Message Test Harness GPES Scenarios Guide

**Background**

The exception scenarios are executed using Message Test Harness tool which is used to simulate Spine. It programmatically responds to a supplier’s systems messaging with a predefined message based on incoming parameters. It is typically used to demonstrate/test a systems’ response to error conditions which cannot be recreated in an environment, because other users would be adversely affected.

**Overview**

There are 2 versions of the GPES rulesets:

1. GPES\_Ruleset.txt

This ruleset is used for all the scenarios and can be executed in a windows environment using MTHResponder\_GPES.bat

1. GPES\_Ruleset\_retrycatch.txt

This ruleset is used for scenarios which test retry behaviour but then require the demonstration of a subsequent retry being responded to.

The MTH Responder requires that there is a value within the incoming message which can be keyed upon in order to differentiate which of the responses is returned. The value that has been used is:

* Scheduled-Query-Instance-ID for COMT\_IN000001GB01 and COMT\_IN000002GB01 messages
* Scheduled-Query-Identifier-ID for COMT\_IN000003GB01 and COMT\_IN000004GB01 messages

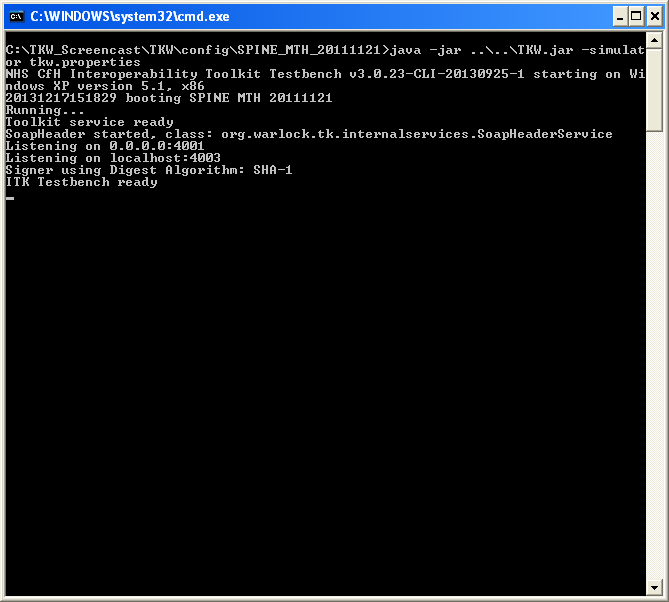
MTH Quick Start:

MTH Client – GPES MTH Responder Configuration

1. Using the latest version of TKWInstaller.jar from <http://systems.hscic.gov.uk/sa/tools> download and install the Toolkit Workbench
2. Start the simulator by navigating to the location TKW\config\SPINE\_MTH\_20111121 using a command line and executing the following:

java -jar ..\..\TKW.jar -simulator tkw.properties

to commence listening for incoming messages.



1. Execute the batch file \TKW\contrib\Additional\_Tools\TKW\_MTH\_Console\Console.bat to start the Client message engine. We use this program to send example messages to MTH locally using default port values which match supplied configuration files. This proves that the MTH is working properly before we attempt to send any messages into MTH from a supplier’s system. (It is also used to switch rulesets on the MTH after the System under test has entered retry mode so that MTH can then respond with an ebXML acknowledgement – see later).
2. Using the Console File>Open navigate to the example message folder: \TKW\contrib\SPINE\_Test\_Messages\MTH\_Test\_Messages\GPES\_Example\_input\_Msg
3. Select the first message and using the send button send this message to the listening MTH. (The sending and listening ports of the client can be checked/changed using the Controls tab)
4. Repeat for all the example messages, ensuring that the appropriate responses are returned to the MTH Client application in the synchronous and asynchronous response windows. There is one example message which corresponds to each of the Compliance exception scenarios for which MTH is applicable. The message examples are named accordingly.

