|  |  |  |  |
| --- | --- | --- | --- |
| Document filename: A Guide to the Toolkit Workbench Autotest Manager | | | |
| Directorate / Programme | HSCIC | Project | Interoperability Toolkit |
| Document Reference | | NPFIT-ELIBR-AREL-DST-0460.01 | |
| Project Manager | Rob Shaw | Status | Approved |
| Owner | Richard Robinson | Version | V2.1 |
| Author | Richard Robinson | Version issue date | 26th May 2017 |

A Guide to the Toolkit Workbench Autotest Manager

## Revision History

|  |  |  |
| --- | --- | --- |
| Version | Date | Summary of Changes |
| 0.1 |  | First draft for comment |
| 1.0 |  | Released Version |
| 1.1 |  | Update including CDA document guide |
| 1.2 |  | Update multiplatform advice |
| 1.3 |  | Rebranding for HSCIC |
| 2.0 | 24/09/2013 | Update of guide after software upgrade |
| 2.1 | 26/05/2017 | Corrected path on page 9 |

## Reviewers

This document must be reviewed by the following people: author to indicate reviewers

|  |  |  |  |
| --- | --- | --- | --- |
| Reviewer name | Title / Responsibility | Date | Version |
| Simon Farrow | SOLUTION ASSURANCE |  |  |

## Approved by

This document must be approved by the following people: author to indicate approvers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Signature | Title | Date | Version |
| Richard Dobson |  | SOLUTION ASSURANCE |  |  |
|  |  |  |  |  |

## Glossary of Terms

|  |  |
| --- | --- |
| Term / Abbreviation | What it stands for |
| ATM | Autotest Manager |
| SUT | System Under Test |

## Related Documents

These documents will provide additional information.

|  |  |  |  |
| --- | --- | --- | --- |
| Ref no | Doc Reference Number | Title | Version |
| 1 | NPFIT-SHR-QMS-PRP-0015 | Glossary of Terms Consolidated.doc | 13 |
| 2 | NPFIT-ELIBR-AREL-DST-0431.05 | A Guide to Toolkit Workbench | 3.3 |
| 3 | NPFIT-ELIBR-AREL-DST-0444 | Supplier Certified Requirements Coverage v5.8xls | 5.8 |
| 4 | NPFIT-ELIBR-AREL-DST-0431.06 | A Guide to the Toolkit Workbench | V3.4 |

Document Control:

The controlled copy of this document is maintained in the HSCIC corporate network. Any copies of this document held outside of that area, in whatever format (e.g. paper, email attachment), are considered to have passed out of control and should be checked for currency and validity.

# Contents

[1.1 Revision History 2](#_Toc367976379)

[1.2 Reviewers 2](#_Toc367976380)

[1.3 Approved by 2](#_Toc367976381)

[1.4 Glossary of Terms 2](#_Toc367976382)

[1.5 Related Documents 2](#_Toc367976383)

[2 Contents 4](#_Toc367976384)

[3 About this Document 5](#_Toc367976385)

[3.1 Purpose 5](#_Toc367976386)

[3.2 Audience 5](#_Toc367976387)

[3.3 Content 5](#_Toc367976388)

[3.4 Caveat 5](#_Toc367976389)

[4 Introduction 6](#_Toc367976390)

[4.1 Installation 8](#_Toc367976391)

[4.2 Directory Structure 9](#_Toc367976392)

[5 Using the TKW Autotest Manager 9](#_Toc367976393)

[5.1 Configuration Tab 9](#_Toc367976394)

[5.2 Run Test Tab 11](#_Toc367976395)

[5.3 Single Run Tab 20](#_Toc367976396)

# About this Document

## Purpose

This document describes the use and configuration of the TKW Autotest Manager tool.

## Audience

The guide is written for all users and interested parties – managers, analysts, architects, development and test staff. It is also intended for HSCIC staff involved in the definition of ITK service standards, message validation and test support for messaging behaviours.

## Content

The TKW Autotest software is distributed as part of the Toolkit Workbench and includes software to execute predefined suites of tests against supplier’s systems acting in a host or client mode.

## Caveat

TKW Autotest Manager has been tested with JRE 1.6 on Windows XP platform. It has not been tested on other platforms however which may produce inconsistent results.

# Introduction

The TKW Autotest software is an extension to the Toolkit Workbench (TKW) and requires this software to be present to be able to function. The Autotest software   
provides the ability to execute multiple tests consecutively against a System Under Test (SUT) and will provide a report of the results which can be navigated to display details of individual outcomes. The tool is provided with testcases which are defined within the Supplier Certified Requirements Coverage spreadsheet. The testcases are grouped by Transport, Domains and bundles and complete sets or subsets of these testcases can be selectively executed. Evidence of the execution of these testcases is required as part of the ITK accreditation process.

The tool will enable suppliers to provide evidence for the ITK accreditation process more easily and vastly reduces the effort to execute required tests. This will enable suppliers to retest and regression test their software more often and will generally expedite the menial process of repeated test execution.

Generally, TKW Autotest is used as a substitute for a client system when the system under test is operating as a host, however it can be used to transmit business and Infrastructure acknowledgements for CDA documents operating as a substitute for the host. It is designed to automatically execute suites of test cases sending them to the system under test.

