



EUROPEAN COMMISSION

DIGIT
Connecting Europe Facility

FS Plugin

Configuration Guide

Domibus File System Plugin

Version [1.01]

Status [Final]

Date: 22/06/2018

Document Approver(s):

Approver Name	Role
Adrien FERAL	CEF Technical Office

Document Reviewers:

Reviewer Name	Role
Cosmin BACIU	CEF Technical Office
Caroline AEBY	CEF Support

Summary of Changes:

Version	Date	Created by	Short Description of Changes
0.01	16/10/2017	Chaouki BERRAH	Initial Document
0.02	25/10/2017	Chaouki BERRAH	Domain Setup added
0.03	27/10/2017	Chaouki BERRAH	Internal Review
0.04	13/11/2017	Chaouki BERRAH	Updated with Cosmin Comments
1.0	21/11/2017	Chaouki BERRAH	Final
1.01	20/06/18	CEF Support	Reference to Nexus replaced by reference to CEF Digital

Table of Contents

1. INTRODUCTION	4
2. CONVENTIONS.....	5
2.1. Example: Sample Configuration file	5
2.2. References.....	6
3. PREREQUISITES.....	7
4. FILE SYSTEM CONFIGURATION PROCEDURE.....	8
4.1. File System Plugin download and installation.....	8
4.2. File System Plugin location configuration	8
4.2.1. Local.....	8
4.2.2. SMB/CIFS	9
4.2.3. SFTP	9
4.2.4. FTP	9
4.3. Message Filter configuration.....	10
4.4. Metadata.xml configuration.....	11
4.5. PMode configuration.....	12
5. TESTING	14
6. FINE TUNING	15
6.1. Delete/Archive Message	15
6.2. Specific Domain Setup.....	15
6.2.1. fs-plugin.properties Domain Configuration changes	15
6.2.2. Metadata.xml Domain Configuration changes	16
6.2.3. PMode Domain Configuration changes.....	17
7. FILE SYSTEM PERMISSIONS	19
7.1. Local File Systems:.....	19
7.2. Remote File Systems:	19
8. CONTACT INFORMATION	20

1. INTRODUCTION

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

The purpose of this guide is to provide detailed information on how to configure and deploy the File System Plugin available in Domibus 3.3 and later versions.

The instructions that are provided are valid for all the platforms supported by Domibus, including WebLogic, Tomcat and Wildfly, with MySQL or Oracle backend.

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 FS Plugin".
- **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 you to designate items like users, passwords, database etc.). It normally contains at least 2 words separated by "_".
- ***Bold and Italic*** is used for variables names that can be modified by the user.
- 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: Sample Configuration file

```
# E.g.: 1
#fspugin.domains.DOMAIN1.order=1

# Regular expression used to match the domain for the reception of messages. This regular
# expression will be evaluated
# against the Service and Action values from the incoming message separated by #.
# E.g.: DOMAIN1SampleService#.*
#fspugin.domains.DOMAIN1.messages.expression=
fspugin.domains.SOMEDOMAIN.messages.expression=SomeDomainService#SomeDomainAction

# The location of the folder that the plugin will use to manage the messages to be sent and
# received in case no domain
# expression matches. This location must be accessible to the Domibus instance. The domain
# locations must be independent
# from each other and should not overlap.
# E.g.: /home/domibus/fs_plugin_data/DOMAIN1
#fspugin.domains.DOMAIN1.messages.location=
fspugin.domains.SOMEDOMAIN.messages.location=/PATH/SOMEDOMAIN

# The user used to access the domain location specified by the property
fspugin.domains.<domain_id>.messages.location.
```

SOMEDOMAIN, **PATH**, **SomeDomainAction** are all user defined names which are chosen by the user.

2.2. References

Ref.	Document	Content outline
[REF1]	https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Domibus	File System plugin ICD document (Interface Control Document).
[REF2]	https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Domibus	Domibus Administration Guide.

