CEF eDelivery Access Point

Test Guide

Document Status:

Status
Final

Document Approver(s):

Name	Role
João RODRIGUES-FRADE	CEF eDelivery

Document Reviewers:

Name	Role
Maarten DANIELS	CEF eDelivery Technical Office
Adrien FERIAL	CEF eDelivery Technical Office
Ioana DRAGUSANU	CEF eDelivery Technical Office
Cédric EDELMAN	CEF eDelivery Support

Summary of Changes:

Version	Date	Created by	Short Description of Changes
1.0	13/05/2016	CEF eDelivery	Domibus 3.1.0 release
1.1	08/09/2016	CEF eDelivery	Updated links
1.2	23/06/2017	CEF eDelivery	Update operation "getMessageStatus" to "getStatus" with the introduction of new message status: "DOWNLOADED".
1.3	07/09/2017	CEF eDelivery	Update new operations names: "submitMessage" and "retrieveMessage".

Table of Contents

Approach and purpose of the document

The present document is the Test Guide of the CEF eDelivery Access Point sample implementation (Domibus).

This document is intended for developers that want to perform a set of checks on their Domibus installation and testers that want to have a starting point to create their own test cases.

The first chapter details the setup of the test environment and the configuration of the test tool.

The second chapter explains the test scenarios and provides guidance on how to run them.

The third chapter describes the verification of the message status in the administration console.

The fourth chapter contains contact information for the CEF Support team.

The applicable terms and conditions of CEF eDelivery can be consulted in its Master Service Arrangement, available on the CEF Digital Single Web Portal:

https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Operations#Operations-masterservice

Glossary

The key terms used in this Test Guide are defined in the CEF Definitions section on the CEF Digital Single Web Portal:

https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/CEF+Definitions

The key acronyms used in this Test Guide are defined in the CEF Glossary on the CEF Digital Single Web Portal:

https://ec.europa.eu/cefdigital/wiki/pages/viewpage.action?spaceKey=CEFDIGITAL&title=CEF+Glossary

1. PREREQUISITES

This chapter details the setup of the test environment and the configuration of the test tool.

1.1. Install and configure two AS4 Access Points

This Test Guide assumes that two AS4 Access Points "blue" and "red" are installed and configured properly as defined in the Quick Start Guide of the latest CEF eDelivery Access Point release package.

1.2. Install SoapUI

Install SoapUI, an open source test tool for Web Service testing. A free version without restrictions is available and this free version is sufficient to run the tests in this package.

Download SoapUI free[1] via and follow the installation instructions on the website. The version used to create the tests is SoapUI 5.2.1[2].

1.3. Configure SoapUI

Inside SoapUI, create a new workspace (File->New Workspace) and load the project in this package (AS4-test-guide-soapui-project.xml) using the import functionality of SoapUI (File->Import Project).

In the Navigator in SoapUI, left-click on the project "AS4_test_guide" and select "Custom Properties" in the Properties panel.

- Set the Property named "localUrl" to the IP address and port of the machine that is running Access Point "blue".
- Set the Property named "remoteUrl" to the IP address and port of the machine that is running Access Point "red".