TKW Autotest works in 3 main modes: Client, Host and Standalone.

* Client mode where ATM prompts that particular requests are transmitted to it from the SUT, based upon a testcase selection.
* Host mode where ATM transmits requests to the SUT based upon a testcase selection
* Standalone mode is an interactive front end to the existing TKW transmitter, simulator and validate modes .

Current functionality of TKW Autotest:

* Supports SUT Host and SUT Client test modes
* Allows user to visually define suites of tests
* Supports the following domains and test packs:
  + Child Screening
  + Correspondence
  + HL7v2 ADT – Patterns 1 and 2
  + HSCI
  + PDS Mini Services
  + Telehealth
  + Client negative acknowledgement tests
* Interactive graphical user interface
* Reports the command line system output from the instance(s) of Toolkit Workbench
* Performs synchronous and asynchronous ITK messaging
* Records all interactions
* Validates all incoming messages
* Supports standalone mode for TKW transmit, simulate and validate functions
* Generates a global report per autotest run, which can be navigated through using hyperlinks to validation reports and test message logs
* Stores Test Runs in a separate directory structure
* Previously executed autotest Test Runs can be “replayed”

SUT Host Mode

TKW Autotest

TKW transmit and synchronous messaging simulator

System Under Test

TKW asynchronous messaging simulator

Asynchronous response acknowledgement

Synchronous message

Synchronous response/ack

Asynchronous response

SUT Client Mode

TKW Autotest

Synchronous response/ack

Synchronous message

Asynchronous response acknowledgement

Asynchronous response

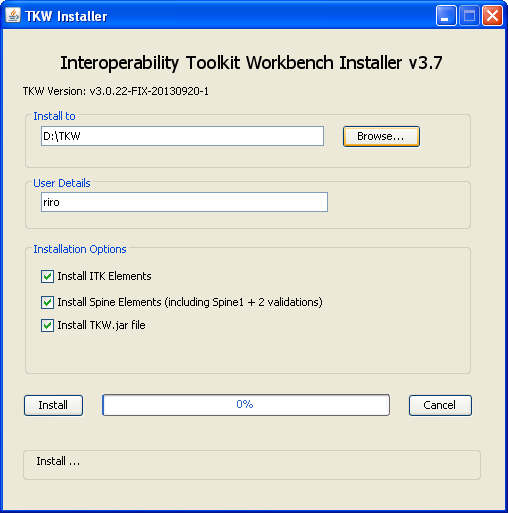
System Under Test

## Installation

TKW Autotest is installed as part of a full installation of TKW (see the Toolkit Workbench Guide for details) and exists within a separate folder within the contrib directory (TKWAutotestManager). As a minimum, ensure that the following installation options are checked:

Install ITK Elements

Install TKW.jar file



TKW is configured to work “out of the box” and should need little in the way of changes to its properties – see 2.3.1

## Directory Structure

TKW Autotest is installed into the TKW/contrib/TKWAutotestManager directory and has the following elements:

***auto\_tests*** directory**:** This directory contains the test evidence from each of the runs performed by AutotestManager. This directory will not be present before the first execution of the tool.

***transforms*** directory: This directory contains xslt transforms used to manipulate test messages during the autotest software execution

***autotest\_logs*** directory**:** This directory is used for the logs which are created when Autotest test runs are executed. It contains both the TKW logs and the TKWAutotestManager java application logs.

***tst*** directory**:** this directory contains autotest script files which are merged instructions for the execution of complete autotest runs. These files allow TKW Autotest to be executed from a “saved” set of testcases. This means that TKW Autotest runs can be re-executed without their particular testcase selection being remade through the frontend. This directory will not be present before the first execution of the tool.

***tstp*** directory**:** This directory contains the autotest script template files. These are files which programmatically describe the individual testcases which are defined in the Supplier Certified Requirements Coverage document. They are built up into tst files as described above.

***Supplier Certified Requirements Coverage vX.x.xls*:** This spreadsheet is a copy of the document available to all suppliers as a record of the requirements which must be covered for ITK Accreditation and a record of supplier self certified statements and evidence mapped against each requirement. **This document must not be moved or altered as it is used by TKW Autotest Manager**

***TKWAutotestManager.properties***: these are properties files which TKW Autotest relies on. Their properties are not intended for direct editing

***TKWAutotestManager.jar*:** Java executable jar file – double clicking will execute the TKW Autotest Manager

# Using the TKW Autotest Manager

Once TKW (including Autotest Manager elements) is successfully installed, the software can be executed from the command line using:

java -jar {path}TKWAutoTestManager.jar

where {path} is the path to the TKW/contrib/TKWAutotestManager

Or, as the TKWAutotestManager.jar is an executable jar file it can be run by double clicking.

## Configuration Tab

TKW Autotest Manager opens, showing the Configuration tab.

This screen contains the main configurations for TKW Autotest Manager:

***Requirements File*:** This is the file path of the ITK Accreditation Team’s Supplier Certified Requirements Coverage spreadsheet. It is populated out of the box.

***Autotest Manager URL*:** This is the URL from which the generated message will be sent - including port of the TKW Autotest software, but must not contain a context path.

