access Point red*). 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 (respectively "**urn:oasis:names:tc:ebcore:partyid-type:unregistered:domibus-blue**" and "**urn:oasis:names:tc:ebcore:partyid-type:unregistered:domibus-red**"). 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"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-gateway"
  xsi:schemaLocation="http://domibus.eu/configuration
file:/C:/development/git-repos/domibus/Domibus-MSH/domibus-
configuration.xsd" party="domibus_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="60"/>
  </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="partyTypeEmpty" value=""/>
      </partyIdTypes>
      <party name="red_gw"
        endpoint="http://158.166.148.65:8080/domibus/
services/msh"
        allowChunking="false"
      >
        <identifier
partyId="urn:oasis:names:tc:ebcore:partyid-type:unregistered:domibus-red"
partyIdType="partyTypeEmpty"/>
        </party>
        <party name="blue_gw"
          endpoint="http://158.166.205.81:8080/domibus/
services/msh"
          allowChunking="false"
        >
          <identifier
partyId="urn:oasis:names:tc:ebcore:partyid-type:unregistered:domibus-blue"
partyIdType="partyTypeEmpty"/>
          </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="eSens"
            policy="esensPolicy.xml"
            signatureMethod="RSA_SHA256" />
        <security name="noSigNoEnc"
            policy="doNothingPolicy.xml"
            signatureMethod="RSA_SHA256" />
    </securities>
    <errorHandlings>
        <errorHandling name="demoErrorHandling"
            errorAsResponse="true"
            businessErrorNotifyProducer="false"
            businessErrorNotifyConsumer="false"
            deliveryFailureNotifyProducer="false"/>
    </errorHandlings>
    <agreements>
        <agreement name="agreementEmpty" value="" type=""/>
    </agreements>
    <services>
        <service name="testService1" value="bdx:noprocess"
type="tcl"/>
    </services>
    <actions>
        <action name="tclAction" value="TC1Leg1"/>
    </actions>
    <as4>
        <receptionAwareness name="receptionAwareness"
retry="5;4;CONSTANT" duplicateDetection="true"/>
        <reliability name="AS4Reliability" nonRepudiation="true"
replyPattern="response"/>
        <reliability name="noReliability" nonRepudiation="false"
replyPattern="response"/>
    </as4>
    <legConfigurations>
        <legConfiguration name="pushTestcase1tclAction"
            service="testService1"

```

```

        action="tclAction"
        defaultMpc="defaultMpc"
        reliability="AS4Reliability"
        security="eSens"
        receptionAwareness="receptionAwareness"
        propertySet="ecodexPropertySet"
        payloadProfile="MessageProfile"
        errorHandling="demoErrorHandling"
        compressPayloads="true">
    </legConfiguration>
</legConfigurations>
<process name="tclProcess"
    agreement="agreementEmpty"
    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="pushTestcase1tclAction"/>
    </legs>
</process>
</businessProcesses>
</db:configuration>

```

## ANNEX 4 DOMIBUS PCONF TO EBMS3 PMODE MAPPING

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

Domibus pconf	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 <b>Messaging / UserMessage</b>	<p>Message Partition Channel allows the partition of the flow of messages from a <i>Sending MSH</i> to a <i>Receiving MSH</i> into several flows, each of which is controlled separately. An MPC also allows merging flows from several <i>Sending MSHs</i> into a unique flow that will be treated as such by a <i>Receiving MSH</i>.</p> <p>The value of this parameter is the identifier of the MPC to which the message is assigned.</p>
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 <b>Messaging/UserMessage/ PartyInfo</b>	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 <b>Messaging/UserMessage/ PartyInfo</b>	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 <b>PMode[1].Protocol.Address</b>	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 <i>Receiver MSH</i> (or <i>Receiver Party</i> ) 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 <b>PMode[1].Security.SendReceipt.NonRepudiation</b>  ReplyPattern maps to <b>PMode[1].Security.SendReceipt.ReplyPattern</b>	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 <b>PMode[1].ReceptionAwareness.Retry=true</b>  PMode[1].ReceptionAwareness.Retry.Parameters retryCount maps to <b>PMode[1].ReceptionAwareness.Retry.Parameters</b>  strategy maps to <b>PMode[1].ReceptionAwareness.Retry.Parameters</b>  duplicateDetection maps to <b>PMode[1].ReceptionAwareness.DuplicateDetection</b>	These parameters are stored in a composite string. <ul style="list-style-type: none"><li>• <i>retryTimeout</i> defines timeout in seconds.</li><li>• <i>retryCount</i> is the total number of retries.</li><li>• <i>strategy</i> defines the frequency of retries. The only <i>strategy</i> available as of now is <i>CONSTANT</i>.</li><li>• <i>duplicateDetection</i> allows to check duplicates when receiving twice the same message. The only <i>duplicateDetection</i> available as of now is <i>TRUE</i>.</li></ul>
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 <i>AgreementRef</i> element is a string that identifies the entity or artifact governing the exchange of messages between the parties.