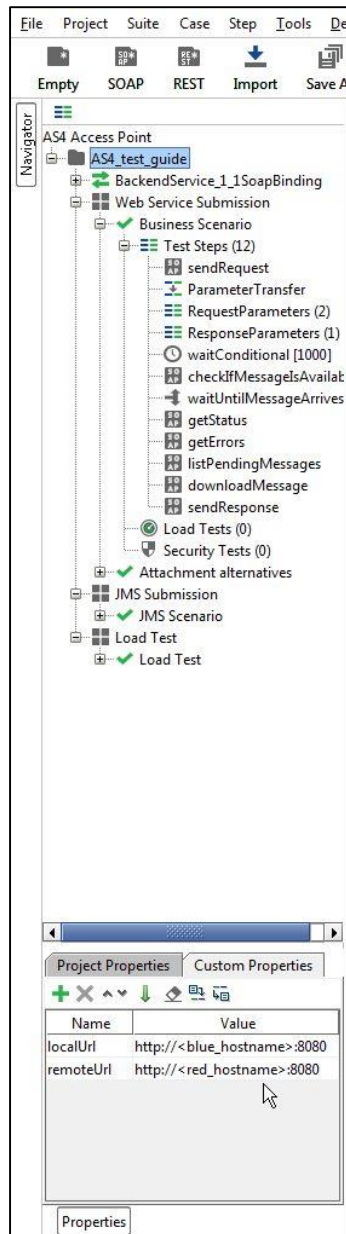
[1] http://www.soapui.org/

[2] https://www.soapui.org/downloads/soapui/source-forge.html

Figure 1 - Custom Properties

Now the service endpoints in the test steps will automatically be created from these values and no further configuration is required.

The WSDL of the services were loaded when creating the test project and they are stored in the project file. If they need to be reloaded, ensure that a local Access Point is up and running. Otherwise the URL of the WSDL (e.g http://localhost:8080/domibus/services/backend?wsdl) will not be accessible.

MTOM (Message Transmission Optimization Mechanism) is a method to efficiently send binary data to and from Web Services. This option is already pre-configured in the test request properties that need it. Changing this value might result in failures when transmitting the binary content of the exchanged messages.
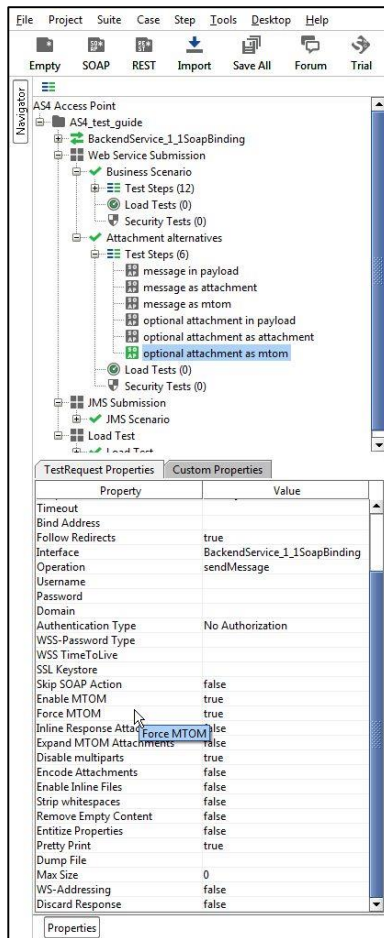
Figure 2 – Test Request properties

No authentication or authorization is required in the current version of the setup. The default setup of the SoapUI project will be sufficient to run the test case.

A default installation based on the Quick Start Guide[3] allows users to send messages via the default Web Service plugin[4]. The Quick Start Guide includes an optional section on deploying the default JMS plugin[5]. To be able to test sending messages to the JMS plugin via the sample SoapUI project, the Apache ActiveMQ libraries[6] need to be added to the lib folder of SoapUI. A restart of SoapUI is required to take the load the new libraries.

---

[3] The Quick Start Guide is available in the documentation section of the Domibus release via
https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Domibus

[4] The Web Service Plugin ICD is available in the documentation section of the Domibus release via
https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Domibus

[5] The JMS plugin ICD is available via is available in the documentation section of the Domibus release via
https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Domibus

[6] http://activemq.apache.org/download.html

2. TEST SCENARIOS

This chapter explains the test scenarios and provides guidance on how to run them.