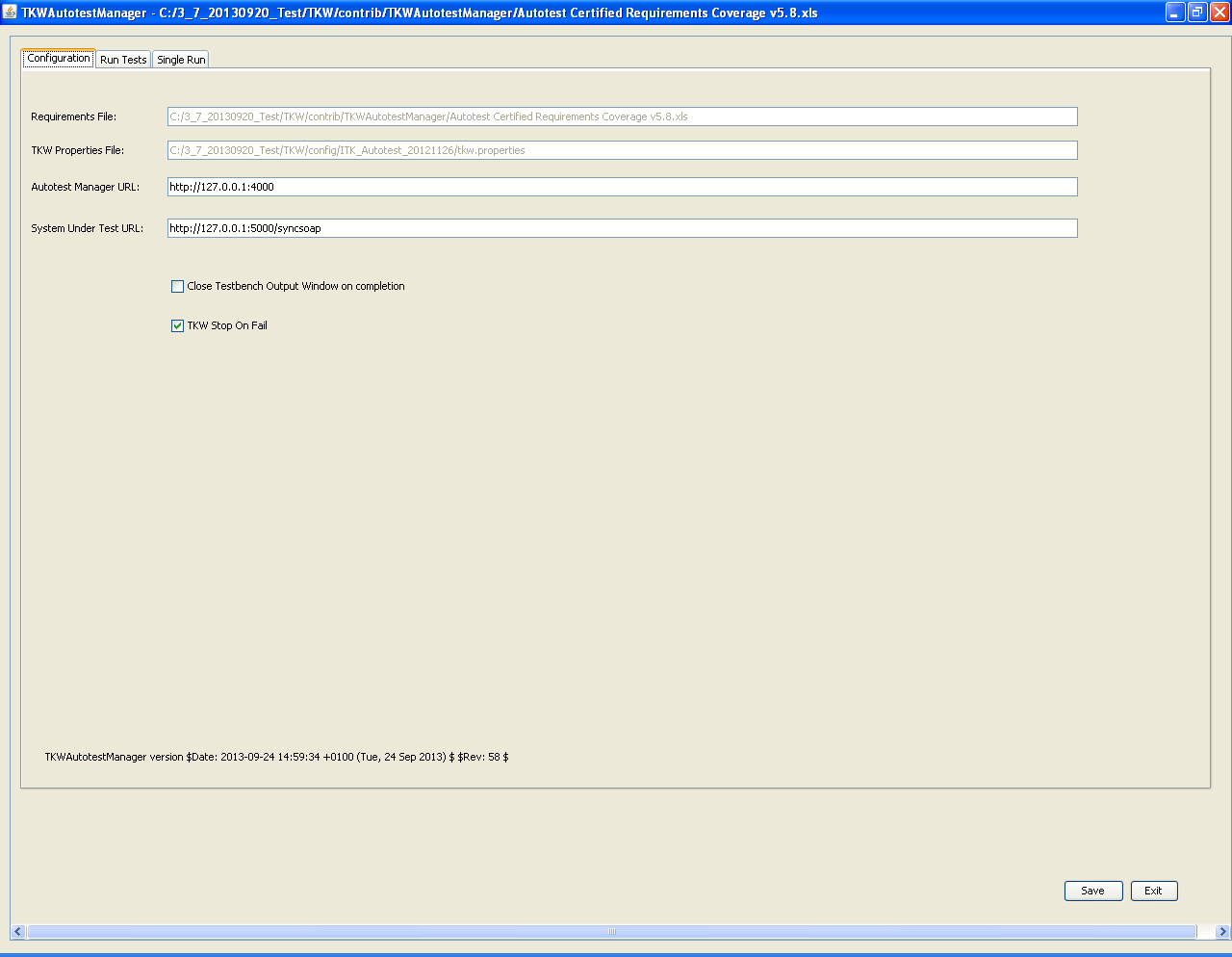
***System Under Test URL*:** This is the URL to which the generated message will be sent (overrides transmitter settings in tkw.properties file - including port of the System Under Test. This must contain a context path (“/” is an acceptable context path)

The Autotest Manager can be run in loopback mode where the to and from URLs are set to the same value. A warning in the status bar at the bottom of the application window will appear as this can produce unexpected results.

***Close Testbench Output Window on completion***: will close the TKW ATM after completion of a test run.

***TKW Stop on Fail:*** will cease execution of a test case when a failure occurs. This state is not a saveable configuration setting

