EUROPEAN COMMISSION

DIRECTORATE-GENERAL INFORMATICS

CEF e-Delivery 3.2.0

Release Notes

Author(s) : CEF
Reviewed by :

Approved by :

Version : 1.00

Date: 17/12/2015

CONTENTS

1. INTRODUCTION

1.1. Release Date and Details

CEF e-Delivery 3.2.0 was released on December 18th, 2015.

CEF e-Delivery 3.2.0 provides a simple and secure way to exchange messages between two parties based on the AS4 profile of ebMS 3.0, version 1.0.

It consists of two Web Application Archives which are the message submission/notification and the message service handler (AS4 implementation).

CEF e-Delivery 3.2.0 is a cross-platform application written in Java.

The Submission Module facilitates an easy integration with the backend systems (back offices). Therefore, it offers flexible options to configure and customize message handling based on the specific needs of the backend systems.

The Message Service Handler (MSH) is based on the Domibus 3.0.1 release by eCodex.

In addition, the Reference Web Service endpoint implementation provides a service to facilitate the testing procedure.

1.2. Prerequisites

- JBoss AS 7.1.1.Final is included in the full distribution.
- MySQL Server, version 5.6.19 or higher can be used.

2. Availability and Description

2.1. Package Overview

The full distribution package is available here:

https://joinup.ec.europa.eu/nexus/content/repositories/releases/eu/europa/ec/cipa/cef-edelivery-distribution/3.2.0-alpha-1/cef-edelivery-distribution-3.2.0-alpha-1-as4-jboss.zip

It includes:

- JBoss Application Server 7.1.1.Final.
- The two WAR files: domibus-app.war and domibus.war.
- The MySQL scripts to setup the database.
- The Quick Start Guide providing assistance to end users when installing the Pilot and its subsequent components.
- The Test Guide and Test Suite (SOAPUI project) allowing end users to complete test scenarios between two instances.

2.2. Detailed Description

2.2.1. Submission Module

The Submission Module persists messages to the database in an ebMS3 format. Messages can then be picked up by the Message Service Handler.

It exposes a simple Message Facade that handles generic message formats.

The Message Facade implements two simple operations, *submit* and *retrieve*. Both operations use generic policies to submit/retrieve a message to/from the database.

The SubmissionPolicy orchestrates the sequence of steps required by a backend application to submit a message to the database:

- **convert** the message to the internal business message object
- **validate** the message in the business object format
- **log** the message in the business object format
- **submit** the message to the database
- **convert** the response to the application-specific format
- **send** the response back to the backend application

While **submit** and **validate** are final methods that cannot be overwriten by the backend application, **convert**, **log** and **send** are abstract methods, meant to be user-specific and to fulfil the client's specific needs.

The RetrievalPolicy follows the model of the Submission policy:

- **retrieve** the message from the database
- **validate** the message in the business object format
- **log** the message in the business object format
- **convert** the response to the application-specific format
- **send** the response back to the backend application

2.2.2. Message Service Handler (MSH)

This is the actual AS4 gateway implementation and it is based on Domibus 3.0.1, released by eCodex.

2.2.3. Notification Module

The main goal of the module is to notify the backend systems (back offices) when a message arrives.

It allows the routing of a message to the proper backend based on four routing criteria:

- From
- To
- Action
- Service

It exposes a simple *NotificationFacade* that handles generic message formats.

The *NotificationFacade* implements one single operation *notify* that uses the *NotificationPolicy* to retrieve the message from the database, convert it to the application-specific format and route it to the responsible backend.

2.2.4. Basic Web Service Endpoint

A Basic Web Service endpoint is provided to illustrate how back offices can integrate with CEF e-Delivery 3.2.0.

Here are the Basic Web Service endpoint characteristics:

SOAP Web Service

- Exposes *submit* operation to send a message.

- Uses the *MessageFacade* to submit the message to the database.

This release selects the backend endpoint based on the *action* routing criteria For testing purposes in this release, PMode files are configured to use a unique *service* and *action* value. Therefore the notification module is configured to push all messages to a mock endpoint, based on the same WSDL as used in the Basic Web Service.

2.2.5. Error Handling

All exceptions are converted to *DomibusException* which extends *RuntimeException*.

These exceptions are caught and translated into *SoapFaults* by the basic endpoint.

2.2.6. Known Limitations

This section provides information about issues encountered during the test phase or lists restrictions due to the default configuration.

- Cannot send multiple attachments, only one attachment is accepted and transferred.
→ In case of multiple attachments, only the last one is sent over the wire, no error is raised/

- No error is received when the gateway at the receiver side is down.

- No error is returned when sending a message where the Sender is not supposed to be linked with the receiving MSH.

- When the sender and the receiver are both using the same gateway, sending the message will fail after it is persisted to the database.

- Maximum length set in the pMode is not enforced properly in the message

- Sending multiple messages back and forth may result in a database lock. After several retries, the sending/receiving will work again

- Creating dummy attributes or values in the message does not generate an error (WSDL limitation)

- Duplicated messages are not checked during the notification. A duplicate message will be notified twice to the receiving endpoint.