



EUROPEAN COMMISSION

DIGIT
Digital Europe Programme

eDelivery Access Point

Test Guide

© European Union, 2023

Reuse of this document is authorised provided the source is acknowledged. The Commission's reuse policy is implemented by Commission Decision 2011/833/EU of 12 December 2011 on the reuse of Commission documents.

Date: 18/08/2023

Document Status:

Status
Final

Document Approver(s):

Name	Role
João RODRIGUES-FRADE	eDelivery
Bogdan DUMITRIU	Project Manager

Document Reviewers:

Name	Role
Ahmed Ghouili	eDelivery Technical Office
Ioana DRAGUSANU	eDelivery Technical Office

Summary of Changes:

Version	Date	Created by	Short Description of Changes
1.0	13/05/2016	CEF eDelivery	Domibus 3.1.0 release
1.01	08/09/2016	CEF eDelivery	Updated links
1.02	23/06/2017	CEF eDelivery	Update operation "getMessageStatus" to "getStatus" with the introduction of new message status: "DOWNLOADED".
1.03	07/09/2017	CEF eDelivery	Update new operations names: "submitMessage" and "retrieveMessage".
1.04	20/03/2018	CEF Support	Reuse notice added
1.05	26/09/2018	Caroline AEBY Ahmed Guili	No more standby service + Multitenancy added
1.06	20/05/2019	Caroline AEBY	2018 => 2019 + approver and reviewer's role update
1.2	19/11/2020	Ahmed Ghouili Caroline AEBY	Screenshots updated 2019 => 2020
1.3	17/12/2020	Ahmed Ghouili Caroline AEBY	Domibus 4.2 FR version
1.4	11/03/2022	Ahmed Ghouili	Update "digital-building-blocks" links. Update "Approach and purpose of the document" section.
1.5	11/03/2022	Chaouki BERRAH	Review and update
1.6	29/04/2022	Caroline AEBY	No more CEF references + links updates
1.7	03/06/2022	Caroline AEBY	eDelivery support FMB update
1.8	27/02/2023	Ahmed Ghouili Caroline AEBY	Updates for Domibus 5.1 FR
1.9	18/08/2023	Caroline AEBY	Removed reference to CEF glossary

Table of Contents

1. PREREQUISITES.....	5
1.1. Install and configure two AS4 Access Points	5
1.2. Install SoapUI.....	5
1.3. Configure SoapUI.....	5
2. TEST SCENARIOS.....	8
2.1. Web Service Submission.....	8
2.1.1. Business Scenario	8
2.1.2. Attachment alternatives.....	11
2.2. JMS Submission	12
2.3. Load Test	14
3. VERIFYING THE MESSAGE STATUS IN THE ADMINISTRATION CONSOLE	15
4. MULTITENANCY.....	17
4.1. Configuration.....	17
4.2. Update the soapUI project	17
4.2.1. Web Service Submission.....	18
4.2.2. JMS Submission	20
4.3. Checking the messages in the admin console	20
4.3.1. User other than access point admin	21
4.3.2. Access point admin user.....	22
5. CONTACT INFORMATION	23

Approach and purpose of the document

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

This document is intended for people/organizations that want to perform a set of basic checks on their Domibus installation. It can be used by developers after updating Domibus. It can also be used by testers as 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 eDelivery Support team.

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

<https://ec.europa.eu/digital-building-blocks/wikis/display/DIGITAL/Master+Service+Arrangement>

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 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 free¹ via and follow the installation instructions on the website. The version used to create the tests is SoapUI 5.2.1².

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>

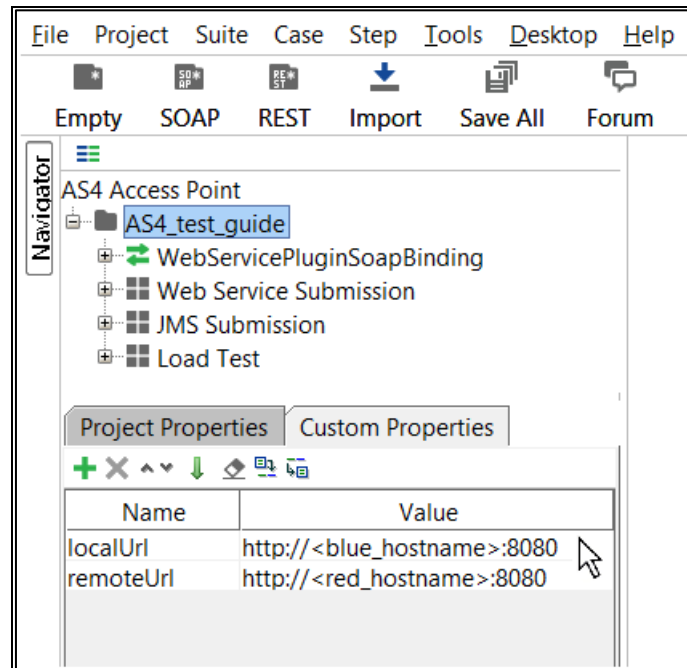


Figure 1 - Custom Properties

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/wsplugin?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.