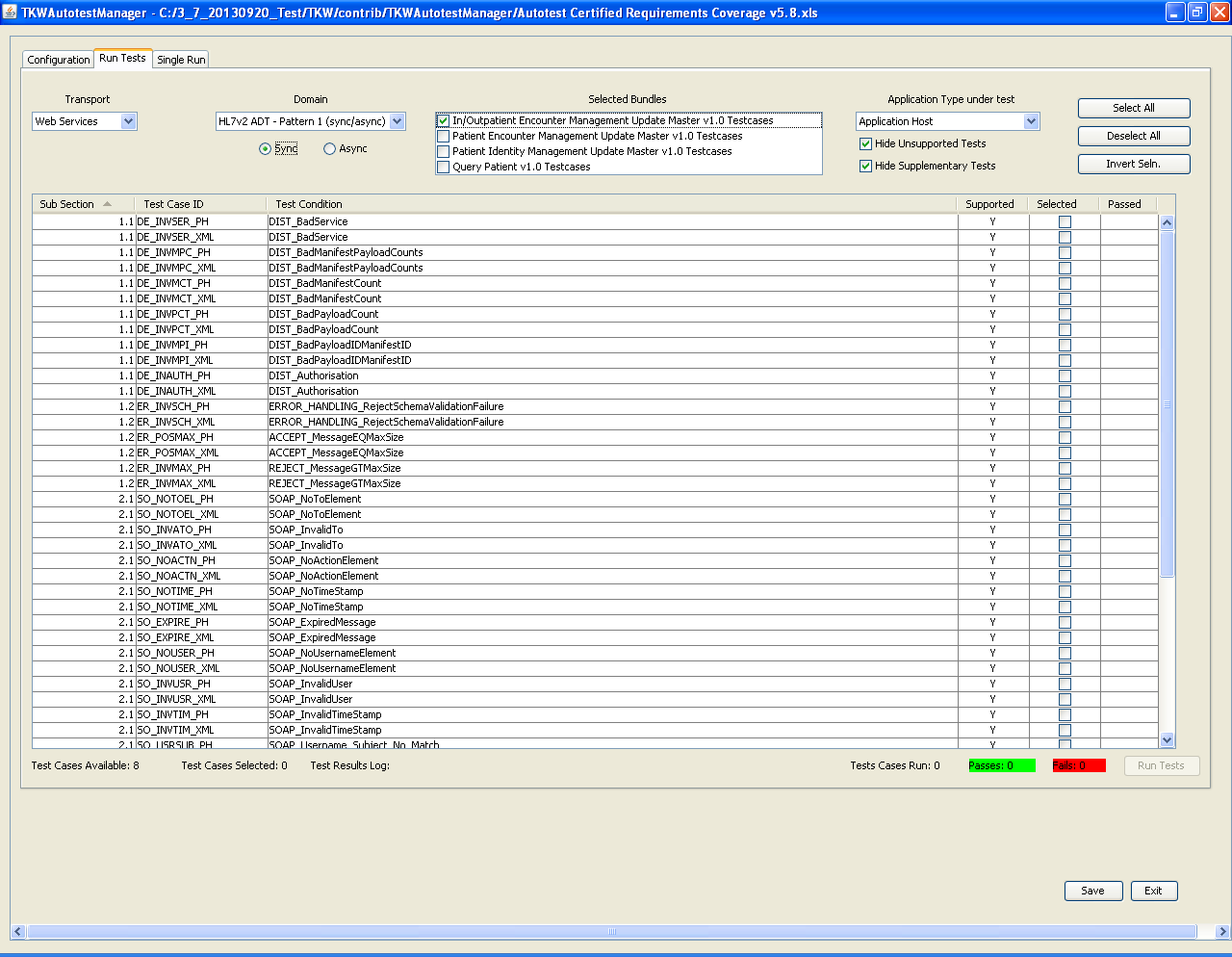
This tab also displays the ATM version details.



## Run Test Tab

The Run Tests tab allows the selection of test cases in order to create a “Test Run”. Autotest executions are called Test Runs which are composed of one or more individual test cases.

The Transport, Domain and Bundle selection lists, form a filter to present a super set of test cases, of which one or more may be selected from the test case selection table to form a Test Run. This allows for Test Runs to be executed at Domain level.



***Selected Transport:*** Currently there is only one selectable transport type which is Web Services. One Transport can be selected per Test Run

***Selected Domain***: Shows the Domain selected and allows the choice via radio button of using synchronous or asynchronous messaging. Asynchronous messaging is currently only applicable to “HL7v2 ADT – Pattern 1 (synch/asynch)” domain – all others will use the default synchronous message type. One Domain can be selected per Test Run

***Selected Bundles*:** Shows the selected bundle which is under test. One or more bundles can be selected per Test Run. Multiple rows in the test case selection table can be selected - right clicking will show a dropdown of check, uncheck or invert selection of the highlighted testcases.

***Application Type under Test*:** Application Types applicable to the Domain selection are displayed: Application Host, Application Client and SMSP Host are available for the System Under Test.

***Hide Unsupported Tests***: Not all test cases are applicable to testing via the TKW Autotest functionality. When this is checked, these tests will not be displayed in the test case selection table. However when the checkbox is unchecked the entries will be displayed but greyed out and for information only – they cannot be selected.

***Hide Supplementary Tests***: Supplementary tests are those which do not form part of the accreditation but are available for suppliers to further test their application. When this is checked then these tests will not be displayed in the test case selection table.

***Test Case Selection Table***: This table displays the result of the macro-filtering of the test cases by domain and bundle as described above. Each row displayed relates to one testcase and is directly related to the Testcases Index tab in the Supplier Certified Requirements Coverage spreadsheet. These can be cross referenced to the spreadsheet by using the test cases “Subsection”, “Test Case ID” and “Test Condition” column entries. A tooltip appears when the mouse pointer is hovered over the Test Condition entry displaying the full Test Condition description from the spreadsheet.

***The Applicability column*** denotes a test cases’ applicability to the bundle – it may be optional or applicable. A tooltip expands on this.