Actions	-	Container
Action	maps to <b>Messaging/ UserMessage/ CollaborationInfo/Action</b>	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 <b>Messaging/ UserMessage/ CollaborationInfo/ Service[@type]</b>	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. <i>request</i> , <i>reply</i> ) 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	<p>maps to <b>PMode.Initiator.Role</b> or <b>PMode.Responder.Role</b> depending on where this is used. In ebMS3 message this defines the content of the following element:</p> <ul style="list-style-type: none"> <li>• For Initiator: <b>Messaging/UserMessage/PartyInfo/From/Role</b></li> <li>• For Responder: <b>Messaging/UserMessage/PartyInfo/To/Role</b></li> </ul>	<p>The required role element occurs once, and identifies the authorized role (<i>fromAuthorizedRole</i> or <i>toAuthorizedRole</i>) of the Party sending the message (when present as a child of the <i>From</i> element), or receiving the message (when present as a child of the <i>To</i> element). The value of the role element is a non-empty string, with a default value of <i>http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/defaultRole</i></p> <p>Other possible values are subject to partner agreement.</p>
Processes	-	Container
PayloadProfiles	-	Container
Payloads	-	Container



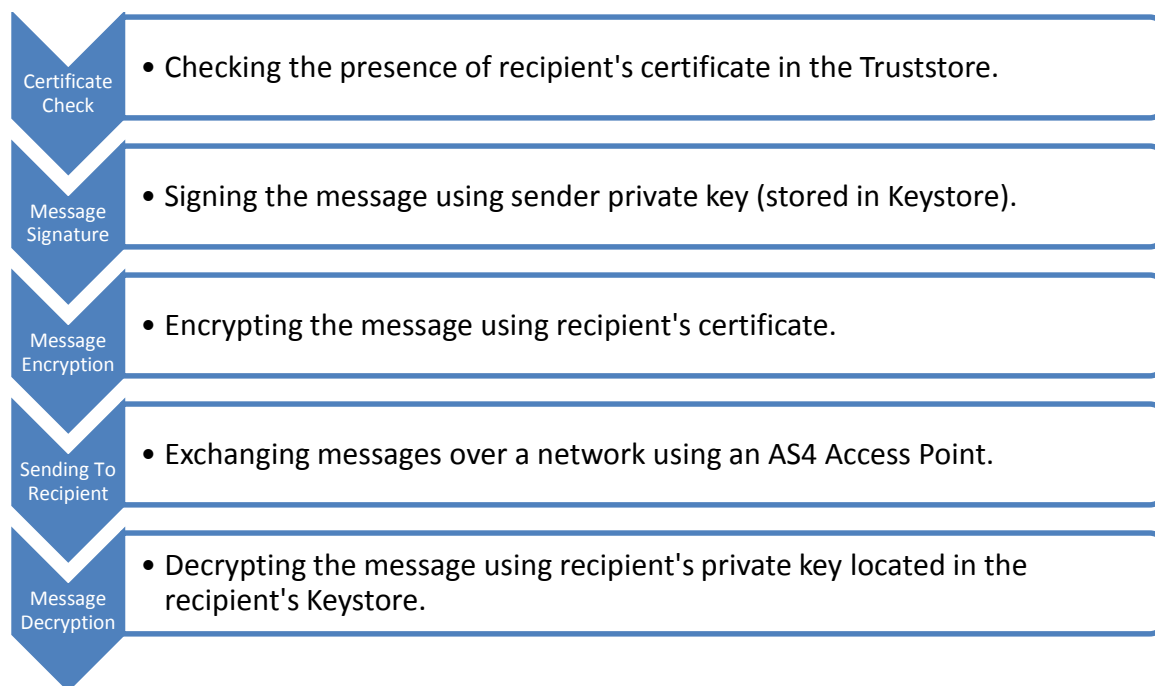
Payload	maps to <b>PMode[1].BusinessInfo.PayloadProfile</b>	<p>This parameter allows specifying some constraint or profile on the payload. It specifies a list of payload parts.</p> <p>A payload part is a data structure that consists of five properties:</p> <ol style="list-style-type: none"> <li>1. <b>name</b> (or Content-ID) that is the <b>part identifier</b>, and can be used as an index in the notation PayloadProfile;</li> <li>2. <b>MIME data type</b> (text/xml, application/pdf, etc.);</li> <li>3. <b>name of the applicable XML Schema file</b> if the MIME data type is text/xml;</li> <li>4. <b>maximum size in kilobytes</b>;</li> <li>5. <b>Boolean</b> string indicating whether the part is <b>expected</b> or <b>optional</b>, within the User message.</li> </ol> <p>The message payload(s) must match this profile.</p>
ErrorHandlings	-	Container
ErrorHandling	-	Container
ErrorAsResponse	maps to <b>PMode[1].ErrorHandling.Report.AsResponse</b>	This Boolean parameter indicates (if <i>true</i> ) 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 <i>false</i> , such errors are not sent over the back-channel.
ProcessErrorNotifyProducer	maps to <b>PMode[1].ErrorHandling.Report.ProcessErrorNotifyProducer</b>	This Boolean parameter indicates whether (if <i>true</i> ) 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 <i>User Message to be sent</i> .