3. PREREQUISITES

Domibus 3.3 or a later version needs to be already installed in a folder called ***cef_edelivery***. Please see the Domibus Administration Guide (see [REF2]) for details.

The FS Plugin can either be installed on the Sending Access Point, the Receiving Access Point or both.

4. FILE SYSTEM CONFIGURATION PROCEDURE

The FS (File System) Plugin configuration procedure includes:

- Downloading and installing the FS Plugin
- Specifying the FS Plugin location and setting its permissions
- Configuring the message filter
- Configuring the metadata.xml file
- Configuring the PMode

Each of these steps is described in details below.

4.1. File System Plugin download and installation

- Download the **domibus-distribution-X.Y.Z-default-fs-plugin.zip** from the CEF Digital site;
- Extract the zip file downloaded previously;
- Copy the **domibus-default-fs-plugin-X.Y.Z.jar** file to **cef_edelivery/conf/domibus/plugins/lib** folder, where **cef_edelivery** is the Domibus installation location);
- Copy the **fs-plugin.xml** and the **fs-plugin.properties** files to **cef_edelivery/conf/domibus/plugins/config**. These 2 files are specific to the Application Server (**tomcat, weblogic or wildfly**) used.

4.2. File System Plugin location configuration

The FS Plugin supports multiple file system types via Apache VFS. There are 4 file systems currently supported:

- Local,
- SMB/CIFS,
- SFTP,
- FTP.

These four file systems are described below.

4.2.1. Local

A local file system is simply a directory on the local physical system. The URI format is:

```
[file://]absolute-path
```


Where `absolute-path` is a valid absolute directory name on the local platform. UNC names are supported under Windows.

This type of file system does not support authentication hence the Domibus user needs read/write access to this directory. See § 7 - File System Permissions for more details.

Examples:

```
file:///home/someuser/somedir
file:///C:/Documents and Settings
/home/someuser/somedir
c:\program files\some dir
c:/program files/some dir
```

4.2.2. [SMB/CIFS](#)

A SMB/CIFS file system is a remote directory shared via Samba or Windows Share, with the following URI format:

```
smb://hostname[:port]/sharename[/relative-path]
```

Notice that a share name is mandatory.

This type of file system supports authentication via user and password domain properties. See section 3.1 of the Domibus Interface Control Document File System Plugin (see [REF1]) for more details.

Examples:

```
smb://somehost/shareA
smb://somehost/shareB/nestedit
```

4.2.3. [SFTP](#)

An SFTP file system is a remote directory shared via SFTP. Uses an URI of the following format:

```
sftp://hostname[:port][[/relative-path]
```

The path is relative to whatever path the SFTP server has configured as base directory, usually the user's home directory.

This type of file system also supports authentication via user and password domain properties. See section 3.1 of the Domibus Interface Control Document File System Plugin (see [REF1]) for more details.

Example:

```
sftp://somehost/pub/downloads/
```

4.2.4. [FTP](#)

An FTP file system is a remote directory shared via FTP. Accepts URIs of the following format:

```
ftp://hostname[:port][/relative-path]
```

The path is relative to whatever path the FTP server has configured as base directory, usually the user's home directory.

This type of file system also supports authentication via user and password domain properties. See section 3.1 of the Domibus Interface Control Document File System Plugin [see (REF1)] for more details.

Example:

```
ftp://somehost/pub/downloads/
```

Due to incompatibilities between the current version of VFS and certain FTP servers on Linux (e.g.: vsftpd), using a relative path prevents some of the plugin's functionality from working correctly. For that reason, in those cases an absolute path must be specified, e.g.:

```
ftp://somelinuxhost/home/someuser/pub/downloads/
```

1. Choose a name for the File System Plugin folder, where various types of messages will be stored (e.g: **filestore** refers to this folder in our documentation). Create the folder and make sure it can be read and written by the user running Domibus.
2. Edit the `cef_edelivery_path` /conf/domibus/plugins/conf/fs-plugin.properties.
3. Set `fsplugin.messages.location=PATH/filestore` where **PATH** is the full path to the **filestore** folder.
4. Restart Domibus.

