EUROPEAN COMMISSION

DIGIT
Connecting Europe Facility

Domibus 3.3 RC1

Quick Start Guide

Date: 29/06/2017

Document Approver(s):

Approver Name	Role
FERIAL Adrien	Technical Office

Document Reviewers:

Reviewer Name	Role
KHALFAOUI Aymen	CEF Support Office
FERIAL Adrien	Technical Office
MIGUEL Tiago	Developer

Summary of Changes:

Version	Date	Created by	Short Description of Changes
V0.1	15/09	EDELMAN Cedric	Initial version based on the QSG of Domibus 3.1.1
V1.0	15/09	FERIAL Adrien	Update for Domibus 3.2
V1.1	23/06/2017	MIGUEL Tiago, BACIU Cosmin	Update for Domibus 3.3-RC1

CONTENTS

INTRODUCTION

The CEF eDelivery Access Point (AP) Domibus implements a standardised message exchange protocol that ensures interoperable, secure and reliable data exchange.

Domibus is the Open Source project of the AS4 Access Point maintained by the European Commission.

The current release of Domibus supports Tomcat, WebLogic and WildFly and contains the following archives:

- **domibus-distribution-3.3-RC1-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-3.3-RC1-tomcat.war.zip** containing the Domibus war for Tomcat
- **domibus-distribution-3.3-RC1-tomcat-configuration.zip** containing the Domibus configuration files for Tomcat
- **domibus-distribution-3.3-RC1-weblogic.war** containing the Domibus war for WebLogic
- **domibus-distribution-3.3-RC1-weblogic-configuration.zip** containing the Domibus configuration files for WebLogic
- **domibus-distribution-3.3-RC1-wildfly-full.zip** containing the full WildFly distribution. Default Web Service plugin is also included in this archive and deployed as the default plugin.
- **domibus-distribution-3.3-RC1-wildfly.war** containing the Domibus war for WildFly
- **domibus-distribution-3.3-RC1-sample-configuration-and-testing.zip** containing a sample of certificates, PMode configuration files and test SoapUI project
- **domibus-distribution-3.3-RC1-sql-scripts.zip** containing SQL scripts creating the schema for MySQL and Oracle
- **domibus-distribution-3.3-RC1-default-jms-plugin.zip** containing the binaries and configuration file for the JMS plugin
- **domibus-distribution-3.3-RC1-default-ws-plugin.zip** containing the binaries and configuration file for the Web Service plugin
- **domibus-distribution-3.3-RC1-sql-scripts.zip** containing the MySQL and Oracle SQL scripts (full and migration)

PURPOSE OF THIS GUIDE

This release contains the AS4 Access Point of the CEF eDelivery Digital Service Infrastructure (DSI). For more information about this release, please refer to CEF Digital.

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 supports Tomcat 8, WebLogic 12c and WildFly 9. It is compatible with Oracle 10g+ and MySQL 5.5+. In this guide, we are covering Tomcat/MySQL configuration.

If you want to install Domibus on Wildfly or Weblogic or if you want to have more information on Domibus configuration in general, please read the Administration Guide available on the release page of Domibus.

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)

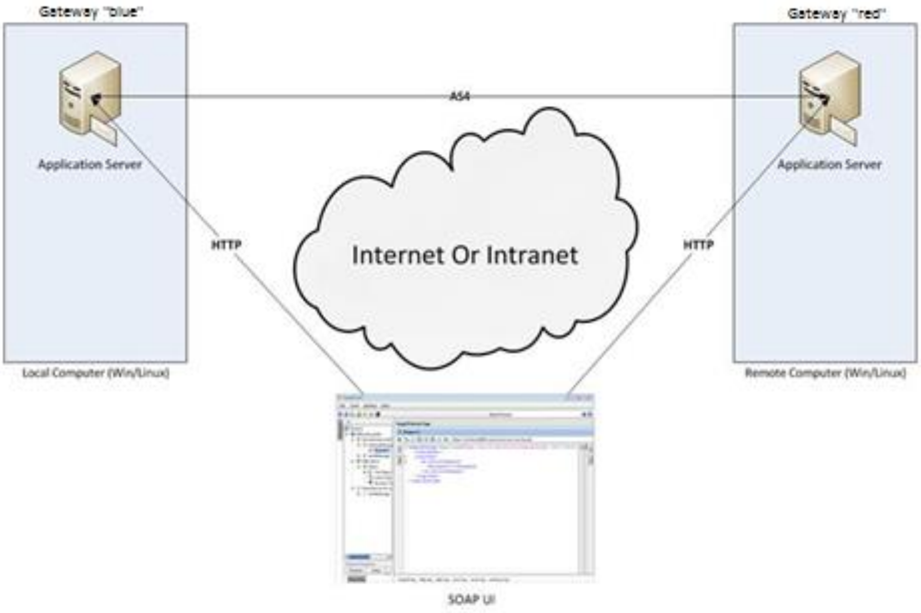


Figure 1 - Installation on two different machines

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. Proxy setup).*

Prerequisites

- Java runtime environment (JRE), version 7 or 8:
  http://www.oracle.com/technetwork/java/javase/downloads/index.html

- 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 JRE but also that the path for JRE and MySQL are set to their respective bin directory.*

Configure your environment

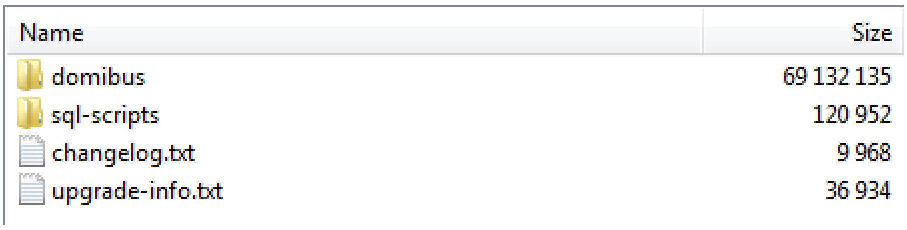
1.1. Package Overview

1.1.1. *Domibus-distribution-3.3-RC1-tomcat-full.zip*

Download the Domibus 3.3-RC1 Distribution from CEF Digital:

https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Domibus+-+v3.3+RC1

This package has the following structure:



Name	Size
domibus	69 132 135
sql-scripts	120 952
changelog.txt	9 968
upgrade-info.txt	36 934

Figure 2 - Package content

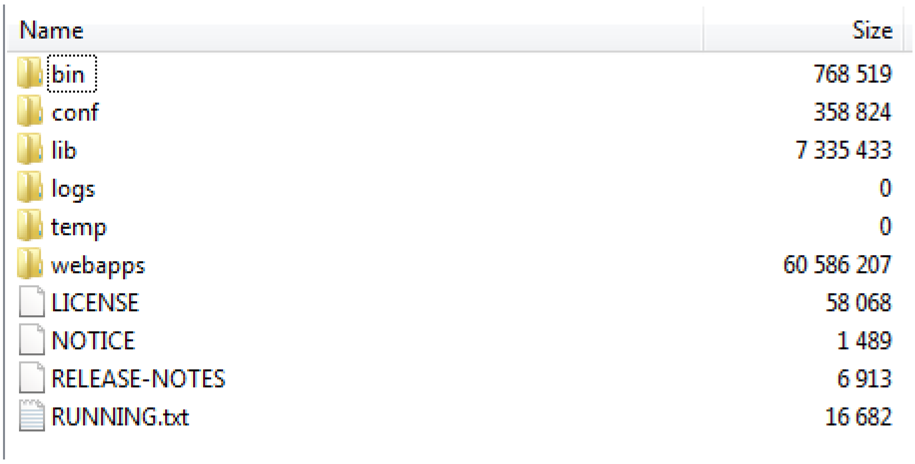
- **<CEF-eDelivery path>/domibus/bin** contains the executable batch file (Windows) and shell script (Linux) which are required to launch the Access Point.

- **<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).

Remark:

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

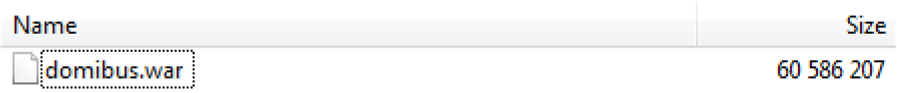
- **<CEF-eDelivery path>/domibus** contains:



Name	Size
bin	768 519
conf	358 824
lib	7 335 433
logs	0
temp	0
webapps	60 586 207
LICENSE	58 068
NOTICE	1 489
RELEASE-NOTES	6 913
RUNNING.txt	16 682

Figure 3 - eDelivery path/domibus content

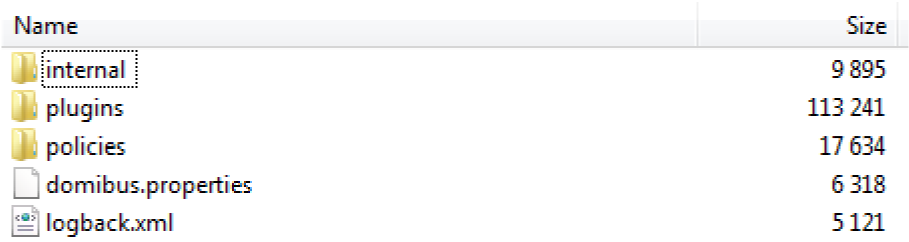
  - ○ ***conf*** folder where you will find the *configuration files (.xml* used to administer your Tomcat and the default domibus configuration files)
  - ○ ***logs*** folder where the logs are stored
  - ○ ***webapps*** folder where the WAR files are stored



Name	Size
domibus.war	60 586 207

Figure 4 - Domibus WAR file

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



Name	Size
internal	9 895
plugins	113 241
policies	17 634
domibus.properties	6 318
logback.xml	5 121

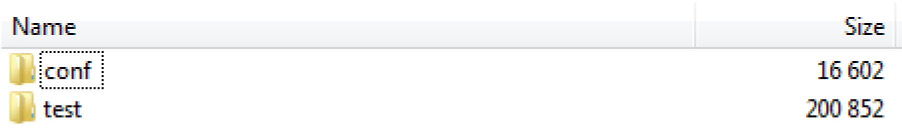
Figure 5 - Domibus configuration files

1.1.2. domibus-distribution-3.3-RC1-sample-configuration-and-testing.zip

Download the Domibus 3.3-RC1 configuration files sample from Nexus:

https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Domibus+-+v3.3+RC1

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



Name	Size
conf	16 602
test	200 852

Figure 6 - Pre-configured files for Domibus

- <**CEF-eDelivery path**>/**test** contains a SOAP UI test project.

- **<CEF-eDelivery path>/domibus/conf/pmodes** contains two AS4 processing modes xml files (one for blue and other 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** that already exists by merging it with its content.*

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-3.3-RC1-tomcat-full.zip** to a location on your physical machine, which will be referred in this document as your
    < CEF-eDelivery path >.

    b. The **/conf** folder in the **domibus-distribution-3.3-RC1-sample-configuration-and-testing.zip** should be unzipped into **<CEF-eDelivery path>/domibus**.

2. Prepare the mysql database (you will need admin rights or elevation to perform some of these operations):

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;"
```

This creates a schema named **domibus** and the user named **edelivery** having all the privileges on the schema **Domibus.**

```
mysql -h localhost -u root --password=root domibus < mysql5innoDb-3.3-RC1-rc1.ddl
```

This creates the required tables in the **domibus** schema.

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 adapt the following database properties from the property file **/conf/domibus/domibus.properties** file*

```
domibus.datasource.xa.property.serverName=localhost

domibus.datasource.xa.property.port=3306

domibus.datasource.xa.property.user=edelivery

domibus.datasource.xa.property.password=edelivery

domibus.datasource.xa.property.url=jdbc:mysql://${domibus.datasource.xa.property
.serverName}:${domibus.datasource.xa.property.port}/domibus?pinGlobalTxToPhysica
lConnection=true

domibus.datasource.url=jdbc:mysql://localhost:3306/domibus?useSSL=false

domibus.datasource.user=edelivery

domibus.datasource.password=edelivery
```

c. Add the 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).

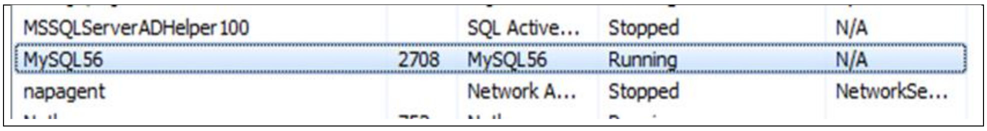
- max_allowed_packet property