The column “***Supported***” states whether this test case is supported by TKW Autotest and can be filtered against as described above.

The “***Selected***” column allows selection of one or more supported test cases to be part of a Test Run. These can be set “en-masse” using the 3 buttons at the top of the test case selection table: Select All, Deselect All and Invert Selection or via right clicking which will show a dropdown of check, uncheck or invert selection of the highlighted testcases.

***Passes/Fails*** column shows the pass/fail status of the testcase row

The selections for each Test Run, once all the required test cases have been selected, can be saved for subsequent execution via the command line by pressing the Save button. This writes to the TKWAutotestManager.properties file and will present the same selections and all other configuration items next time the tool is opened.

### TKW Autotest Test Run Execution

Once one or more supported test cases are selected, the Run Tests button is enabled. The number of test cases selected is totalled at the bottom of the table.

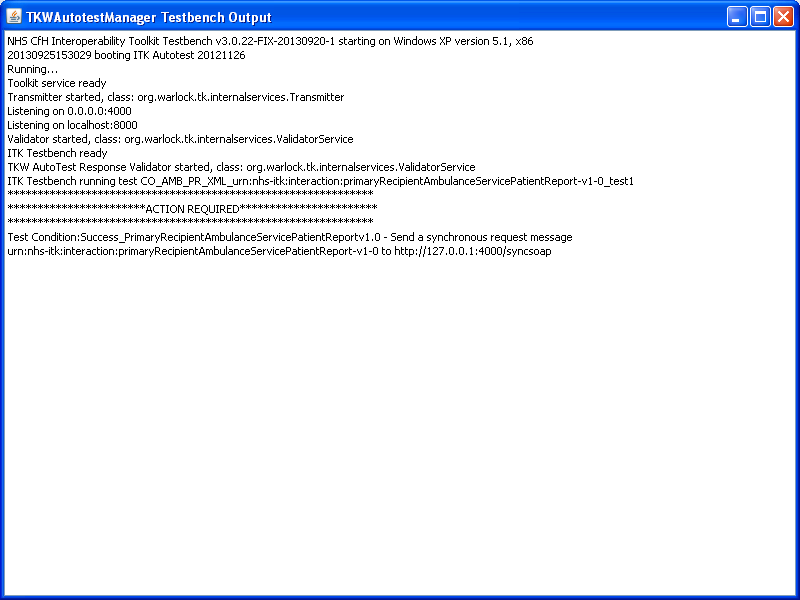
When the Run Tests button is pressed the selected Autotest Test Run is executed. A new window appears: TKWAutotestManager Testbench Output. This is the collected standard output from the instance/instances of TKW which have been executed in order to process a Test Run. It is colour-coded with “information” appearing in black and “error/warning” messages appearing in red. Depending on the volume of tests performed, this may take some time to complete.

#### Client Application Execution

When executed with ApplicationType under test as Application Client, ATM will request particular messages be sent synchronously/asynchronously to a particular [url:port](file:///D:\Documents%20and%20Settings\riro\Application%20Data\Microsoft\Word\port).

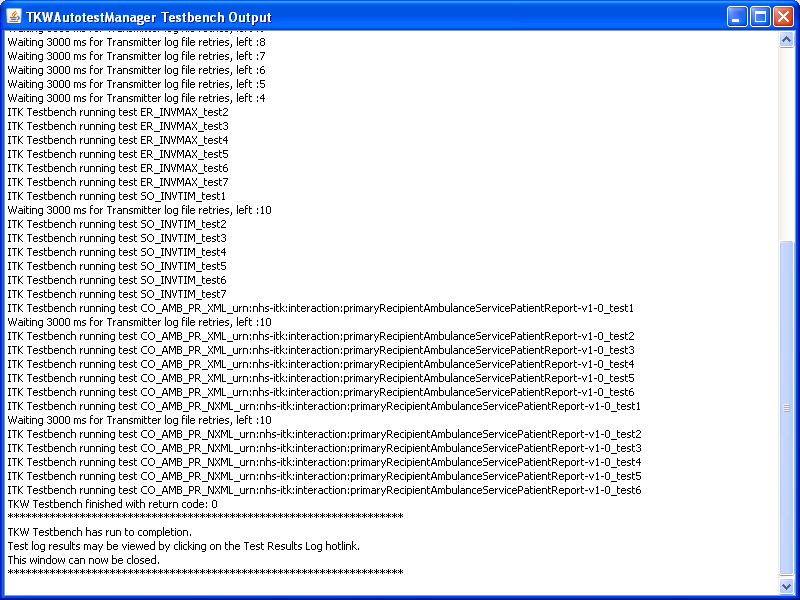
ATM will wait until a message is sent to this endpoint – ATM will respond appropriately, whereupon any further requests may be made depending on the selections made previously.

ATM will then validate all the requests, reporting back to the Autotest GUI.



#### Host Application Execution

When executed with ApplicationType under test as Application Host, ATM will execute the transmission of each of the messages connected with the test case from an instance of TKW in client mode. It will then record any synchronous or asynchronous responses from the SUT and will validate these responses, reporting back to the Autotest GUI.



#### Results from Client or Host Application Execution

When the execution has successfully completed the following message will be displayed:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TKW Testbench has run to completion.