ProcessErrorNotifyConsumer	maps to <b>PMode[1].ErrorHandling.Report.ProcessErrorNotifyProducer</b>	This Boolean parameter indicates whether (if <i>true</i> ) 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 <i>received User message</i> .
DeliveryFailureNotifyProducer	maps to <b>PMode[1].ErrorHandling.Report.DeliveryFailuresNotifyProducer</b>	When sending a message with this reliability requirement ( <i>Submit</i> invocation), one of the two following outcomes shall occur: - The Receiving MSH successfully delivers ( <i>Deliver</i> invocation) the message to the Consumer. - The Sending MSH notifies ( <i>Notify</i> 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 <i>legs</i> . 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 <i>General Parameters</i> ), 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 <i>BusinessInfo group</i> .
Process	-	In <i>Process</i> everything is plugged together.

[Domibus pconf to ebMS3 mapping](#)

## ANNEX 5 INTRODUCTION TO AS4 SECURITY

To secure the exchanges between Access Points "blue" and "red" (*Access Point "blue"* is sending a message to *Access Point "red"* in this example), it is necessary to set up each Access Point's *keystore* and *truststore* accordingly. The diagram below shows a brief explanation of the main steps of this process:



In order to exchange B2B messages and documents between *Access Points* blue and red, it is necessary to check the following:

For blue	For red
Check that <i>red_gw</i> certificate (public key of red) is in <i>truststore.jks</i> of blue, if not add it.	Check that <i>blue_gw</i> certificate (public key of blue) is in <i>truststore.jks</i> of red, if not add it.
Check that the <i>blue_gw</i> private key is in the <i>keystore.jks</i> , if not add it.	Check that <i>red_gw</i> private key is in the <i>keystore.jks</i> , if not add it.
In <i>domibus-security.xml</i> : the keystore alias should be <i>blue_gw</i> , you may edit the keystore password (by default <i>test123</i> ), and the path to <i>keystore.jks</i> (if you change it).	In <i>domibus-security.xml</i> edit: the alias property to <i>red_gw</i> , the keystore password (by default <i>test123</i> ) if you need to, and the path to <i>keystore.jks</i> (if you change it).

In a production environment, each participant would need a certificate delivered by a certification authority and remote exchanges between business partners would be managed by each partner's PMode (that should be uploaded on each Access Point).

*Remark:*

*It is necessary to open the required ports when Access Point blue or Access Point red is behind a local firewall. e.g. port 8080 is not opened by default in Windows; we would need to create a dedicated rule on Windows firewall to open TCP 8080 port. See annex "[Firewall Settings](#)".*