2.1. Web Service Submission

2.1.1. Business Scenario

In this scenario, the sending Access Point (party blue_gw with ID "domibus-blue") will first send a request to the receiving Access Point (party red_gw with ID "domibus-red"). Then, the receiving Access Point is being polled for new messages. When the new message has arrived, it will be downloaded from the receiving Access point. Finally the Access Points will switch roles and the receiving Access Point will send a response to the sending Access Point.
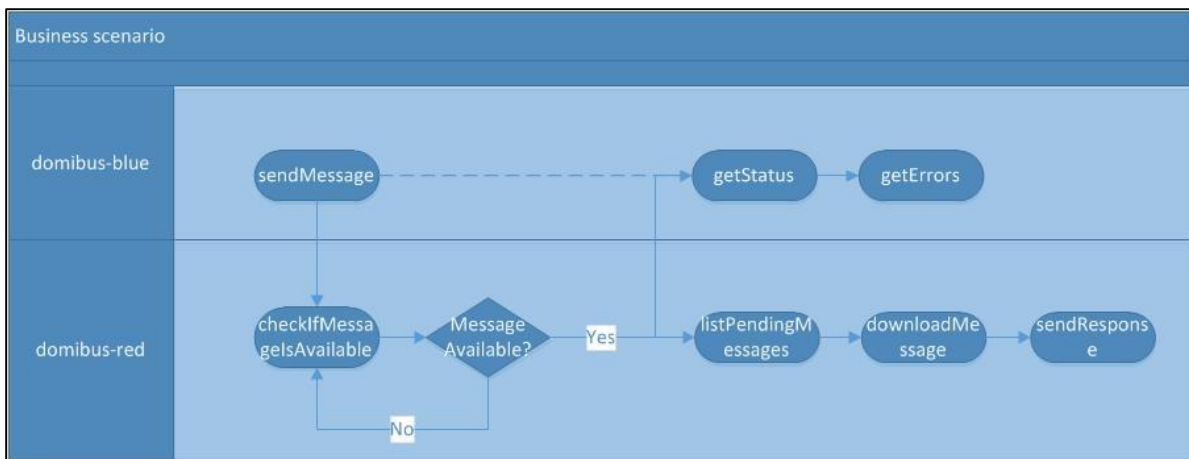

Figure 3 – Overview of the business scenario

The detailed steps are listed below:

1. [domibus-blue] Step "sendMessage" submits a message to the Web Service plugin of the Access Point that is located at localUrl. The message contains an XML message according to the sample PMode configuration files as defined in the Quick Start Guide ("businessContentPayload"). Its MimeType must be set to "text/xml". The "businessContentAttachment" is optional and can contain any type of binary attachment. Regardless of the specific type of attachment used, its content type must be set to "application/octet-stream". In this specific test step, the Attachment is not present. To modify the Payload or Attachment, refer to chapter 2.1.2 "Attachment alternatives". The response from the Web Service plugin contains the message ID of the AS4 message that will be exchanged between the sending and receiving Access Points.

2. [domibus-blue] In the background, the Access Point running at localUrl uses the message from step 1 as input to construct an AS4 message and send it to the Access Point that is running at remoteUrl.

3. [domibus-red] Steps "ParameterTransfer", "RequestParameters" and "ResponseParameters" are used for storing the MessageID, FromPartyID and ToPartyID parameters of the AS4 message. These parameters will be used in subsequent steps for verifying that the message is correctly received and can be downloaded from the receiving Access Point.

4. [domibus-red] Steps "waitConditional", "checkIfMessageIsAvailable" and "waitUntilMessageArrives" are looping and waiting for the message to arrive at the receiving Access Point.

5. [domibus-red] In the background, the Access Point that is running at remoteUrl will receive the message that is sent in step 2. Upon reception, the message will be stored in the database of the Access Point. Afterwards, the message is available via the "listPendingMessages" and "retrieveMessage" Web Service operations exposed by the Access Point. If the "retrieveMessage" operation succeeds, the Access Point will delete the message from the database (assuming that the retention parameters are set according to the sample configuration from the Quick Start Guide).

6. [domibus-blue] Step "getStatus" calls the "getStatus" operation at the sending Access Point. If the message is correctly received and acknowledged by the receiving Access Point, the operation will return the value "ACKNOWLEDGED".

7. [domibus-blue] Step "getMessageErrors" calls the "getMessageErrors" operation at the sending Access Point. If the message is correctly received and acknowledged by the receiving Access Point, the operation will return a response where the "getMessageErrorsResponse" element is empty.

8. [domibus-red] Step "downloadMessage" submits a retrieve request to the Web Service plugin of the receiving Access Point that is located at remoteUrl. The input of the operation is

the MessageID stored in step 3. The output of the operation is a response containing the contents of the message sent and additional information such as the message timestamp.