```
# 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 JVM parameters

Domibus expects a single environment variable **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 lines right after the comments from the beginning of the file:

```
…
set CATALINA_HOME=<your_installation_path>
set JAVA_OPTS=%JAVA_OPTS% -Dfile.encoding=UTF-8 -Xms128m -Xmx1024m -XX:PermSize=64m -
XX:MaxPermSize=256m
set JAVA_OPTS=%JAVA_OPTS% -Ddomibus.config.location=%CATALINA_HOME%\conf\domibus
…
```

- **For Linux/Unix : Edit <CEF-eDelivery path>/domibus/bin/setenv.sh** by adding the following lines right after the comments from the beginning of the file:

```
…
export CATALINA_HOME=<your_installation_path>
export JAVA_OPTS=$JAVA_OPTS -Dfile.encoding=UTF-8 –Xms128m –Xmx1024m -
XX:MaxPermSize=256m
export JAVA_OPTS="$JAVA_OPTS -Ddomibus.config.location=$CATALINA_HOME/conf/domibus"
…
```

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

   Execute:

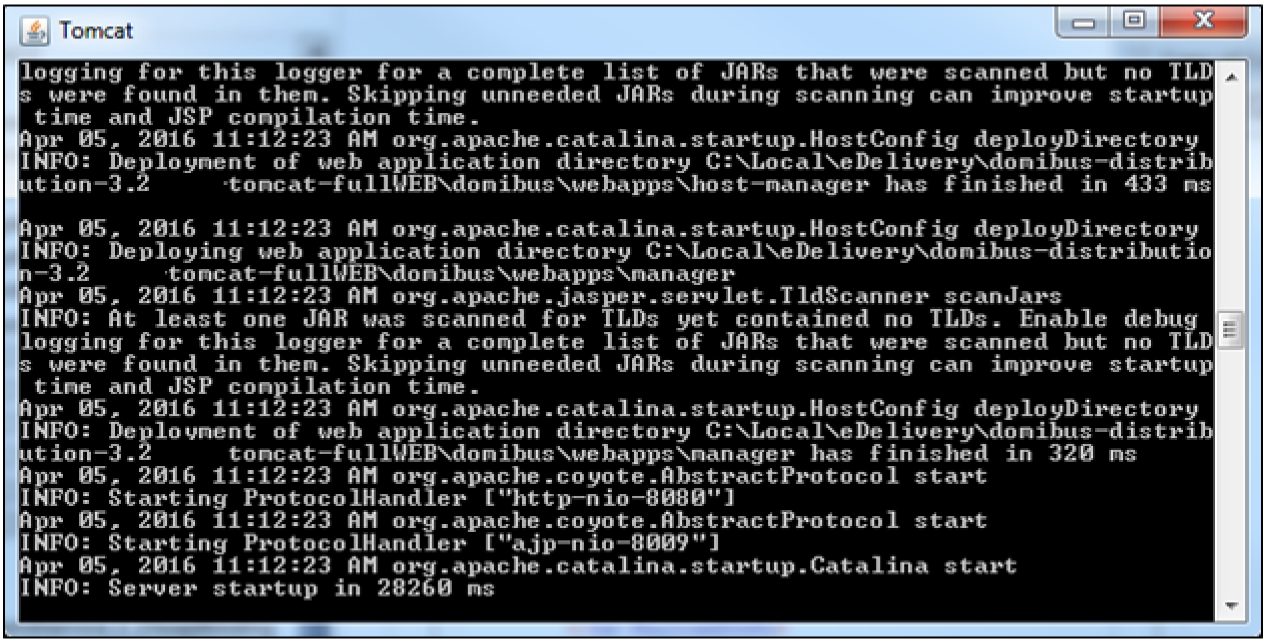
   - For Windows :

```
cd <CEF-eDeLivery path>\bin\
startup.bat
```

   - For Linux/Unix :

```
cd <CEF-eDeLivery path>/bin/
./startup.sh
```

Expected result:



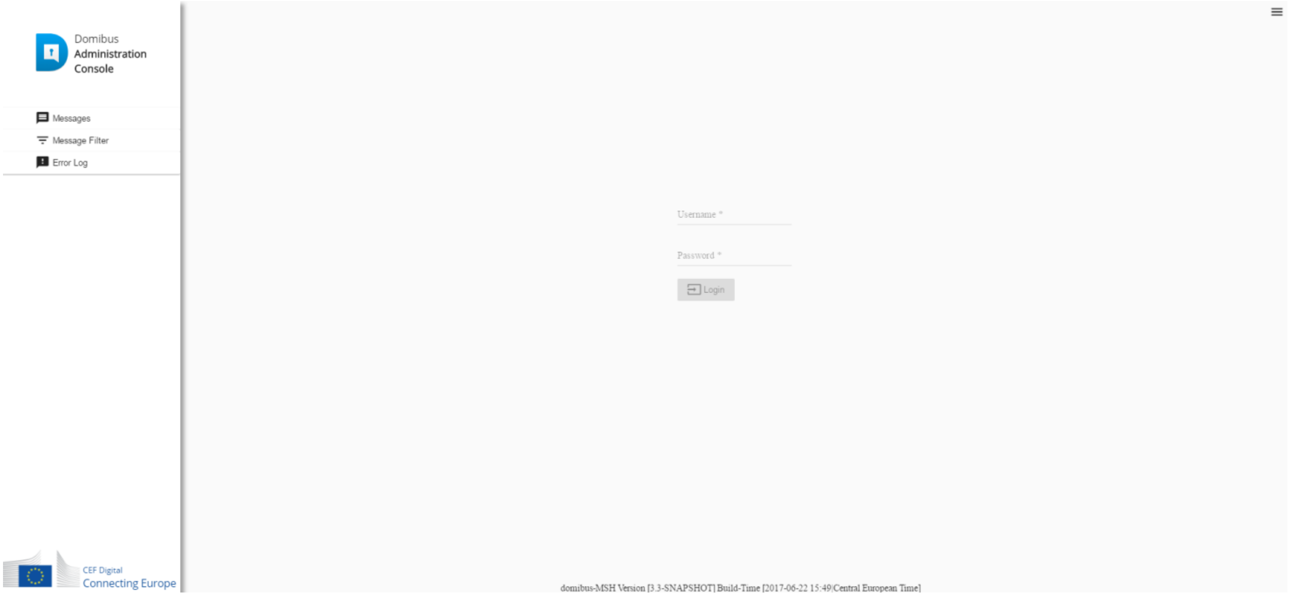
```
logging for this logger for a complete list of JARs that were scanned but no TLD
s were found in them. Skipping unneeded JARs during scanning can improve startup
 time and JSP compilation time.
Apr 05, 2016 11:12:23 AM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deployment of web application directory C:\Local\eDelivery\domibus-distrib
ution-3.2     tomcat-fullWEB\domibus\webapps\host-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-3.2     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-3.2     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
```