Test log results may be viewed by clicking on the Test Results Log hotlink.

This window can now be closed.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

For each Test Run, Autotest creates an autotest script file (.tst) in the tst folder and names it with a timestamp. This is a merge of the programmatic instructions for each of the test cases chosen and will allow for this Test Run to be repeated. See section: 2.4

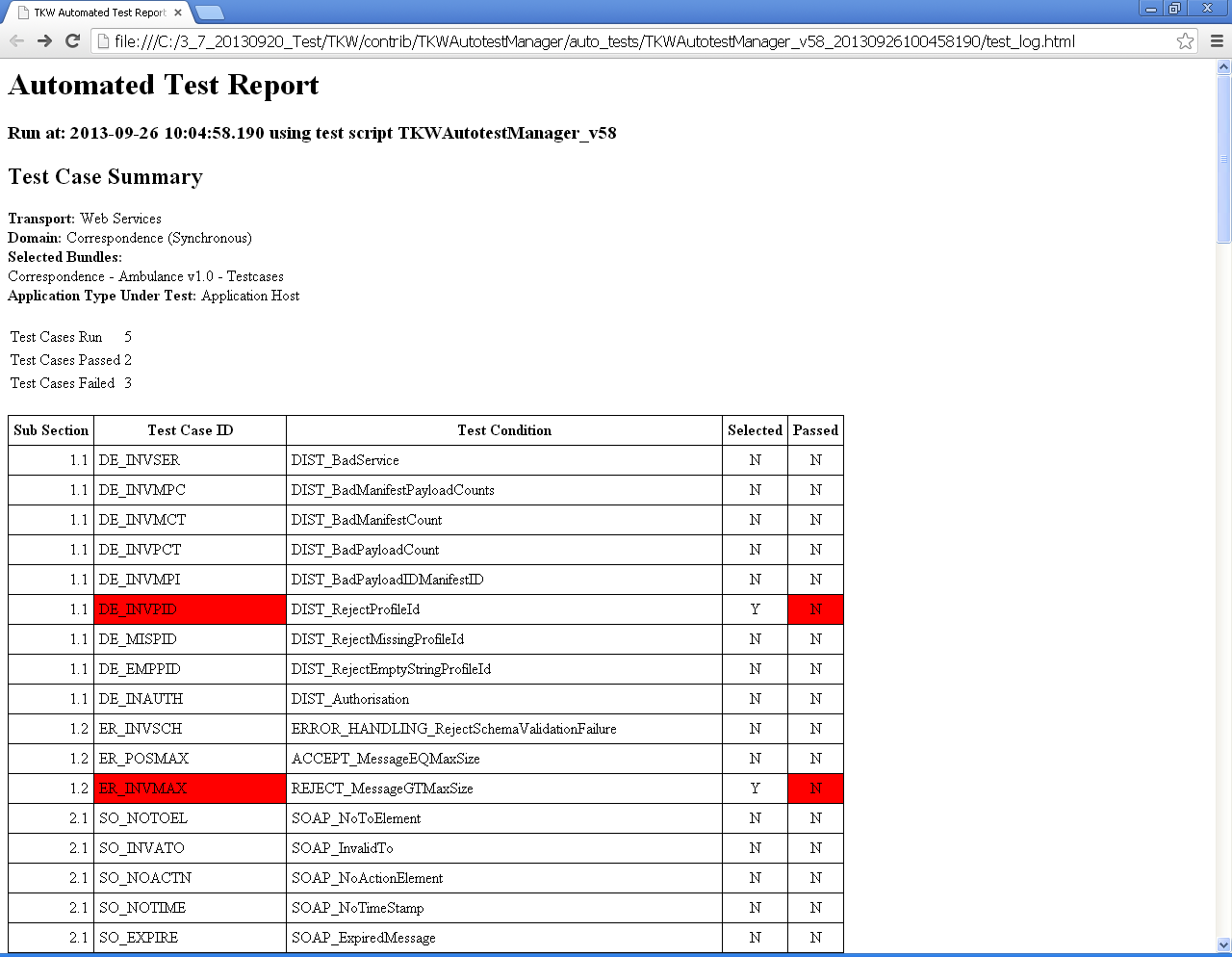
### Test Evidence

After a successful Test Run a Test Results log hyperlink will appear below the test case selection table with a confirmation of how many tests were run and a tally of passed and failed tests.

If a Test Run fails this will be reported to the TKW Autotest Manager interface as a red banner in the status bar at the bottom of the application pane and the hyperlink and summary of passes and fails will not be displayed

Clicking this hyperlink after a successful execution will open an Automated Test Report in the same style as a validation report created using TKW. This contains:

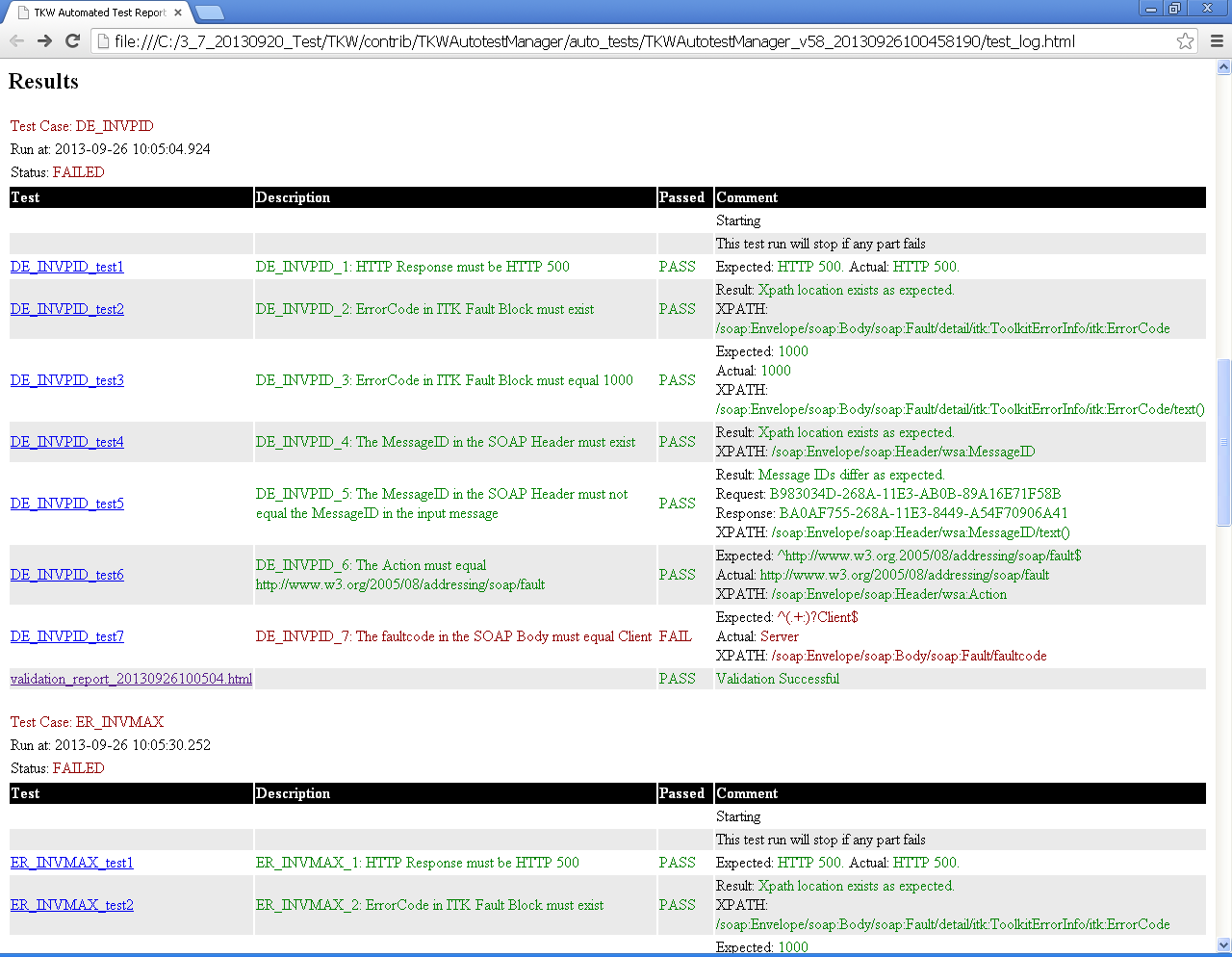
* a timestamp
* a Test Case Summary
  + Transport
  + Domain
  + Selected Bundles
  + Application Type Under Test
* a representation of the testcase selection table from ATM
* a summary of passed/failed tests