The following folders should **automatically** be created under the **filestore** folder, after the successful start of Domibus:

- **IN:** Folder where the messages are received.
- **SENT:** Folder where the messages sent are stored.
- **FAILED:** Folder where the messages that failed to be sent are stored.
- **OUT:** Folder from where the messages are sent.

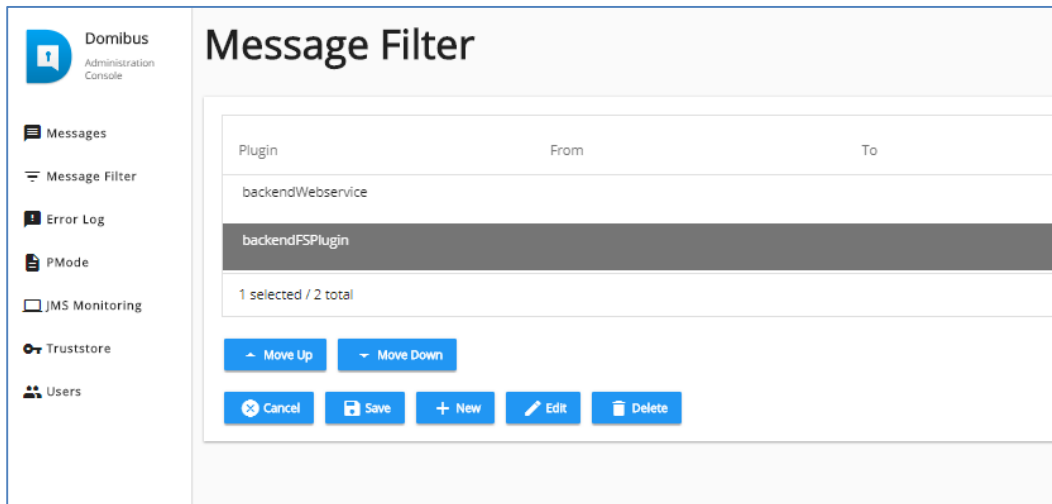
Remark:

It is possible to configure multiple locations for incoming and outgoing files to be sent with the Domibus file plugin. The procedure of configuring different domains is described in §6.2 - Specific Domain Setup.

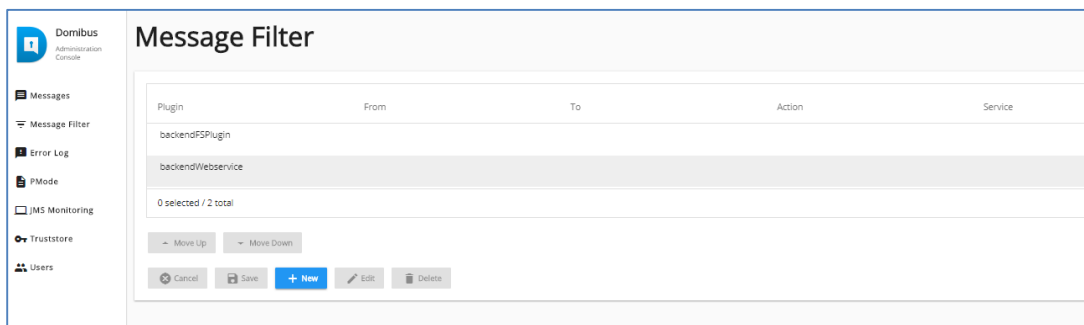
4.3. Message Filter configuration

Log on the Administration Console. Make sure that the FS Plugin is the first in the **order** of Plugins that are processed using the following procedure in the Admin Console:

1. Choose **Messages filter**:



2. Select the **backendFSPlugin**, then click on **Move Up** to move it up to the top of the list:



3. Save the changes.

4.4. Metadata.xml configuration

The **metadata.xml** contains the details of the sender and receiver party IDs as well as other information and affects all future message exchanges immediately after the file is placed in the OUT(tray) folder of the Sending Access Point.

