**Test Guide**

**e-Delivery pilot for BRIS**

**Document history**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Comment | Modified pages |
| 1.00 | 13/05/2015 | Creation for BRIS demo | All pages |
|  |  |  |  |

CONTENTS

[1. Overview 3](#_Toc419127129)

[2. Prerequisites 3](#_Toc419127130)

[2.1. Complete the BRIS quick start guide 3](#_Toc419127131)

[2.2. Install SoapUI 3](#_Toc419127132)

[2.3. Configure SoapUI 4](#_Toc419127133)

[3. Test Scenarios 7](#_Toc419127134)

[3.1. Submit BRIS Document via web-services 7](#_Toc419127135)

# Overview

This document contains information to set up the test and demo framework for the BRIS pilot.

First the test environment will be set up and the test tool will be configured.

Then the different test scenarios are explained and guidance is provided on how to run the existing tests.

# Prerequisites

## Complete the BRIS quick start guide

Ensure that you have a working environment by following the steps defined in the "Quick start guide BRIS".

## 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 via <http://www.soapui.org/> and follow the installation instructions on the website. (The version used to create the tests can be downloaded via <http://sourceforge.net/projects/soapui/files/soapui/5.1.3/SoapUI-x64-5.1.3.exe/download>)

When running the installer, ensure that the HermesJMS option is checked as this functionality is needed the tests that interface with the JMS queues.

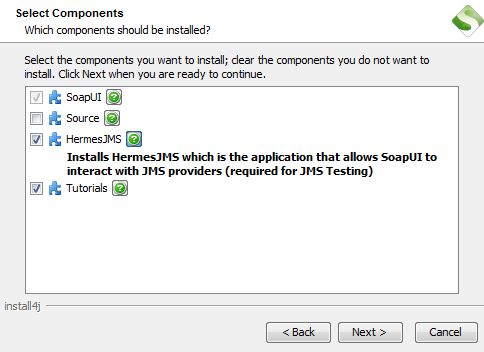


Figure 1 - HermesJMS option

To be able to submit JMS messages via a groovy script in the test cases, you need to add the file jboss-client.jar (present in the test package) to the SoapUI installation folder in bin\ext (e.g. "C:\Program Files\SmartBear\SoapUI-5.1.3\bin\ext") and restart SoapUI.

## Configure SoapUI

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

In the Navigator in SoapUI, left-click on the project "BRIS\_test\_guide" and select "Custom Properties" in the Properties panel.

* Set the Property named "local\_url" to the IP address and port of the machine that is running the sending BRIS connector. If this is your local machine, then the value can remain "localhost".
* Set the Property named "remote\_url" to the IP address and port of the machine that is running the receiving BRIS connector (most likely this is a virtualbox running on your machine or a remote machine). If it is the virtualbox running on your machine, the value can remain "http://192.168.56.11:8080".
* Set the Property named "URL" to the JMS interface of machine that is running the sending BRIS connector. If this is your local machine, then the value can remain "remote://localhost:4447".
* Set the Property named "ForwardURL" to the JMS interface of machine that is running the receiving BRIS connector (most likely this is a virtualbox running on your machine or a remote machine). If it is the virtualbox running on your machine, the value can remain "remote://192.168.56.11:4447".

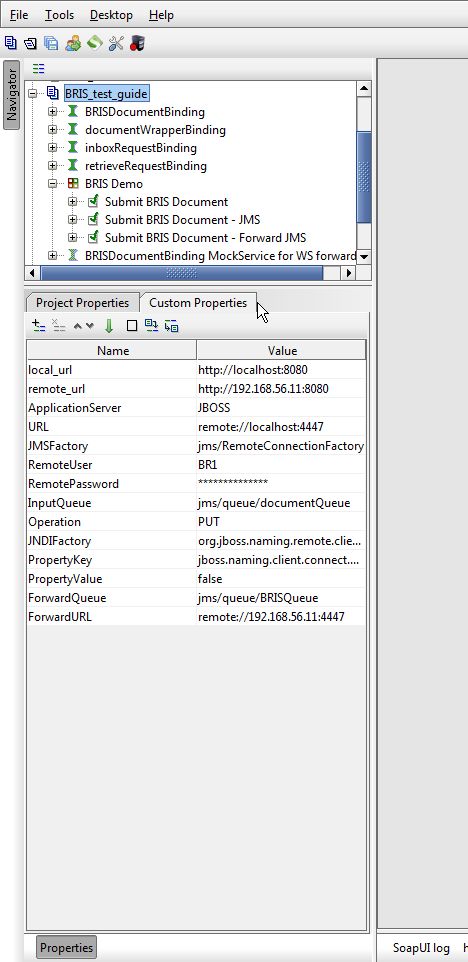


Figure 2 - Custom Properties

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

The WSDLs of the BRIS services were loaded when creating the test project and they are stored in the project file. If you want to reload them, ensure that your local connector is up and running. Otherwise the URLS of the WSDLs (e.g. <http://localhost:8080/connector/wsdl/BRISDocument-2.0.wsdl>) will not be accessible.

MTOM (Message Transmission Optimization Mechanism) is a method of efficiently sending 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.