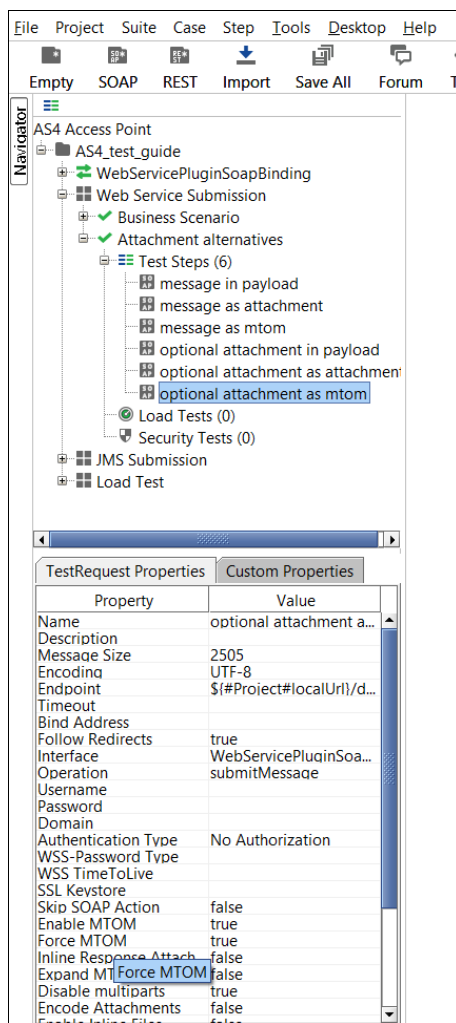


Figure 2 – Test Request properties

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 Guide³ allows users to send messages via the default Web Service plugin⁴. The Quick Start Guide includes an optional section on deploying the default JMS plugin⁵. To be able to test sending messages to the JMS plugin via the sample SoapUI project, the Apache ActiveMQ libraries⁶ 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/digital-building-blocks/wikis/display/DIGITAL/Domibus>

⁴ The Web Service Plugin ICD is available in the documentation section of the Domibus release via <https://ec.europa.eu/digital-building-blocks/wikis/display/DIGITAL/Domibus>

⁵ The JMS plugin ICD is available via is available in the documentation section of the Domibus release via <https://ec.europa.eu/digital-building-blocks/wikis/display/DIGITAL/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.

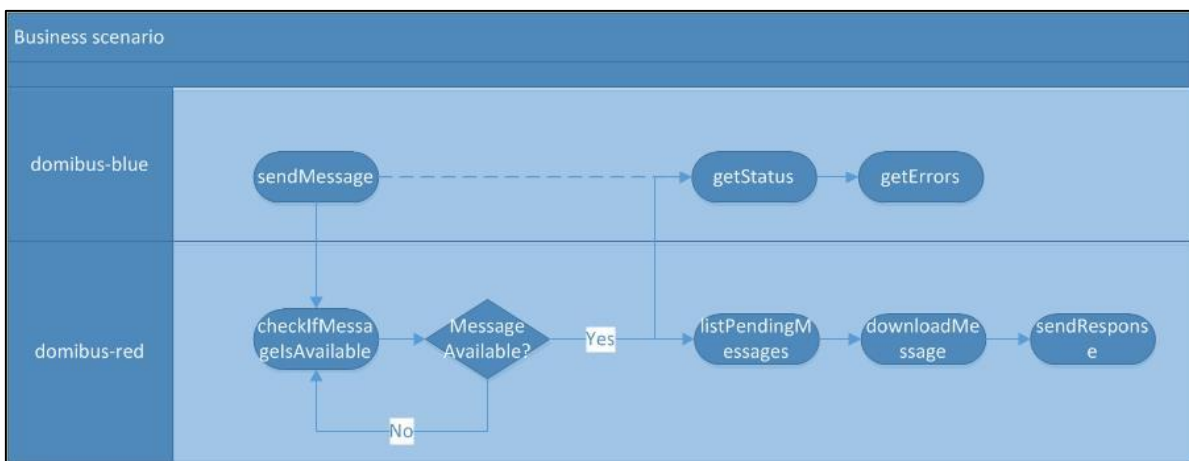


Figure 3 – Overview of the business scenario

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" and "ResponseParameters" are used for storing the MessageID parameter of the AS4 message. This parameter 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 "getStatusWithAccessPointRole" 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 "getMessageErrorsWithAccessPointRole" 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.