A sample **metadata.xml** file is provided which needs to be modified to match the **PMode** specifications of the sending and receiving Access points used.

Hereafter a sample metadata.xml file:

```
<?xml version="1.0" encoding="UTF-8" ?>
<UserMessage xmlns="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/">
  <PartyInfo>
    <From>
      <PartyId type="urn:oasis:names:tc:ebcore:partyid-type:unregistered">domibus-
blue</PartyId>
      <Role>http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/ns/core/200704/initiator</Role>
    </From>
    <!--Optional:-->
    <To>
      <PartyId type="urn:oasis:names:tc:ebcore:partyid-type:unregistered">domibus-
red</PartyId>
```

```

    <Role>http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/ns/core/200704/responder</Role>
    </To>
  </PartyInfo>
  <CollaborationInfo>
    <!--You may enter the following 4 items in any order-->
    <!--Optional:-->
    <!--<AgreementRef type="">A1</AgreementRef-->
    <Service type="tc1">bdx:noprocess</Service>
    <Action>TC1Leg1</Action>
  </CollaborationInfo>
  <MessageProperties>
    <!--1 or more repetitions:-->
    <!--originalSender and finalRecipient are mandatory-->
    <Property name="originalSender">urn:oasis:names:tc:ebcore:partyid-
type:unregistered:C1</Property>
    <Property name="finalRecipient">urn:oasis:names:tc:ebcore:partyid-
type:unregistered:C4</Property>
  </MessageProperties>
</UserMessage>

```

Copy the metadata.xml file to the sender's **filestore/OUT** folder.

4.5. PMode configuration

Using the sample PMode file provided with the Domibus software:

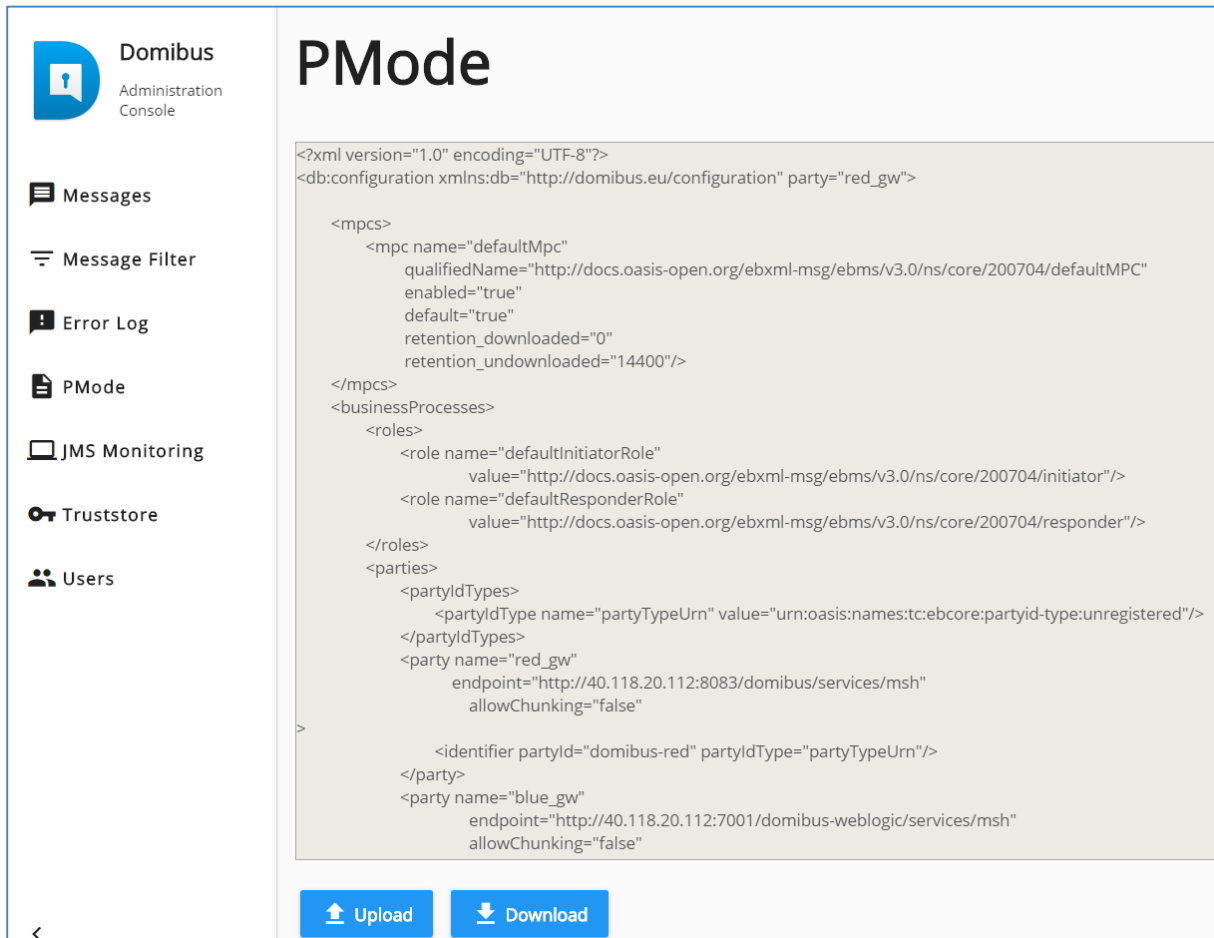
Edit the PMode file and remove the 2 instances of **payloadProfile="MessageProfile"** highlighted in yellow below:

```

<?xml version="1.0" encoding="UTF-8"?>
.....
.....
    <LegConfiguration name="pushTestcase1tc2Action"
      service="testService1"
      action="tc2Action"
      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"
.....
.....
.....