Figure 7 - 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 console (http://localhost:8080/domibus) in your browser as below:



domibus-MSH Version [3.3-SNAPSHOT] Build-Time [2017-06-22 15:49|Central European Time]

Figure 8 - 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 the ports of the red Access Point but also, create a new database schema, update the domibus.properties file and change the ActiveMQ ports before starting the server to avoid conflicts.*

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:

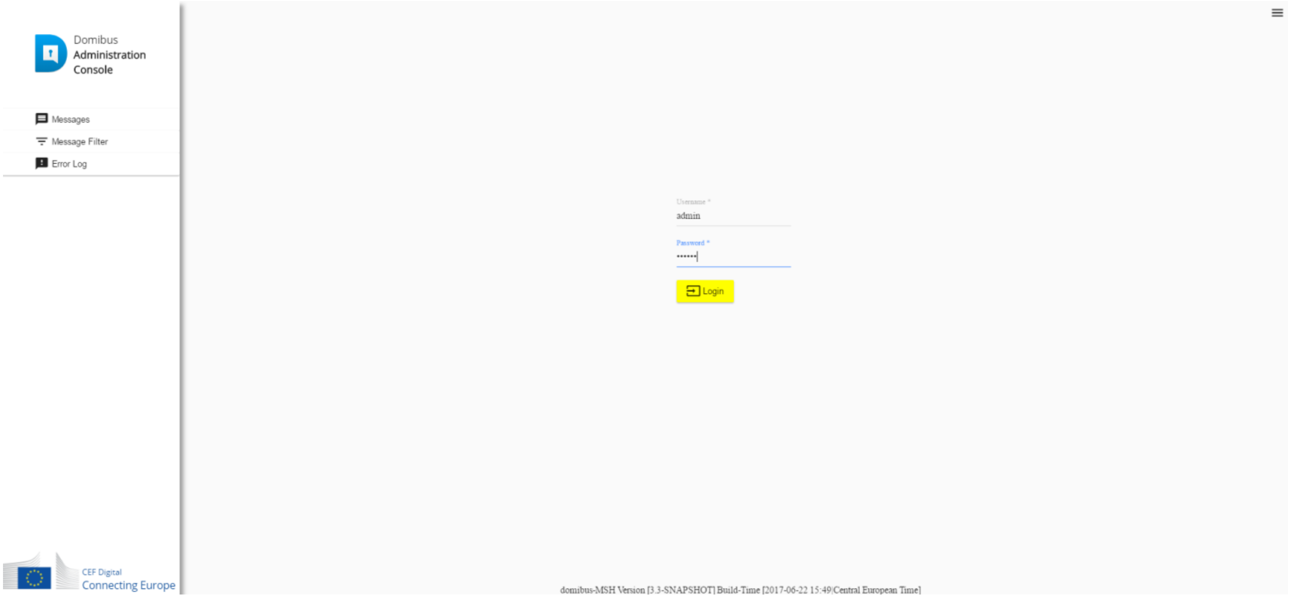
```
<party name="red_gw"
         endpoint="http://<red_hostname>:8080/domibus/services/msh"
            allowChunking="false">
        <identifier partyId="domibus-red" partyIdType="partyTypeUrn"/>
</party>
<party name="blue_gw"
         endpoint="http://<blue_hostname>:8080/domibus/services/msh"
 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

Figure 9 - PMode view

For more details about the provided PMode, please see Annex 4.

Upload the PMode file on both Access Points:

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



Username *
admin

Password *
••••••

domibus-MSH Version [3.3-SNAPSHOT] Build-Time [2017-06-22 15:49|Central European Time]

Figure 10 - Login to the administration console

b. Click on the ***PMode menu and then XML*** tab. Click on Upload button

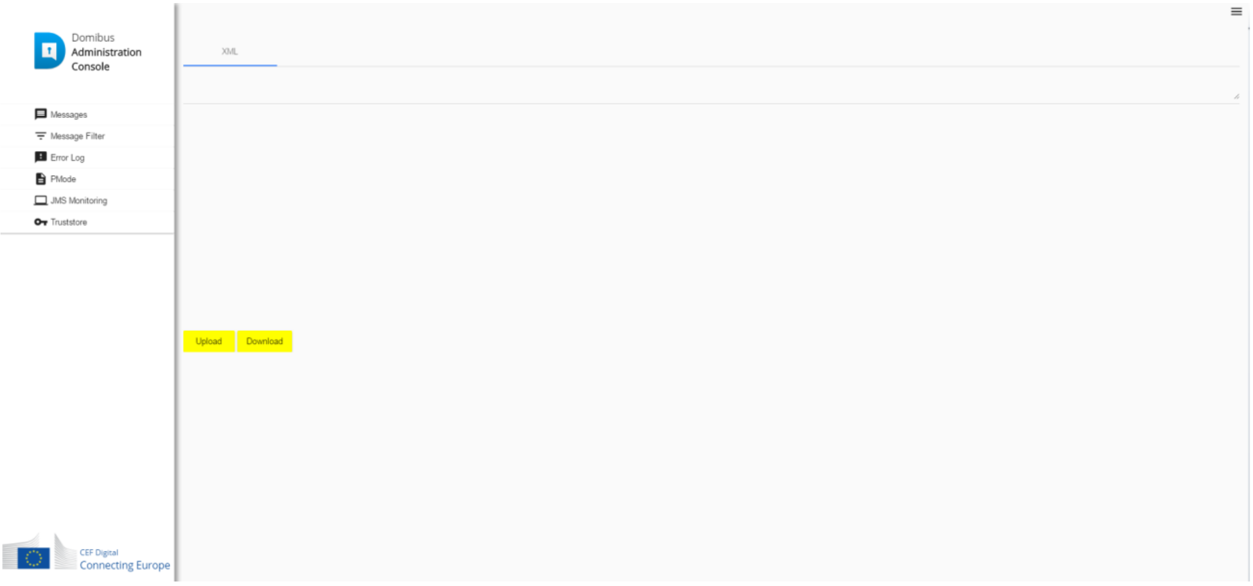


Figure 11 - PMode update

c. A popup appears where you need to select the PMode file.

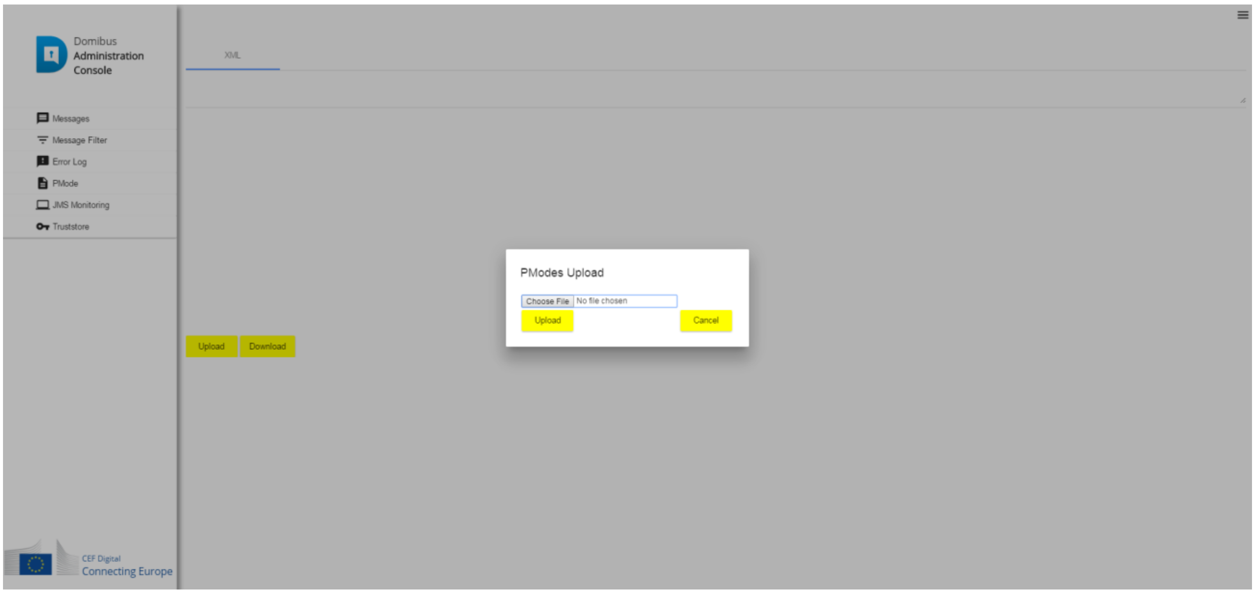