9. [domibus-red] Step "sendResponse", submits a message to the Access Point that is located at localUrl. The message contains an XML message according to the sample PMode configuration files as defined in the Quick Start Guide ("businessContentPayload"). Its MimeType must be set to "text/xml". The "businessContentAttachment" is optional and can contain any type of binary attachment. Regardless of the specific type of attachment used, its content type must be set to "application/octet-stream". In this specific test step, the Attachment is not present. To modify the Payload or Attachment, refer to chapter 2.1.2 "Attachment alternatives". The response from the Web Service plugin contains the message ID of the AS4 message that will be exchanged between the sending and receiving Access Points.

10. [domibus-red] In the background, the Access Point running at remoteUrl will use the message from step 9 as input to construct an AS4 message and send it to the Access Point that is running at localUrl.

To run this test scenario, open the "demo" test case and click the green "play" button. Each test step will be executed and a green progress bar will indicate the successful completion of the test steps.
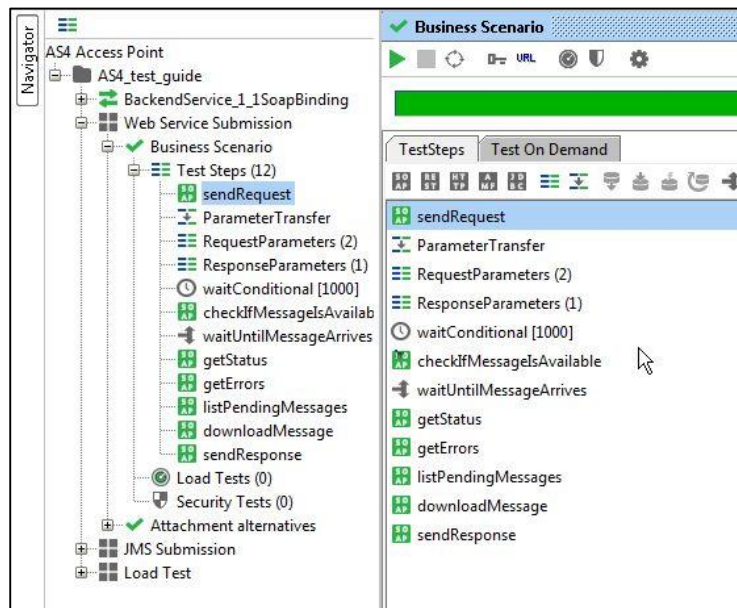

Figure 4 – Running the business scenario

2.1.2. Attachment alternatives

The sample PMode configuration files as defined in the Quick Start Guide define two payloads:

- A mandatory "businessContentPayload" (cid:message) for which the MimeType must be set to "text/xml".

- An optional "businessContentAttachment" (cid:attachment) for which the MimeType must be set to "application/octet-stream".

The test case "Attachment alternatives" contains different test steps with alternative options for submitting a message to the default Web Service plugin.

1. Test step "message in payload" contains an XML message which is base64 encoded inside the Soap Body of the request to the Web Service plugin.

2. Test step "message as attachment" contains an XML message which is added as an attachment in the request message of SoapUI. SoapUI will base64 encode the message and include it in the Soap Body of the request to the Web Service plugin.

3. Test step "message as mtom" contains an XML message which is added as an attachment in the request message of SoapUI. Since the TestRequest Properties "Enable MTOM" and "Force MTOM" are enabled, SoapUI will send the message as a multipart MIME message and include the XML in a separate MIME part of the request to the Web Service plugin.

4. Test step "optional attachment in payload" adds an optional attachment in addition to the XML message. The attachment is base64 encoded inside the Soap Body of the request to the Web Service plugin.

5. Test step "optional attachment as attachment" adds an optional attachment in addition to the XML message. The attachment is added as an attachment in the request message of SoapUI. SoapUI will base64 encode the message and include it in the Soap Body of the request to the Web Service plugin.

6. Test step "optional attachment as mtom" adds an optional attachment in addition to the XML message. The attachment is added as an attachment in the request message of SoapUI. Since the TestRequest Properties "Enable MTOM" and "Force MTOM" are enabled SoapUI will send the message as a multipart MIME message and include the attachment in a separate MIME part of the request to the Web Service plugin.

To visualise the raw message that SoapUI sends, click on the tab "Raw" after submitting the message.