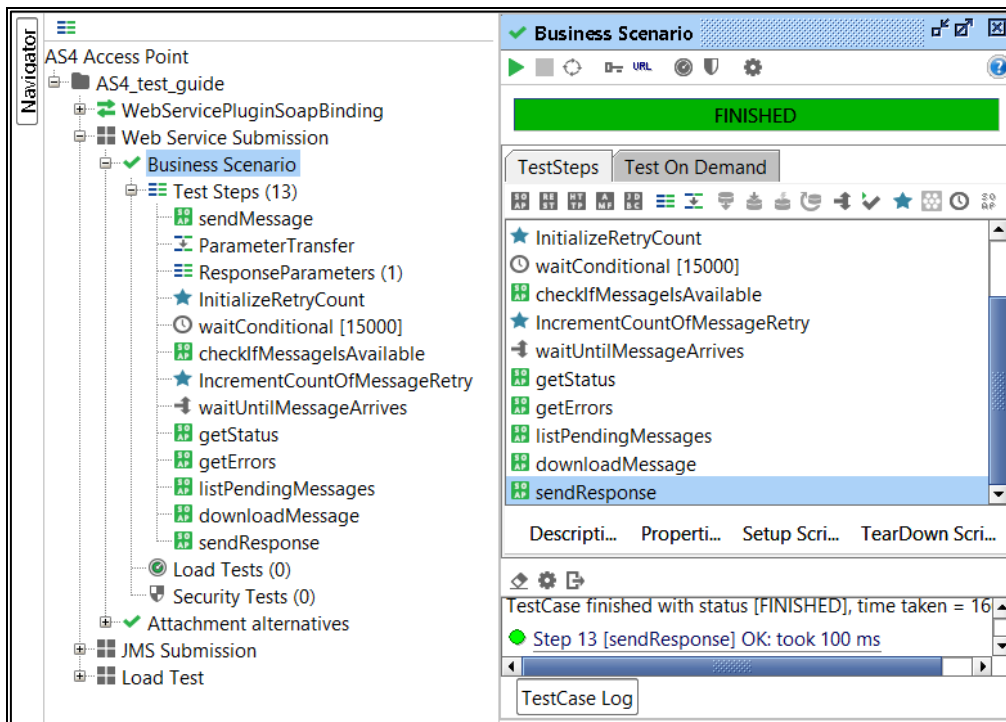


Figure 4 – Running the business scenario

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.

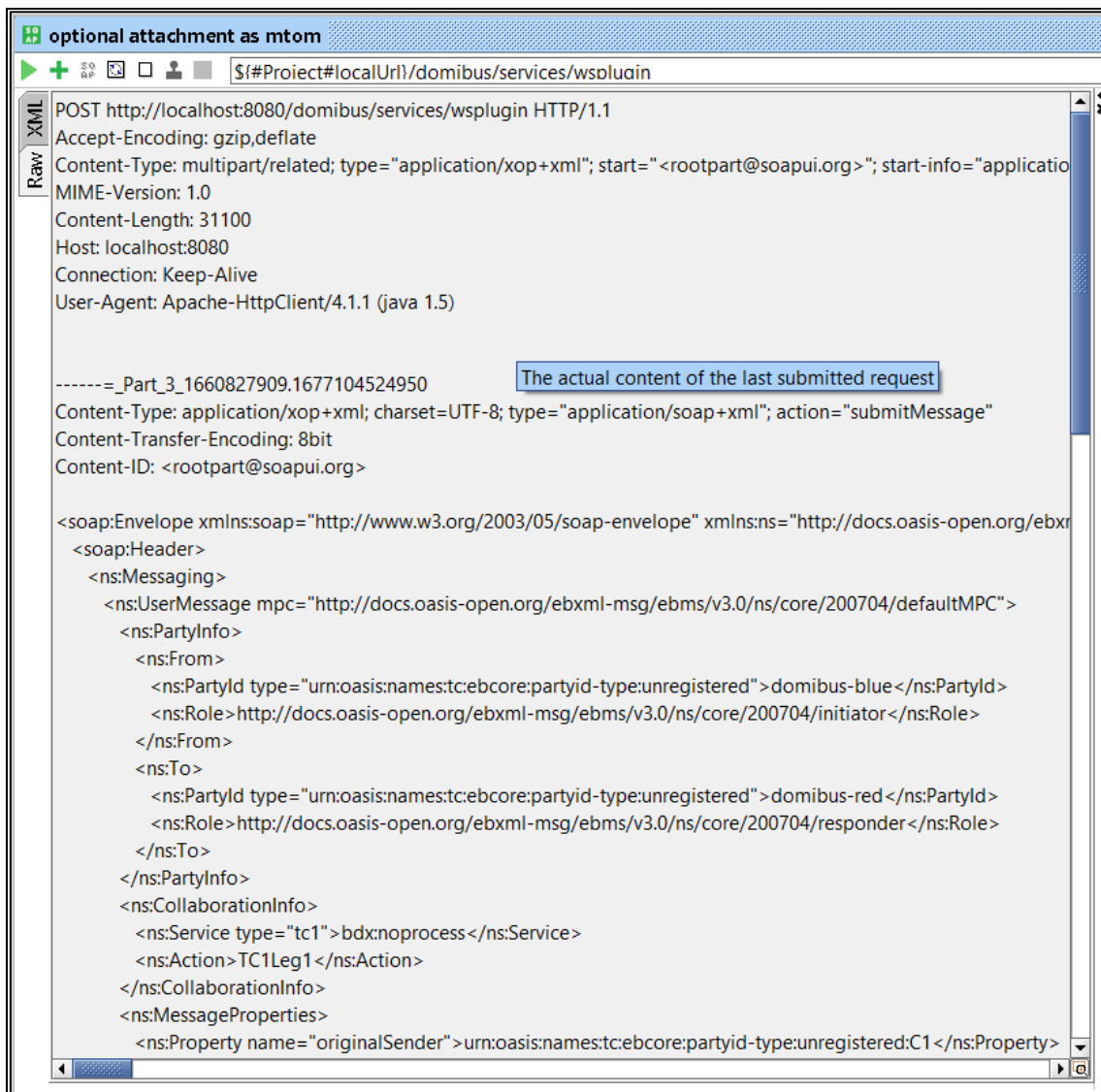


Figure 5 – Visualising the raw message that SoapUI sends.