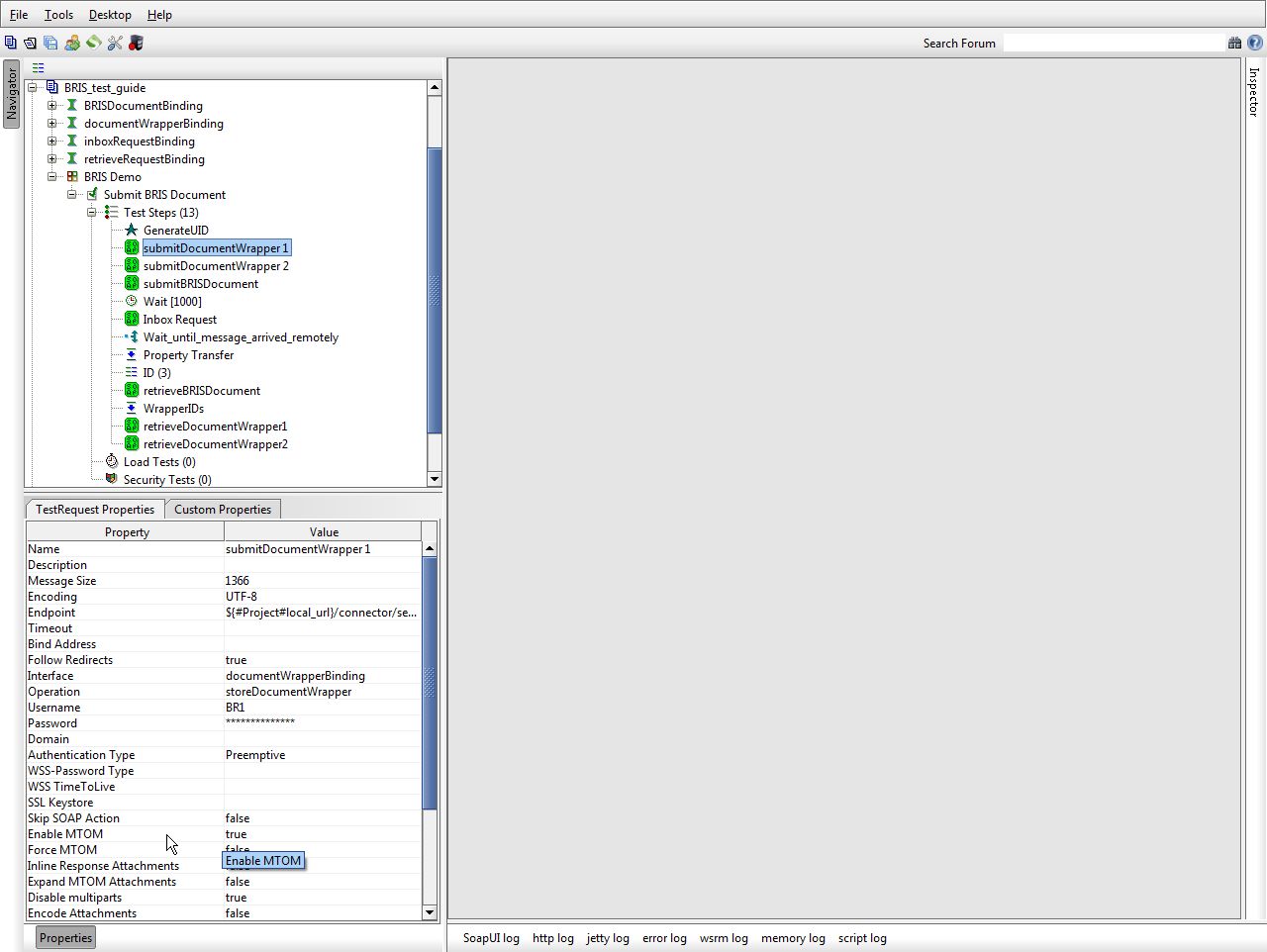


Figure 3 - TestRequest properties

The usernames and passwords for the Parties are already pre-configured with the following values:

PARTY\_BR1: username "BR1", password "etrustex".

PARTY\_ECP: username "ECP", password "etrustex".

Note that pre-emptive authentication is already configured in the test request, so no action is required. With pre-emptive authentication, you send the basic authentication information immediately, even before the server gives an unauthorized response.

# Test Scenarios

## Submit BRIS Document via web-services

In this scenario, the party BR1 will send messages to the party ECP via the web-service interface of the connector.

ECP will poll for new messages and retrieve them using the web-service interface of the connector.