Figure 12 - PMode select file

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

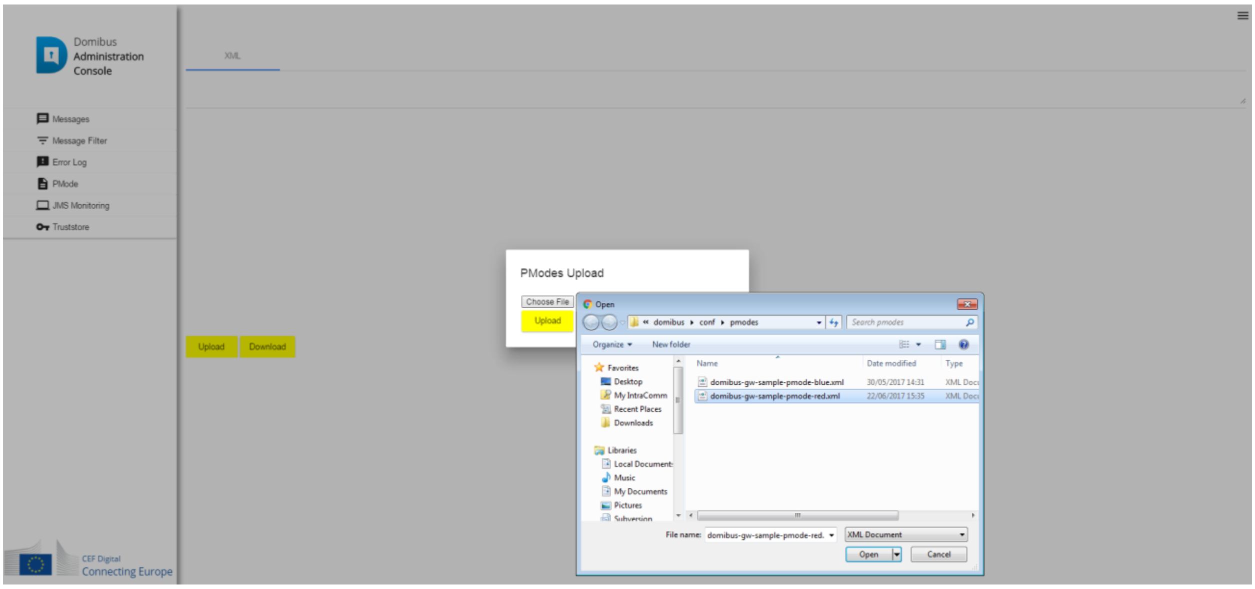
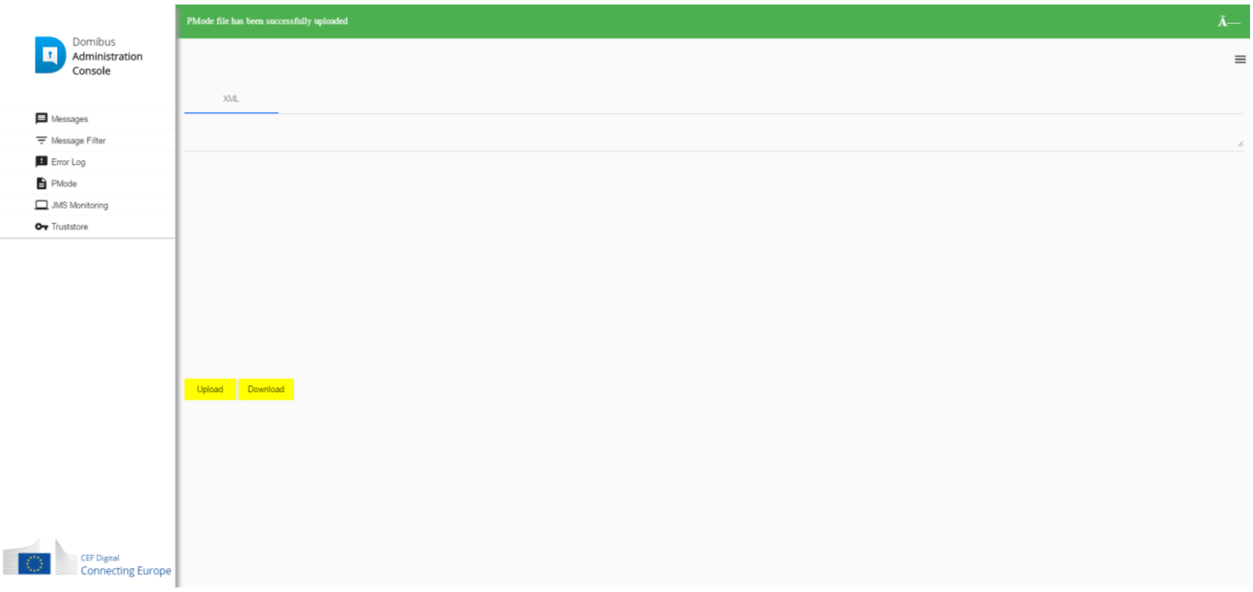


Figure 13 - PMode uploading file

e. The success of the operation is shown below:



PMode file has been successfully uploaded

Figure 14 - PMode upload success

Remark:

- *Every time a PMode is updated, the truststore is also refreshed from the file system.*

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 (Check Interface Control Document for the Default WS Plugin in the Single Web Portal for more details on the WSDL[1]).

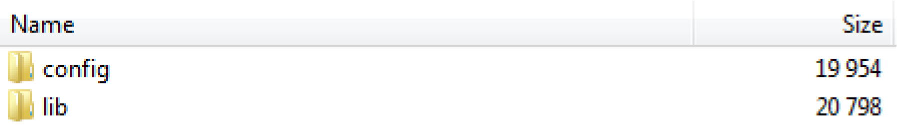
Please refer to the Test Guide for more detail regarding the Testing with a SoapUI Project.

Default plugins

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

The Web Service plugin is deployed by default with the tomcat-full distribution.

The Default JMS plugin is provided in a different archive, **domibus-distribution-3.3-RC1-default-jms-plugin.zip** including the binaries (**domibus-default-jms-plugin-3.3-RC1.jar**) and the configuration files (**jms-plugin.xml** and **jms-business-defaults.properties**).



Name	Size
config	19 954
lib	20 798

To use the JMS plugin copy the configuration files mentioned above (**jms-plugin.xml** and **jms-business-defaults.properties**) to **<CEF-eDelivery path>/domibus/conf/domibus/plugins/config** and the plugin jar file(**domibus-default-jms-plugin-3.3-RC1.jar**) to **<CEF-eDelivery path>/domibus/conf/domibus/plugins/lib**.