Supplier Client – GPES MTH Responder Configuration

1. Now that the MTH Client and GPES MTH Responder are confirmed to be working together correctly, we need to substitute the supplier system for the MTH Client.
2. Open t\TKW\config\SPINE\_MTH\_20111121\tkw.properties and change any of the properties according to your system location. These values define how MTH Responder acts. Within this file tks.Toolkit.listenaddr and tks.Toolkit.listenport need to be changed to the port where MTH should expect the incoming message and to send the response message to.
3. Open \TKW\config\SPINE\_MTH\_20111121\simulator\_config\siab-test-sds-ref.xml. This is the store of SDS values for the exchanged messages
4. For each of the incoming interactions used in the MTH (COMT\_IN000001GB01, COMT\_IN000002GB01, COMT\_IN000003GB01, COMT\_IN000004GB01), update the supplier system to use the values outlined here. If any of these values is needs changing, make sure this is reflected in siab-test-sds-ref.xml.
5. For each of the outgoing interactions from the MTH (MCCI\_IN010000UK13) ensure that all the values in the entry in siab-test-sds-ref.xml are correct wrt to the supplier’s system values.
6. Using the supplier’s system send messages to MTH ensuring that the outgoing message will contain the appropriate key value as per the following table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| GPES Exception Type | Example Message Name | Key Element | Element Value | MTH Ruleset to use | Expected response |
| GPET-Q | COMT\_IN000001GB01\_01Runtime Parameters message Retry Behaviour\_NoEbxmlAck.xml | Scheduled-Query-Instance-ID | A11111-000000000001-R | testmthconfig\_MTHGPESTestScenarios.xml | No ebXML Acknowledgement |
| GPET-Q | COMT\_IN000003GB01\_02SPINE returns ebXML errors to GPET-Q MHS\_ebXMLError.xml | Scheduled-Query-Identifier-ID | A22222-000000000001-R | testmthconfig\_MTHGPESTestScenarios.xml | ebXML Error response |
| GPET-Q | COMT\_IN000001GB01\_03RTP message retries terminate on receipt of successful ebXML Acknowledgement from SPINE.xml | Scheduled-Query-Instance-ID | A33333-000000000001-R | testmthconfig\_MTHGPESTestScenarios.xml then testmthconfig\_retrycatch\_MTHGPESTestScenarios.xml | An HTTP 202 Accepted followed by an ebXML acknowledgement |
| GPET-Q | COMT\_IN000001GB01\_06RTP message - Negative MCCI received from SPINE TMS\_NACK.xml | Scheduled-Query-Instance-ID | A66666-000000000001-R | testmthconfig\_MTHGPESTestScenarios.xml | Asynchronous negative MCCI |
| GPET-Q | COMT\_IN000001GB01\_07SPINE Returns a HTTP 500 SoapFault to GPET-Q MHS\_SOAPFault.xml | Scheduled-Query-Instance-ID | A77777-000000000001-R | testmthconfig\_MTHGPESTestScenarios.xml | HTTP 500 SOAP Fault |
| GPET-E | COMT\_IN000002GB01\_ 01Runtime Parameters Response message Retry Behaviour\_TMS\_NoEbxmlAck.xml | Scheduled-Query-Instance-ID | A11111-000000000001-R | testmthconfig\_MTHGPESTestScenarios.xml | No ebXML Acknowledgement |
| GPET-E | COMT\_IN000004GB01\_02SPINE returns ebXML errors to GPET-E MHS\_ebXMLError.xml | Scheduled-Query-Identifier-ID | A22222-000000000001-R | testmthconfig\_MTHGPESTestScenarios.xml | ebXML Error response |
| GPET-E | COMT\_IN000002GB01\_03RTPR message retries terminate on receipt of successful ebXML Acknowledgement.xml | Scheduled-Query-Instance-ID | A33333-000000000001-R | testmthconfig\_MTHGPESTestScenarios.xml then testmthconfig\_retrycatch\_MTHGPESTestScenarios.xml | An HTTP 202 Accepted followed by an ebXML acknowledgement |
| GPET-E | COMT\_IN000002GB01\_06RTPR message - Negative MCCI received from SPINE TMS\_NACK.xml | Scheduled-Query-Instance-ID | A66666-000000000001-R | testmthconfig\_MTHGPESTestScenarios.xml | Asynchronous negative MCCI |
| GPET-E | COMT\_IN000002GB01\_07SPINE Returns a HTTP 500 SoapFault to GPET-E MHS\_SOAPFault.xml | Scheduled-Query-Instance-ID | A77777-000000000001-R | testmthconfig\_MTHGPESTestScenarios.xml | HTTP 500 SOAP Fault |

Notes:

* It may be easier to change the key values in the MTH rulesets to match those which will be in the messages sent to MTH responder
* For the test 04 - RTP message retries terminate on expiry of the Persist Duration use the message value for 01 - Runtime Parameters Response message Retry Behaviour\_TMS\_NoEbxmlAck\_010301, i.e. A11111-000000000001-R

ebXML retry Catching

There are 2 tests which require the demonstration of the retry and subsequent catch of an ebXML acknowledgement: RTP/RTPR message retries terminate on receipt of successful ebXML Acknowledgement from SPINE.

These two tests are performed this way:

1. Start the MTH using the GPES\_Ruleset.txt
2. From the System Under Test, transmit the initiating message to GPES MTH (COMT\_IN000002GB01\_03RTPR message retries terminate on receipt of successful ebXML Acknowledgement.xml or COMT\_IN000001GB01\_03RTP message retries terminate on receipt of successful ebXML Acknowledgement from SPINE.xml as appropriate)
3. GPES MTH will respond with “HTTP/1.1 202 Accepted” only and the System Under Test should enter retry mode as per its contract properties
4. Before the second retry is attempted by SUT:
   1. Select the Controls tab on the already opened MTH console and click “set new rules and responses”
   2. Select from the retrycatch ruleset: GPES\_Ruleset\_retrycatch.txt
5. When the 2nd retry is attempted by the System Under Test GPES MTH will respond with an ebXML acknowledgement at which point the SUT should stop retrying