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 based on the jms 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. Currently, this code is only valid for Tomcat server deployment. In the future, we will provide a test suite that will also support WebLogic and WildFly servers.

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).

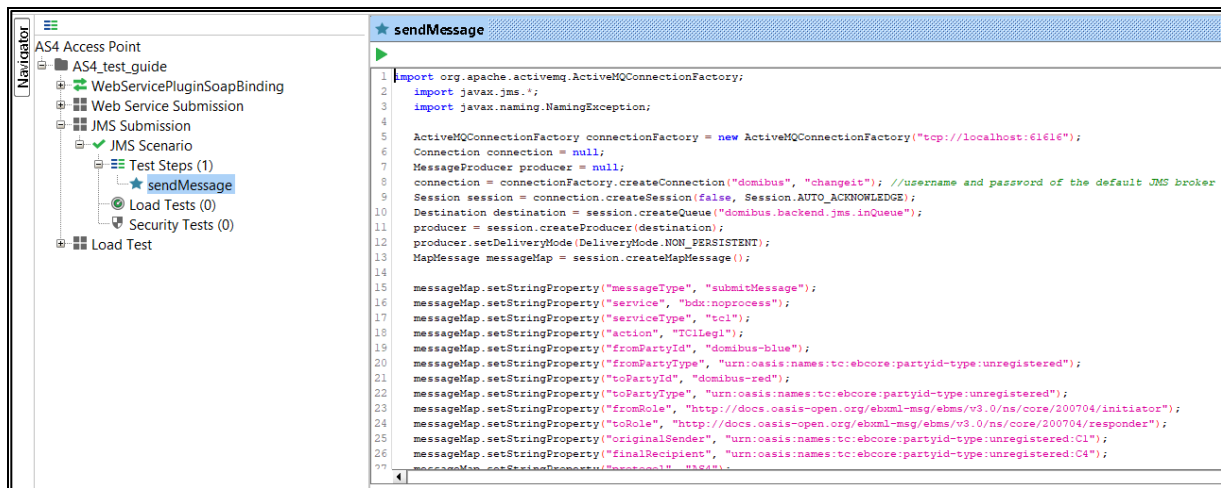


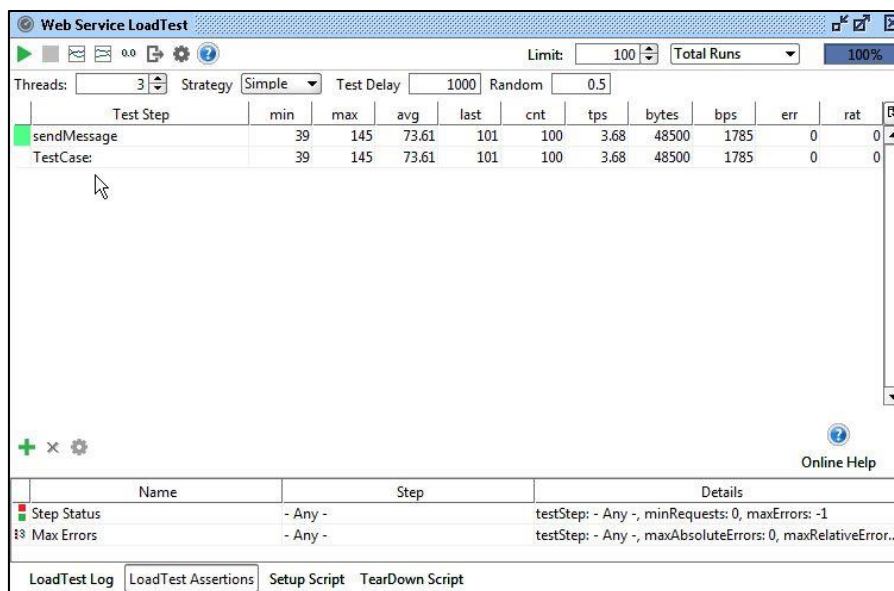
Figure 6 – Sending a JMS message

2.3. Load Test

The Web Service LoadTest will send several 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.



The screenshot shows the 'Web Service LoadTest' window. At the top, there are controls for 'Threads' (set to 3), 'Strategy' (Simple), 'Test Delay' (1000), and 'Random' (0.5). The 'Limit' is set to 100 and 'Total Runs' is 100%. Below these controls is a table with the following data:

Test Step	min	max	avg	last	cnt	tps	bytes	bps	err	rat
sendMessage	39	145	73.61	101	100	3.68	48500	1785	0	0
TestCase:	39	145	73.61	101	100	3.68	48500	1785	0	0

At the bottom of the window, there is a table with columns 'Name', 'Step', and 'Details':

Name	Step	Details
Step Status	- Any -	testStep: - Any -, minRequests: 0, maxErrors: -1
Max Errors	- Any -	testStep: - Any -, maxAbsoluteErrors: 0, maxRelativeError...