An additional step is required to define filters for routing the messages towards each plugin.

Open Administration Console using your credentials (by default: login = ***admin***; password = ***123456***) to http://localhost:8080/domibus and go to the **MessageFilter** page. Use Move Up and Move Down to move the preferred plugin to the top and press Save.

[1] https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Domibus+-+v3.3+RC1

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: ***admin*** Password: ***123456*** http://localhost:8080/domibus	Username: ***admin*** Password: ***123456*** http://localhost:8080/domibus
Database Schema	edelivery	edelivery
Database connector	Username: ***edelivery*** Password: ***edelivery*** jdbc:mysql://localhost:3306/domibus*	Username: ***edelivery*** Password: ***edelivery*** jdbc:mysql://localhost:3306/domibus
DB username/passwor	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):

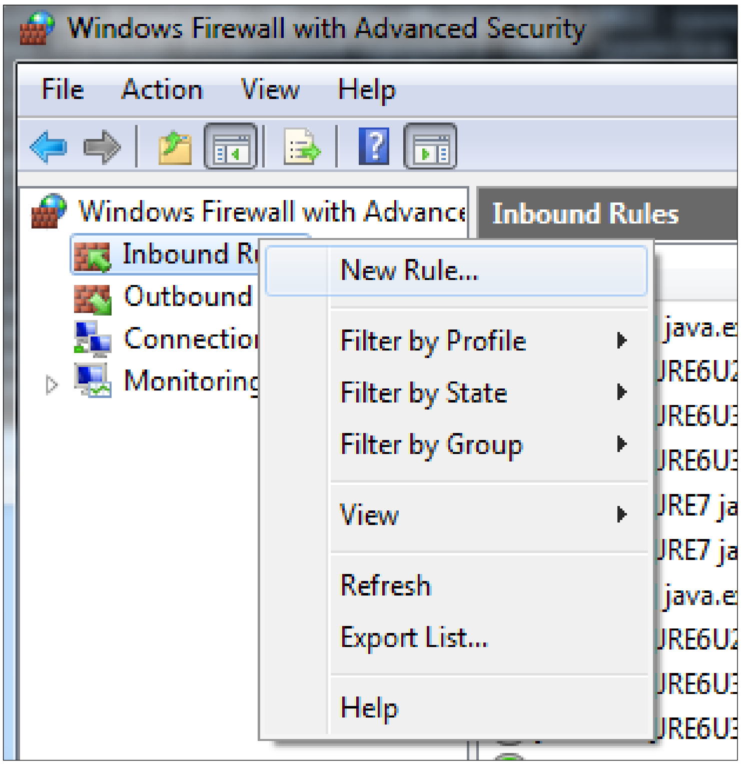
- 8080 (HTTP port)
- 3306 (MySQL port)

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

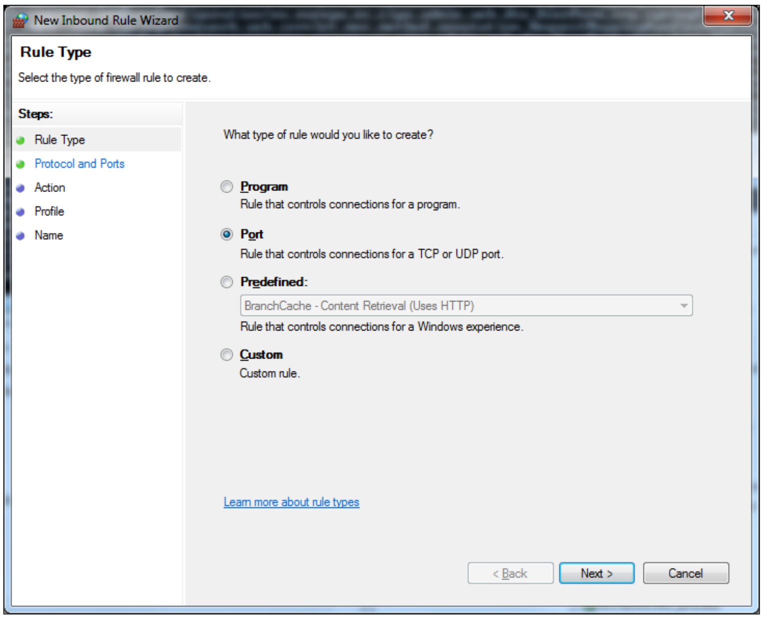
1. Click on **Start** then on **Control Panel**

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*:



Rule Type

Select the type of firewall rule to create.

What type of rule would you like to create?

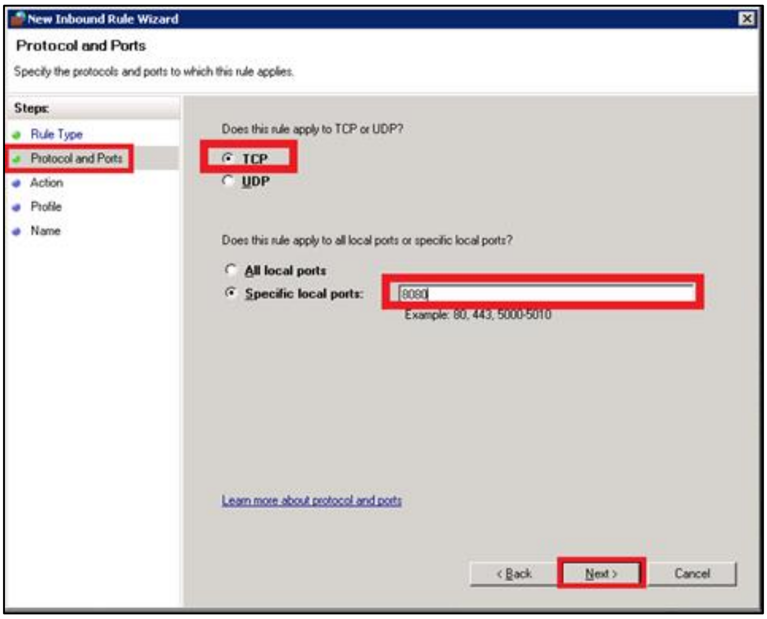
○ **Program**
Rule that controls connections for a program.

● **Port**
Rule that controls connections for a TCP or UDP port.

○ **Predefined:**