```

Upload the **PMode** to the Access point using the PMode Option in the admin console:



Domibus
Administration Console

Messages
Message Filter
Error Log
PMode
JMS Monitoring
Truststore
Users

PMode

```
<?xml version="1.0" encoding="UTF-8"?>
<db:configuration xmlns:db="http://domibus.eu/configuration" party="red_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://40.118.20.112:8083/domibus/services/msh"
        allowChunking="false"
      >
        <identifier partyId="domibus-red" partyIdType="partyTypeUrn"/>
      </party>
      <party name="blue_gw"
        endpoint="http://40.118.20.112:7001/domibus-weblogic/services/msh"
        allowChunking="false"
      >
      </party>
    </parties>
  </businessProcesses>
</db:configuration>
```

Upload Download

5. TESTING

In this section we will demonstrate, using an example, how we can send a message with a Payload from one Access Point to another.

To configure the communication parameters in the sending Access Point:

1. Place the **metadata.xml** file under the sender's **filestore/OUT** folder.
2. Any subsequent file(s) placed in the **filestore/OUT** folder will automatically be sent to the receiver's Access Point and placed under the receiver's **filestore/IN** folder.

Example:

If **payload1** is the filename of the payload file placed in the sender's **filestore/OUT** with the following contents:

```
PAYLOAD EXAMPLE FOR TESTING PURPOSES IN TEXT FORMAT
1
2
3
4
5
6
7
8
9
```

Then c3318932-72c2-4016-a210-f1517ab0d66d@domibus.eu.bin will be the new filename found in the receiver's **filestore/IN** folder.

The contents (or payload) will be the same as the one sent:

```
PAYLOAD EXAMPLE FOR TESTING PURPOSES IN TEXT FORMAT
1
2
3
4
5
6
7
8
9
```

Remark:

- *In this example, the current **Message ID**, c3318932-72c2-4016-a210-f1517ab0d66d@domibus.eu.bin, is used to rename the file sent.*
- *When the same file is sent again and again (regardless of its contents), its name will be different each time at the Receiving end. This helps to distinguish between the payloads sent.*

6. FINE TUNING

6.1. Delete/Archive Message

The **fs-plugin.properties** can be setup to either delete sent or failed messages or archive them in the sent or failed folders under filestore.