Below the table are tabs for 'LoadTest Log', 'LoadTest Assertions', 'Setup Script', and 'TearDown Script'. There are also icons for '+', 'x', and a gear, and an 'Online Help' link.

Figure 7 – Executing a load test

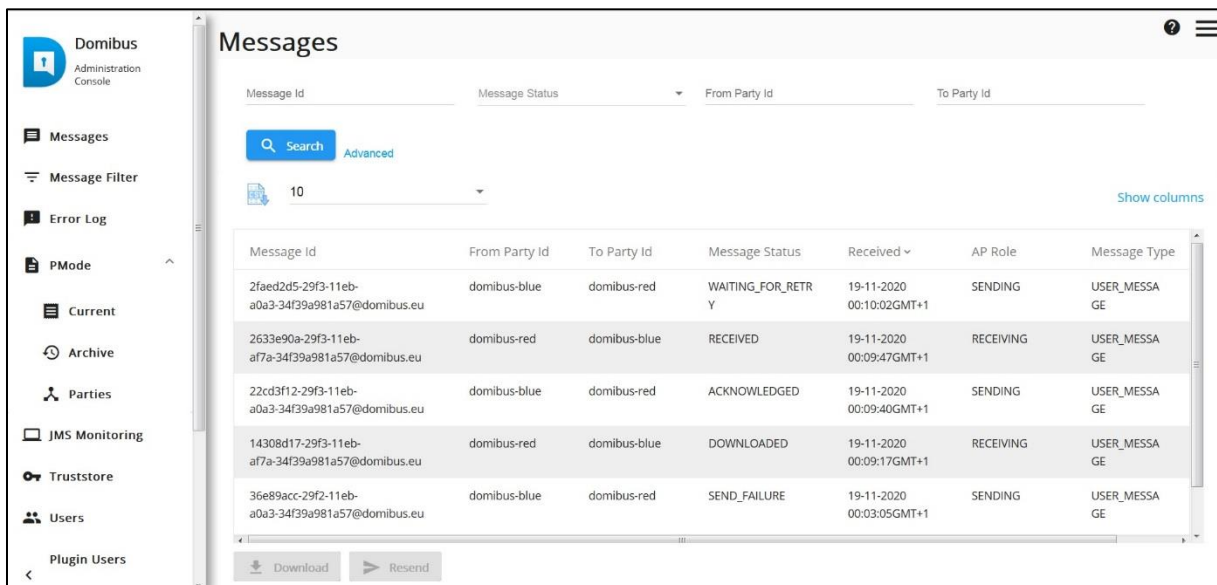
3. VERIFYING THE MESSAGE STATUS IN THE ADMINISTRATION CONSOLE

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 "22cd3f12-29f3-11eb-a0a3-34f39a981a57@domibus.eu" has the status "ACKNOWLEDGED" on the sending Access Point. This indicates that the message has been successfully received and acknowledged from the receiving Access Point.

As another example, the same figure below shows that the message with ID "2633e90a-29f3-11eb-af7a-34f39a981a57@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 "14308d17-29f3-11eb-af7a-34f39a981a57@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).



The screenshot shows the Domibus Administration Console interface. The main content area is titled "Messages" and displays a table with the following columns: Message Id, From Party Id, To Party Id, Message Status, Received, AP Role, and Message Type. The table contains six rows of message data.

Message Id	From Party Id	To Party Id	Message Status	Received	AP Role	Message Type
2faed2d5-29f3-11eb-a0a3-34f39a981a57@domibus.eu	domibus-blue	domibus-red	WAITING_FOR_RETRY	19-11-2020 00:10:02GMT+1	SENDING	USER_MESSAGE
2633e90a-29f3-11eb-af7a-34f39a981a57@domibus.eu	domibus-red	domibus-blue	RECEIVED	19-11-2020 00:09:47GMT+1	RECEIVING	USER_MESSAGE
22cd3f12-29f3-11eb-a0a3-34f39a981a57@domibus.eu	domibus-blue	domibus-red	ACKNOWLEDGED	19-11-2020 00:09:40GMT+1	SENDING	USER_MESSAGE
14308d17-29f3-11eb-af7a-34f39a981a57@domibus.eu	domibus-red	domibus-blue	DOWNLOADED	19-11-2020 00:09:17GMT+1	RECEIVING	USER_MESSAGE
36e89acc-29f2-11eb-a0a3-34f39a981a57@domibus.eu	domibus-blue	domibus-red	SEND_FAILURE	19-11-2020 00:03:05GMT+1	SENDING	USER_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 "36e89acc-29f2-11eb-a0a3-34f39a981a57@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 "2faed2d5-29f3-11eb-a0a3-34f39a981a57@domibus.eu" has the status "WAITING_FOR_RETRY" on the sending Access Point.

4. MULTITENANCY

Since release 4.0, Domibus supports multitenancy: on each access point, multiple domains can be configured with each domain having its own set of configuration files (Keystore, Truststore, PMode, Domain properties, etc.). This allows message exchange to be extended to domain level.