BranchCache - Content Retrieval (Uses HTTP)
Rule that controls connections for a Windows experience.

○ **Custom**
Custom rule.

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



Protocol and Ports

Specify the protocols and ports to which this rule applies.

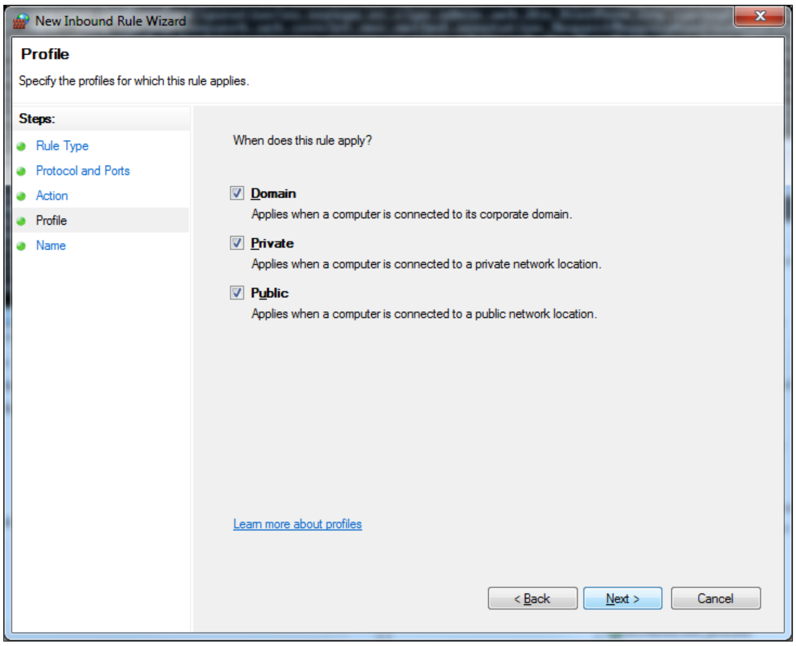
Does this rule apply to TCP or UDP?

● **TCP**
○ **UDP**

Does this rule apply to all local ports or specific local ports?

○ **All local ports**
● **Specific local ports:** 8080
Example: 80, 443, 5000-5010

6. Click on *Next*:



New Inbound Rule Wizard

Profile

Specify the profiles for which this rule applies.

Steps:

- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

When does this rule apply?

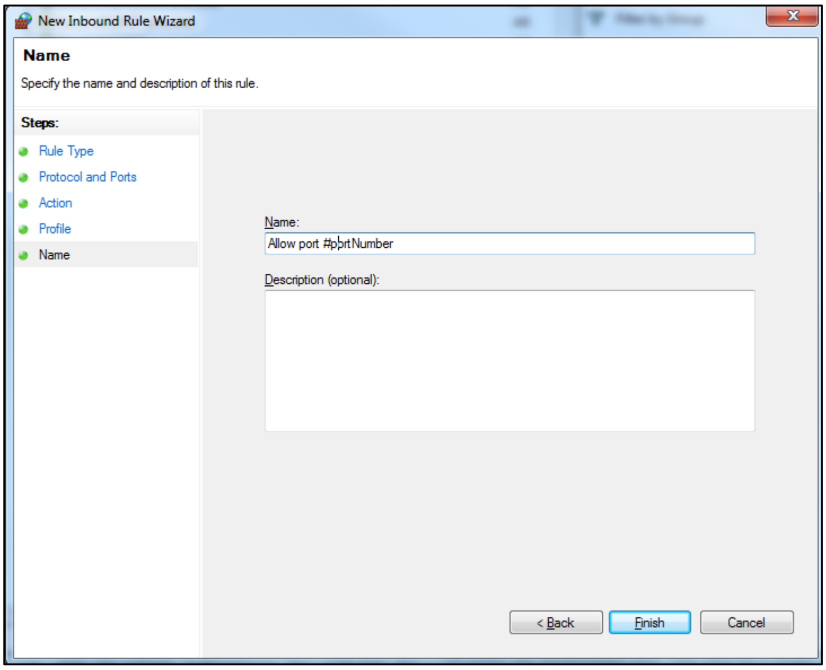
☑ **Domain**
Applies when a computer is connected to its corporate domain.

☑ **Private**
Applies when a computer is connected to a private network location.

☑ **Public**
Applies when a computer is connected to a public network location.

Learn more about profiles

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



New Inbound Rule Wizard

Name

Specify the name and description of this rule.

Steps:

- Rule Type
- Protocol and Ports
- Action
- Profile
- Name

Name:
Allow port #portNumber

Description (optional):

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 "**domibus- blue** " and "**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" 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="0" />
   </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>
```