- *fsplugin.messages.sent.action=delete: Message Deleted after being sent*
- *fsplugin.messages.sent.action=archive: Message Archived in the filestore/SENT folder after being sent. Filename is **original_filename_Message_ID**, e.g: payload1_[c3318932-72c2-4016-a210-f1517ab0d66d@domibus.eu.bin](#)*
- *fsplugin.messages.failed.action=delete: Message Deleted after send failure*
- *fsplugin.messages.failed.action=archive: Message Archived in the filestore/FAILED folder after being sent. Filename is **original_filename_Message_ID**, e.g: payload1_[c3318932-72c2-4016-a210-f1517ab0d66d@domibus.eu.bin](#)*

6.2. Specific Domain Setup

6.2.1. *fs-plugin.properties Domain Configuration changes*

In order to configure multiple locations for incoming and outgoing files to be sent with the Domibus file plugin, multiple domains can be setup in order to store messages in Domain specific locations.

The following additional steps have to be performed to the purpose:

- Create a **SOMEDOMAIN** folder, before (re)starting Domibus where **SOMEDOMAIN** is a user defined name. You can have as many **SOMEDOMAIN** folders as needed (**SOMEDOMAIN1**, **SOMEDOMAIN2** etc...) for your multiple locations purpose.
- Add the following details to the **fs-plugin.properties** files (see the below highlighted sections in **Yellow**):
 - The new Domain expression which includes the domain name and location (e.g: **SOMEDOMAIN**), the Domain Service (e.g: **SomeDomainService**) and the Domain Action (e.g: **SomeDomainAction**).

```
.
# E.g.: 1
#fsplugin.domains.DOMAIN1.order=1

# Regular expression used to match the domain for the reception of messages. This regular
expression will be evaluated
# against the Service and Action values from the incoming message separated by #.
# E.g.: DOMAIN1SampleService#.*
#fsplugin.domains.DOMAIN1.messages.expression=
fsplugin.domains.SOMEDOMAIN.messages.expression=SomeDomainService#SomeDomainAction

# The location of the folder that the plugin will use to manage the messages to be sent and
received in case no domain
# expression matches. This location must be accessible to the Domibus instance. The domain
locations must be independent
# from each other and should not overlap.
# E.g.: /home/domibus/fs_plugin_data/DOMAIN1
#fsplugin.domains.DOMAIN1.messages.location=
```

```
fspPlugin.domains.SOMEDOMAIN.messages.Location=/PATH/SOMEDOMAIN
```

```
# The user used to access the domain location specified by the property
fspPlugin.domains.<domain_id>.messages.Location.
# This value must be provided if the location access is secured at the file system level so
that users from other
.
```

Remark

- Create the **SOMEDOMAIN** folder before (re)starting Domibus.
- **IN**, **OUT**, **SENT** and **FAILED** folders will be automatically created under **SOMEDOMAIN**.

6.2.2. Metadata.xml Domain Configuration changes

Insert an additional Service and Action (see highlighted section below in **Yellow**) to the existing **metadata.xml**:

```
<?xml version="1.0" encoding="UTF-8" ?>
<UserMessage xmlns="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/">
  <PartyInfo>
    <From>
      <PartyId type="urn:oasis:names:tc:ebcore:partyid-type:unregistered">domibus-
blue</PartyId>
      <Role>http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/ns/core/200704/initiator</Role>
    </From>
    <!--Optional:-->
    <To>
      <PartyId type="urn:oasis:names:tc:ebcore:partyid-type:unregistered">domibus-
red</PartyId>
      <Role>http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/ns/core/200704/responder</Role>
    </To>
  </PartyInfo>
  <CollaborationInfo>
    <!--You may enter the following 4 items in any order-->
    <!--Optional:-->
    <!--<AgreementRef type="">A1</AgreementRef-->
    <Service type="tc1">SomeDomainService</Service>
    <Action>SomeDomainAction</Action>
  </CollaborationInfo>
  <MessageProperties>
    <!--1 or more repetitions:-->
    <!--originalSender and finalRecipient are mandatory-->
    <Property name="originalSender">urn:oasis:names:tc:ebcore:partyid-
type:unregistered:C1</Property>
    <Property name="finalRecipient">urn:oasis:names:tc:ebcore:partyid-
type:unregistered:C4</Property>
  </MessageProperties>
</UserMessage>
```