Figure 5 – Visualising the raw message that SoapUI sends.

2.2. JMS Submission

The default JMS plugin provides an alternative way to submit messages to a sending Access Point. Instead of performing a Web Service request, users can post a message to a JMS queue on the sending Access Point. The sending Access Point will read the message from its queue, construct an AS4 message and send it to the receiving Access Point where it will be available for download.

The test step sendMessage in the JMS Scenario test suite contains a groovy script to post a message to the JMS queue of the sending Access Point.

The script assumes that the configuration of the sending Access Point is done according to the instructions in the Quick Start Guide. If custom settings are used, the variables in the script need to be updated.

Upon running the script, the message is posted to the JMS queue of the sending Access Point. If the script executes successfully, a log message "message sent" will be written to the script Log Output panel in SoapUI. Contrary to the Web Service plugin, the JMS plugin does not return the message ID that will be used for constructing the AS4 message which is sent to the receiving Access Point. However, optionally, the message ID can be set in the JMS message and in this case the sending Access Point will use this message ID (if it is not a duplicate message ID which has been used before).
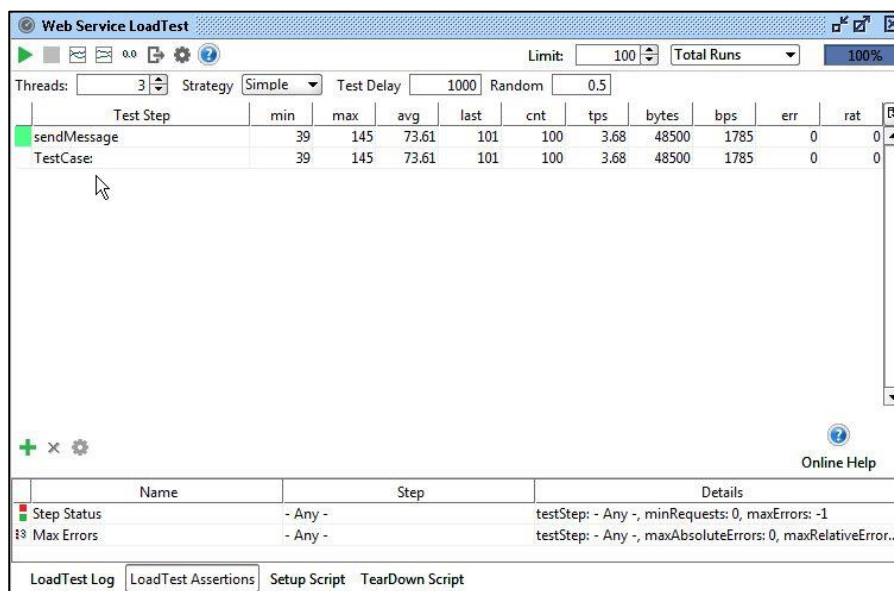
Figure 6 – Sending a JMS message

2.3. Load Test

The Web Service LoadTest will send a number of messages in parallel to the Web Service interface of the sending Access Point.

In the default configuration of the LoadTest, the same small message is being sent 100 times in 3 parallel threads using a simple load testing strategy. This LoadTest can be updated to send other or larger messages and to use different settings based on specific needs. To update the attachments of the load test, follow the guidelines described in 2.1.2 "Attachment alternatives".