4.1. Configuration

By following the instructions in the admin guide⁷, multitenancy can be configured on the previously installed access points “blue” and “red”.

4.2. Update the soapUI project

Assuming the following configuration (instructions in the admin guide):

- Domains “blue_Domain1” and “blue_Domain2” are configured on “blue” access point.
- “blue_user1” is linked to domain “blue_Domain1” and “blue_user2” is linked to domain “blue_Domain2”.
- Domains “red_Domain1” and “red_Domain2” are configured on “red” access point.
- “red_user1” is linked to domain “red_Domain1” and “red_user2” is linked to domain “red_Domain2”.

Since plugin users are defined for each domain, authentication needs to be added on each request.

⁷ The Admin guide is available in the documentation section of the Domibus release via <https://ec.europa.eu/digital-building-blocks/wikis/display/DIGITAL/Domibus>

4.2.1. Web Service Submission

For webservice submission, authentication needs to be added on each soap request test step.

First select “Add New Authorization” in the Authorization tab:

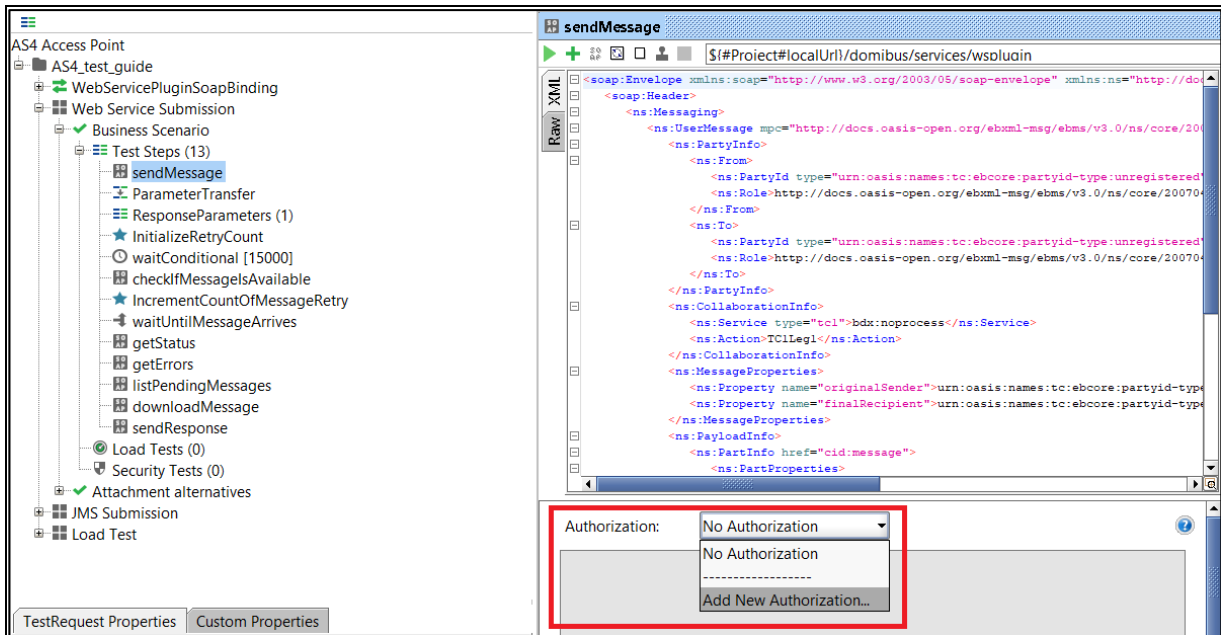


Figure 9 – Adding new authorization.

Then, select “Basic” authorization value:

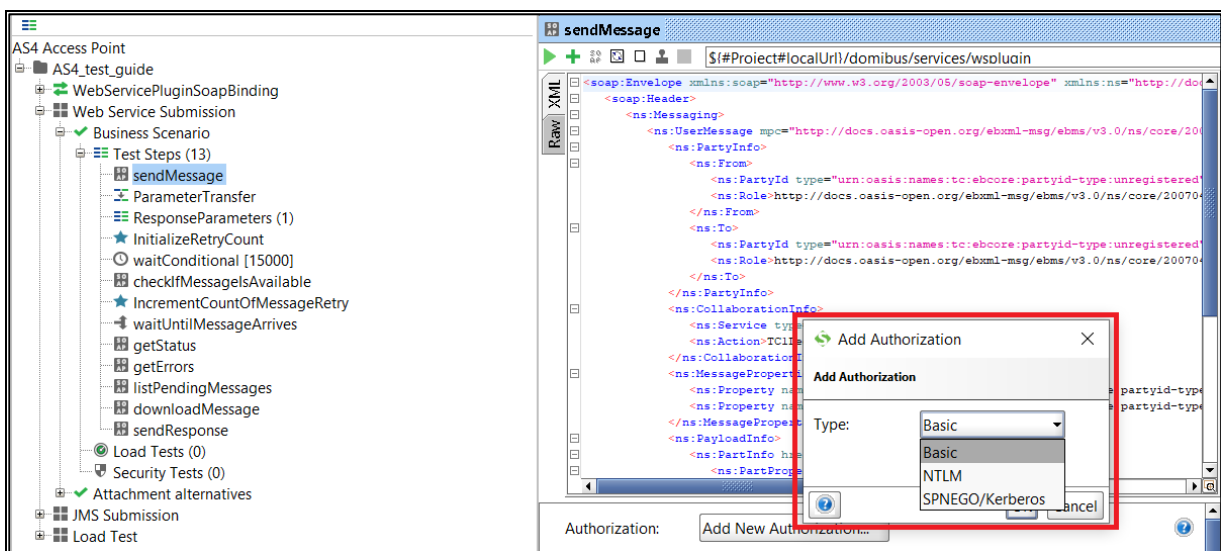


Figure 10 – Selecting basic authorization.