Annex 4 Domibus pconf to ebMS3 PMode mapping

The following table provides additional information concerning the Domibus PMode configuration (pconf) 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 **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

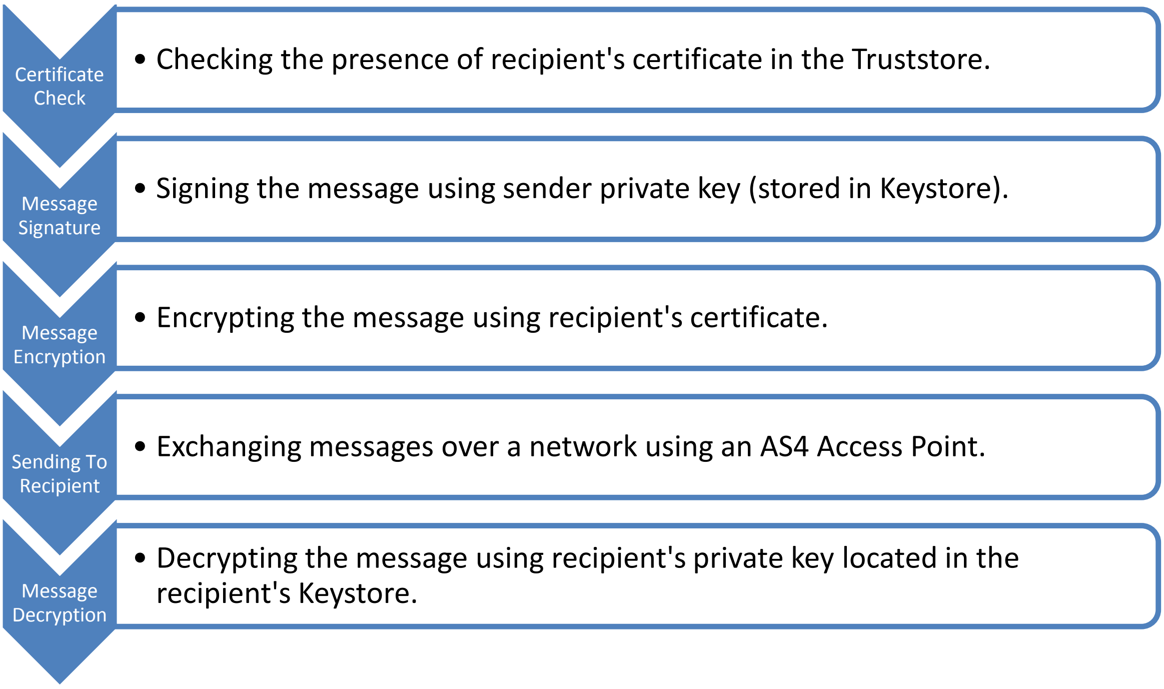
Payload	maps to **PMode[1].BusinessInfo.PayloadProfile**	This parameter allows specifying some constraint or profile on the payload. It specifies a list of payload parts. A payload part is a data structure that consists of five properties: 1. **name** (or Content-ID) that is the **part identifier**, 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.

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:



Certificate Check

- Checking the presence of recipient's certificate in the Truststore.

Message Signature

- Signing the message using sender private key (stored in Keystore).

Message Encryption

- Encrypting the message using recipient's certificate.

Sending To Recipient

- Exchanging messages over a network using an AS4 Access Point.

Message Decryption

- Decrypting the message using recipient's private key located in the recipient's Keystore.

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 *red_gw* certificate (public key of red) is in gateway_trustore.jks of blue, if not add it.	Check that *blue_gw* certificate (public key of blue) is in gateway_trustore.jks of red, if not add it.
Check that the blue_gw private key is in the gateway_keystore.jks, if not add it.	Check that red_gw private key is in the gateway_keystore.jks, if not add it.
In *domibus.properties*: the keystore alias should be *blue_gw*, you may edit the keystore password (by default *test123*), and the path to gateway_keystore.jks and gateway_truststore.jks (if you change it).	In *domibus.properties* edit: the alias property to *red_gw*, the keystore password (by default *test123*) if you need to, and the path to gateway_keystore.jks and gateway_truststore.jks (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".*