The detailed steps are listed below:

1. A unique ID is generated in GenerateUID. This will ensure that you can run the test case multiple times without changing values or parameters throughout the requests. If the same ID would be sent in different subsequent request, an error would be generated by the local connector.
2. A binary file is submitted in submitDocumentWrapper 1 and submitDocumentWrapper 2. The binary files are different JPG files with a flag of Europe. These binary files are not yet sent to the recipient, but are uploaded to the local connector so they can be referenced later by a BRIS Document.
3. A BRIS document is submitted. This BRIS document refers to the two Document Wrappers submitted in step 2 and additionally, contains a third binary attachment in the "SpecificContent" element. This is a base64 encoded JPG with a map of Europe. When this BRIS Document is submitted to the local connector, it will build a message that contains the BRIS Document and the two referenced Document Wrappers and send it to the recipient on the receiving connector (ECP).
4. The recipient polls if a new message is available by performing an Inbox Request. The Inbox Response contains the details of the messages that are sent to the recipient and that are not yet marked as retrieved.
5. The recipient in this test scenario is configured to keep on polling for a message with the ID that is submitted in step 3. Once this message is available, it will use the Inbox Response details to call the Retrieve service of the receiving connector.
6. The Retrieve service response will contain the BRIS document that was sent from the sender (BR1) to the receiver (BR2). It contains the base64 encoded attachment and the references to the 2 Document Wrappers.
7. Using the two Document Wrapper IDs extracted from the received BRIS Document, the recipient can call the Retrieve Document Wrapper service of the receiving connector to retrieve the contents of the Document Wrappers themselves, including the binary files.

To run this test scenario, open the "Submit BRIS Document" 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.

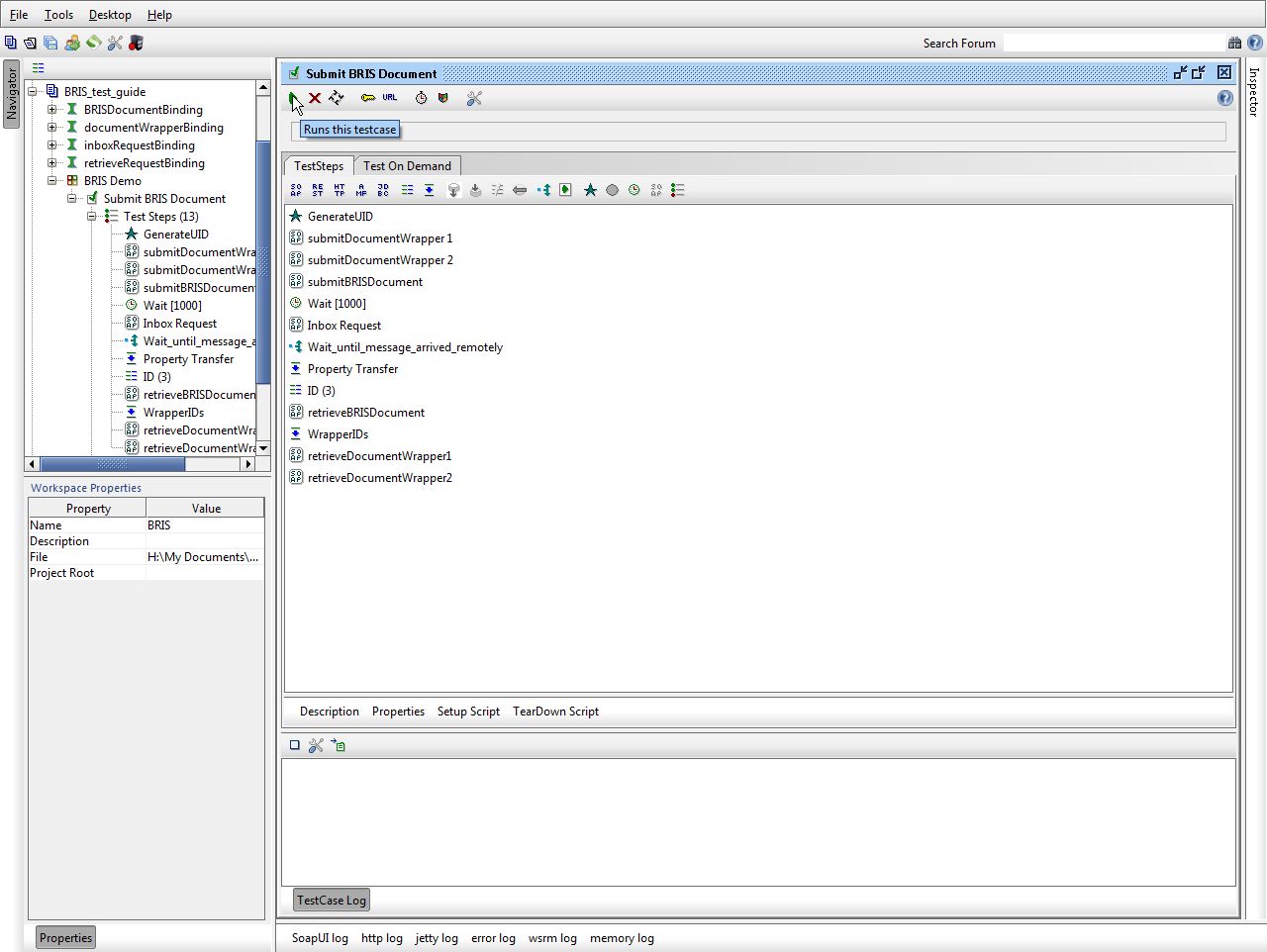


Figure 4 - Run test scenario

To verify that the binary attachments are correctly retrieved by ECP, you can click on the