Finally, type the user/password linked to the targeted domain (sender side) and select “Authenticate pre-emptively” option:

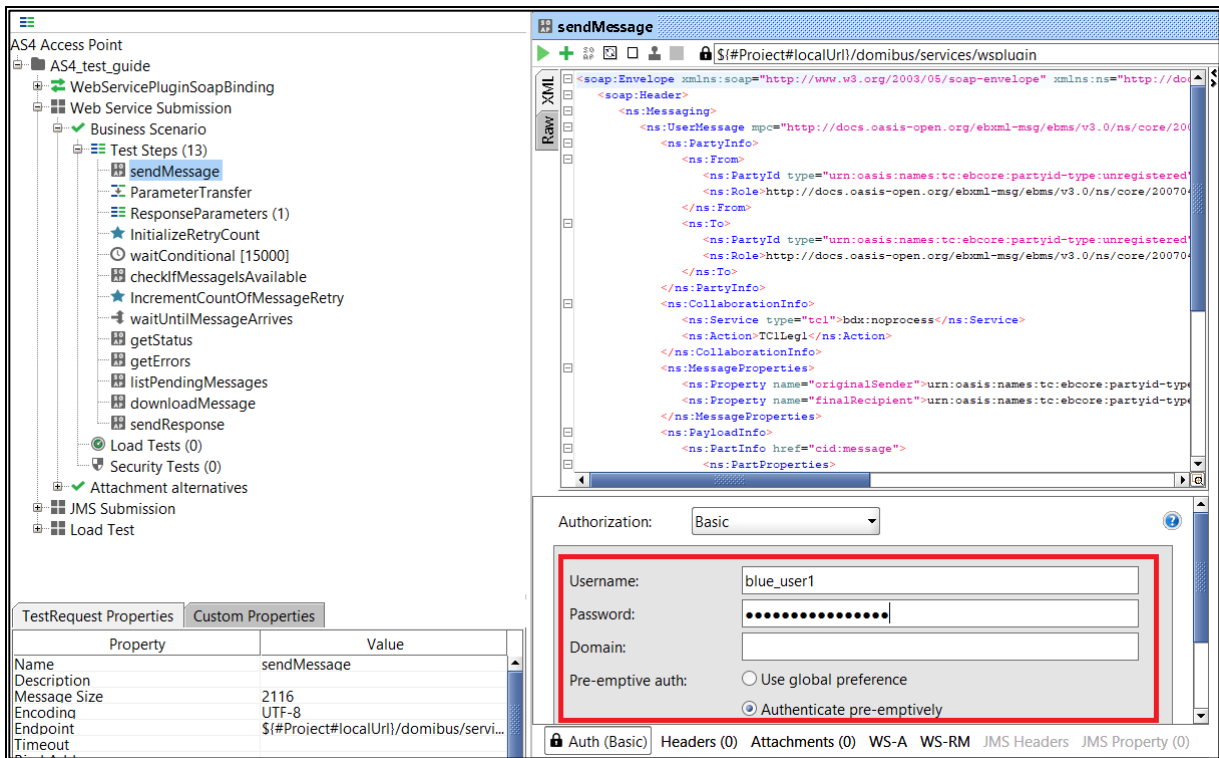


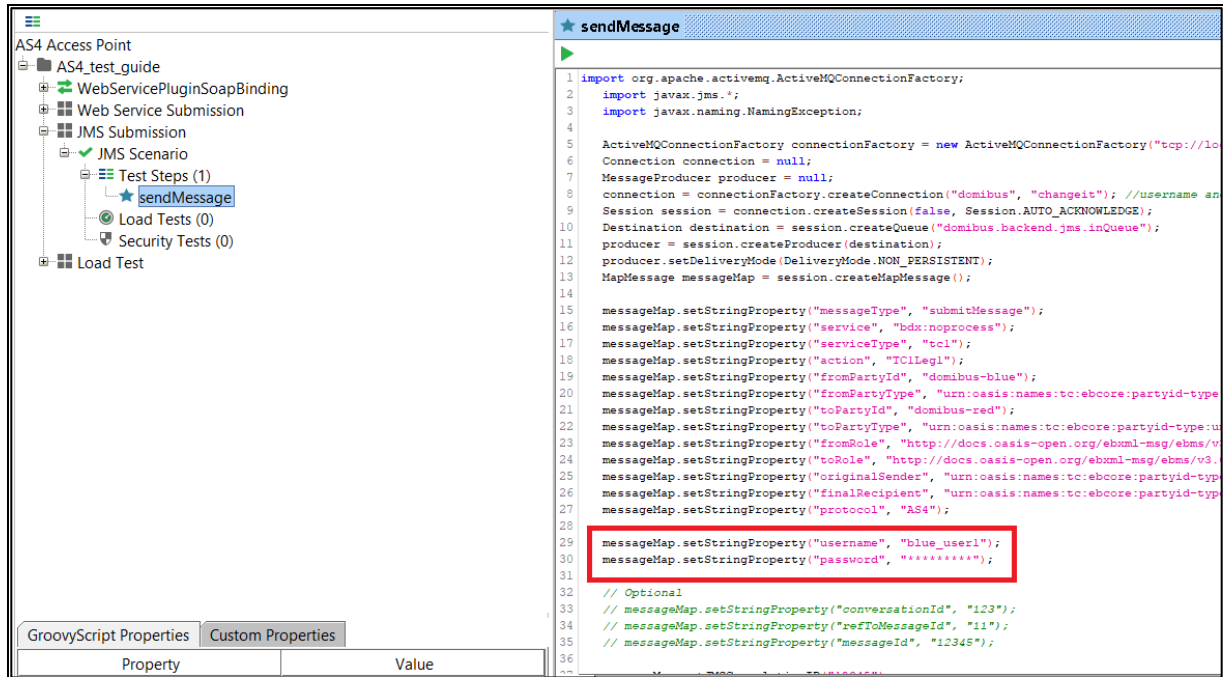
Figure 11 – Authorization credentials.

In this case, by authenticating with plugin user “blue_user1”, the domain “blue_domain1” will be selected as sender of the message from the “blue” access point.

Depending on the PMode configuration (see admin guide), the message will be received either by “red_domain1” or “red_domain2” from the “red” access point.

4.2.2. JMS Submission

For jms submission, authentication needs to be added via extra properties in the groovy test step:



The screenshot shows a test runner interface with a tree view on the left and a code editor on the right. The tree view shows a test scenario named 'JMS Scenario' with a test step named 'sendMessage'. The code editor shows a Groovy script for sending a message. The script sets various properties for a message, including 'username' and 'password' which are highlighted in a red box.

```
1 import org.apache.activemq.ActiveMQConnectionFactory;
2 import javax.jms.*;
3 import javax.naming.NamingException;
4
5 ActiveMQConnectionFactory connectionFactory = new ActiveMQConnectionFactory("tcp://lo
6 Connection connection = null;
7 MessageProducer producer = null;
8 connection = connectionFactory.createConnection("domibus", "changeit"); //username an
9 Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
10 Destination destination = session.createQueue("domibus.backend.jms.inQueue");
11 producer = session.createProducer(destination);
12 producer.setDeliveryMode(DeliveryMode.NON_PERSISTENT);
13 MapMessage messageMap = session.createMapMessage();
14
15 messageMap.setStringProperty("messageType", "submitMessage");
16 messageMap.setStringProperty("service", "bdx:noprocess");
17 messageMap.setStringProperty("serviceType", "tcl");
18 messageMap.setStringProperty("action", "TCLLeg1");
19 messageMap.setStringProperty("fromPartyId", "domibus-blue");