The LoadTest will complete successfully if all the messages are successfully submitted to the Web Service plugin of the Sending Access point. However, this does not mean that all messages will be transferred successfully between the sending and the receiving Access Points. The latter can be verified in the message log of the Administration Console of both the sending and receiving Access Points.
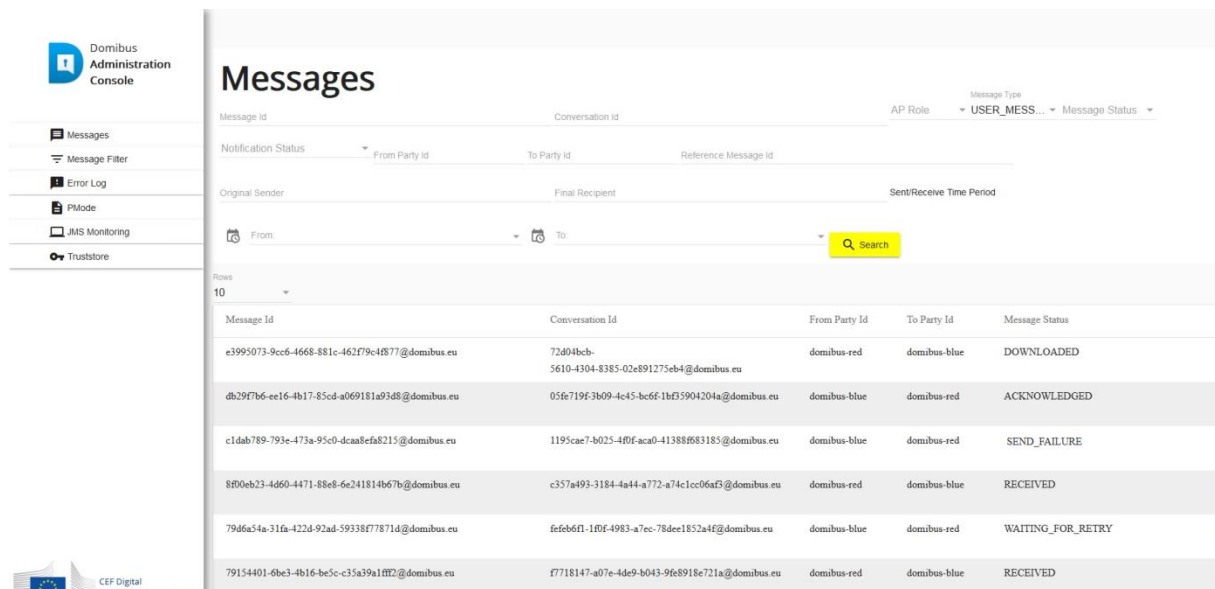

Figure 7 – Executing a load test

3. VERIFYING THE MESSAGE STATUS IN THE ADMINISTRATION CONSOLE

The status of messages that are sent or received can be verified in the administration console. Both administration consoles of the sending and receiving Access Point can be used for this purpose. Upon sending a message to the Web Service plugin, the synchronous response contains the message ID of the AS4 message that will be exchanged between the sending and receiving Access Points. This "messageID" can be used to verify the status of the message in the sending and receiving Access Points.

As an example, the figure below shows that the message with ID "db29f7b6-ee16-4b17-85cd-a069181a93d8@domibus.eu" has the status "ACKNOWLEDGED" on the sending Access Point. This indicates that the message has been received successfully received and acknowledged from the receiving Access Point.

As another example, the same figure below shows that the message with ID "8f00eb23-4d60-4471-88e8-6e241814b67b@domibus.eu" has the status "RECEIVED" on this Access Point. It indicates that the message has been successfully sent from the other Access Point to this Access Point.

As a third example, the same figure below shows that the message with ID "e3995073-9cc6-4668-881c-462f79c4f877@domibus.eu" has the status "DOWNLOADED" on this Access Point. It indicates that the initially received message has been successfully downloaded from this Access Point (assuming that the retention time for downloaded messages has not yet expired for this message).

Figure 8 – Verifying the message status in the administration console.

If a message cannot be sent because for example the other Access Point is not available, then the administration console on the sending Access Point will indicate that the message is in state "SEND_FAILURE" once the number of "SendAttempts" has reached the value in "SendAttemptsMax". Note that with the current component configuration, the message sender is not notified in any other way of the message failure (since the "submitMessage" operation synchronously returned an HTTP 200 OK with the "messageID"). In the figure above, the message with ID "c1dab789-793e-473a-95c0-dcaa8efa8215@domibus.eu" has the status "SEND_FAILURE" on the sending Access Point.

While the retrial policy is active and a message is being resent, its status will be "WAITING_FOR_RETRY" and the "NextAttempt" value will define the time when the next retry attempt will be made. In the figure above, the message with ID "79d6a54a-31fa-422d-92ad-59338f77871d@domibus.eu" has the status "WAITING_FOR_RETRY" on the sending Access Point.

4. 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