6.2.3. PMode Domain Configuration changes

Insert the following additions (see highlighted section below in **Yellow**) to the existing Pmode:

Remark:

- *SomeDomainService, SomeDomainAction and SOMEDOMAIN are user defined names.*
- *SomeDomainService describes the service specific to the new Domain.*
- *SomeDomainAction describes the action that the new domain service can perform.*
- *SOMEDOMAIN is the name of the subfolder of the File System Plugin folder where messages for the new domain will be stored.*

```
<?xml version="1.0" encoding="UTF-8"?>
.....
.....
.....
    <services>
        <service name="testService1" value="bdx:noprocess"
type="tc1"/>
        <service name="SomeDomainService" value="SomeDomainService"
type="tc1"/>
    </services>
    <actions>
        <action name="tc1Action" value="TC1Leg1"/>
        <action name="tc2Action" value="TC2Leg1"/>
        <action name="SomeDomainAction" value="SomeDomainAction"/>
    </actions>
.....
.....
.....
        <LegConfiguration name="pushTestcase1tc3Action"
            service="SomeDomainService"
            action="SomeDomainAction"
            defaultMpc="defaultMpc"
            reliability="AS4Reliability"
            security="eDeliveryPolicy"
receptionAwareness="receptionAwareness"
            propertySet="ecodexPropertySet"
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"/>
    </process>
```

```
</responderParties>
  <Legs>
    <Leg name="pushTestcase1tc1Action"/>
    <Leg name="pushTestcase1tc2Action"/>
    <Leg name="pushTestcase1tc3Action"/>
  </Legs>
</process>
  </businessProcesses>
</db:configuration>
```

Once the PMode are loaded in the sender and receiver Access points, the user will be able to drop the Payload into the /PATH/**SOMEDOMAIN/OUT** folder. The payload will be received in the receiver's /PATH/**SOMEDOMAIN/IN** folder.

7. FILE SYSTEM PERMISSIONS

Domibus must be able to write the incoming messages to the specified **IN folder at the receiver end**, whether a domain setup is used or not.

7.1. Local File Systems:

For **local** file systems, the **Domibus user** must have the necessary **write permissions** to the **IN, OUT, SENT and FAILED folders**.

7.2. Remote File Systems:

For remote file systems like SMB (Server Message Block), SFTP or FTP protocols, where the location access is secured at the file system level and where users from other domains cannot access its contents, the **fs-plugin.properties** file must include the credentials of a user (or users) allowed to access the **IN, OUT, SENT and the FAILED folders** specific to each domain.

The fields to setup include are described in the following *example*. See Section 3.1 of the Domibus Interface Control Document File System Plugin (see [REF1]) for more details:

```
.....  
fsplugin.domains.DOMAIN1.messages.user= user_name  
fsplugin.domains.DOMAIN1.messages.password=user_password  
.....
```

Remark:

- *A typical error that would occur if the sent Payload cannot be written to the receiver's **IN** folder is described here:*

Example:

```
2017-10-18 15:25:08,772 [] [] ERROR e.d.p.f.w.FSSendMessagesService:66 -  
Error setting up folders for domain: SOMEDOMAIN  
eu.domibus.plugin.fs.exception.FSSetUpException: IO error setting up  
folders  
    at  
eu.domibus.plugin.fs.FSFileManager.getEnsureChildFolder(FSFileManager.java:  
104)  
    at
```

In this situation, the message is moved to the **sender's SENT** folder, even though it has not actually been successfully delivered to the **receiver's IN** folder.

The message will only be released to the **receiver's IN** tray when the correct (e.g.: **write**) permissions are set.

8. 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*