20 messageMap.setStringProperty("fromPartyType", "urn:oasis:names:tc:ebcore:partyid-type
21 messageMap.setStringProperty("toPartyId", "domibus-red");
22 messageMap.setStringProperty("toPartyType", "urn:oasis:names:tc:ebcore:partyid-type:u
23 messageMap.setStringProperty("fromRole", "http://docs.oasis-open.org/ebxml-msg/ebms/v
24 messageMap.setStringProperty("toRole", "http://docs.oasis-open.org/ebxml-msg/ebms/v3.
25 messageMap.setStringProperty("originalSender", "urn:oasis:names:tc:ebcore:partyid-typ
26 messageMap.setStringProperty("finalRecipient", "urn:oasis:names:tc:ebcore:partyid-typ
27 messageMap.setStringProperty("protocol", "AS4");
28
29 messageMap.setStringProperty("username", "blue_user1");
30 messageMap.setStringProperty("password", "*****");
31
32 // Optional
33 // messageMap.setStringProperty("conversationId", "123");
34 // messageMap.setStringProperty("refToMessageId", "11");
35 // messageMap.setStringProperty("messageId", "12345");
36
```

Figure 12 – Setting user authentication for JMS.

4.3. Checking the messages in the admin console

Messages sent/received for each domain can only be viewed in the admin console by users linked to the targeted domain (or by the admin of the access point).

4.3.1. User other than access point admin

When connected to the admin console with a standard user (not an access point admin user), the domain linked to that user is displayed in the “messages” page. Only messages sent/received for that domain are displayed:



Figure 13 – Admin console for normal user.

4.3.2. Access point admin user

When connected to the admin console with access point admin user, all messages of all domains can be viewed. The messages of each domain can be displayed by toggling between domains via the selection on the top right corner of the admin console:



Figure 14 – Admin console for AP admin.

5. CONTACT INFORMATION

eDelivery Support Team

By email: EC-EDELIVERY-SUPPORT@ec.europa.eu

Support Service: 8am to 6pm (Normal EC working Days)