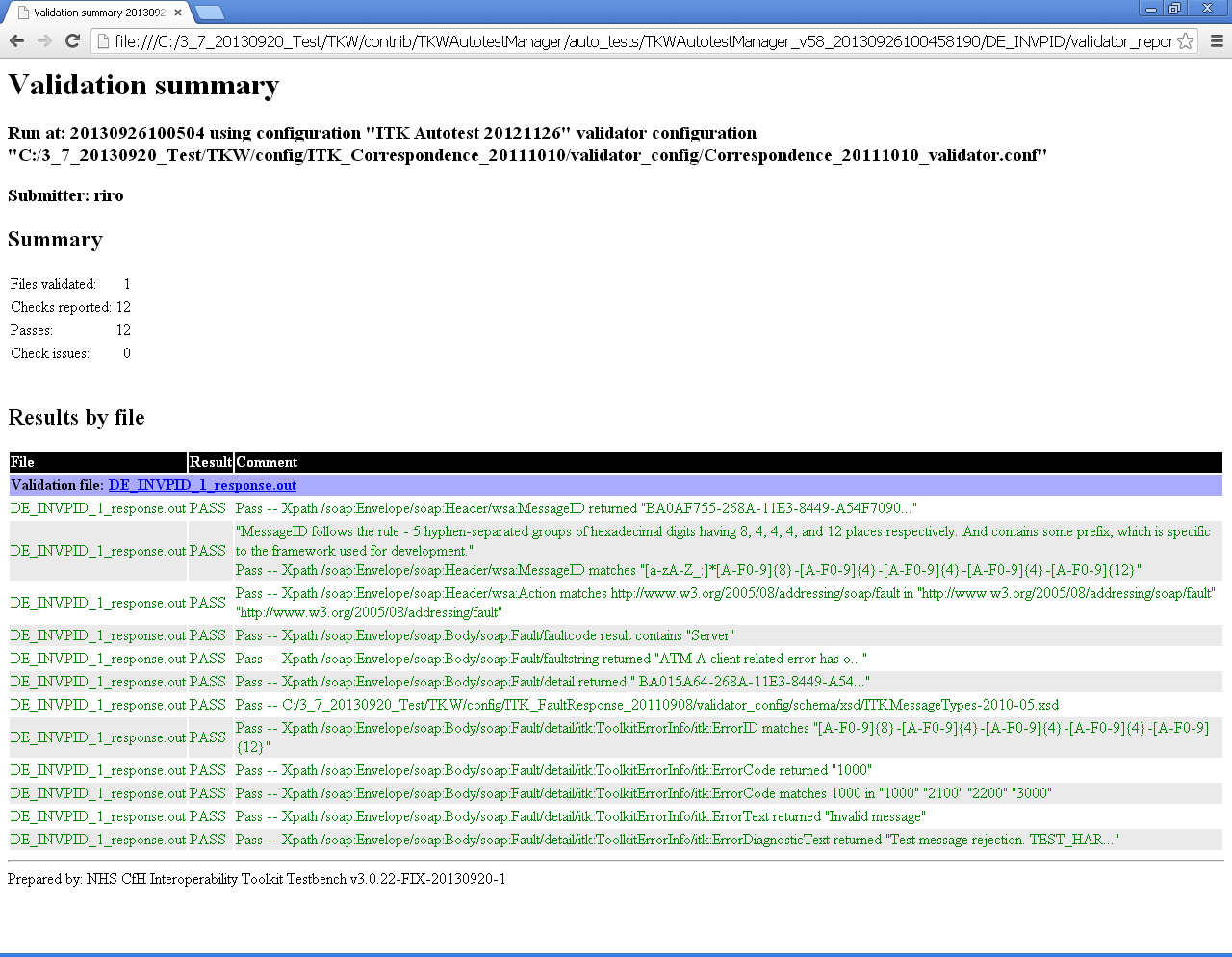
For each of the test cases there is recorded:

* Schedule name of the test case
* Timestamps of each execution activity
* A test reference with a hyperlink leading to the message log for this test
* A PASS/FAIL indicator
* Execution comment with more information on the row
* Validation Report hyperlink per test case

Rows containing test fails are coloured red. The figure below shows an example of a test report.



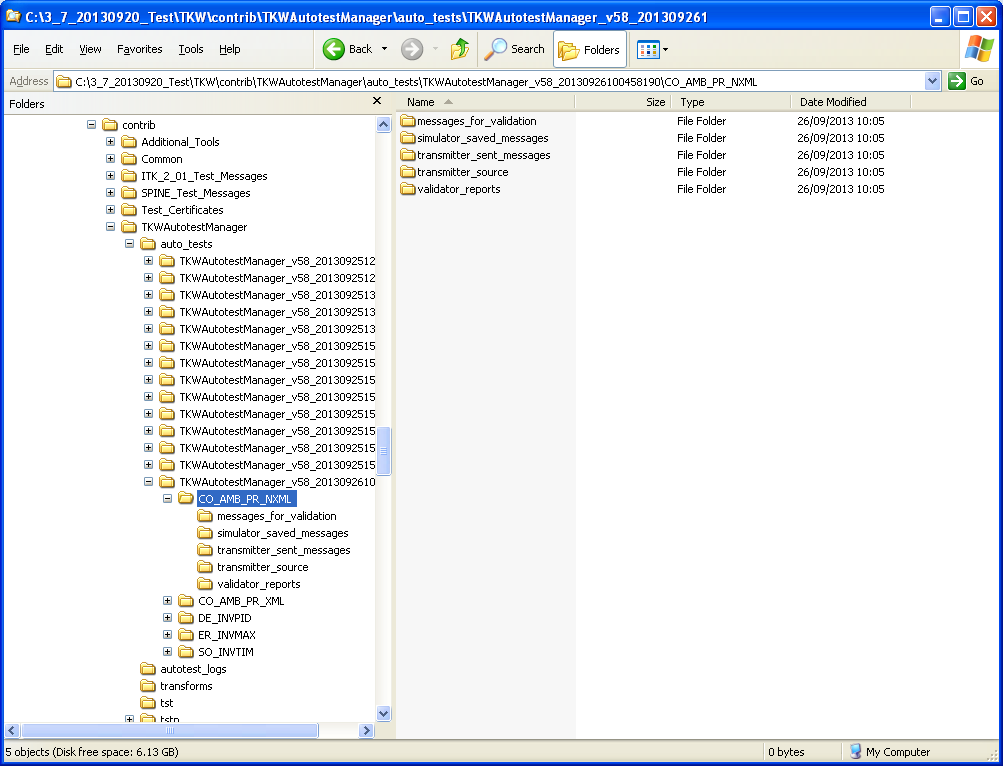
Opening the Validator Report hyperlink will show a list of the exchanged messages which were recorded during the Testcase. The results are summarised at the top and each of the messages has a validation result against it



All the test evidence for each Test Run is stored in its own folder within the Autotest directories: {path}/TKW/contrib/TKWAutotestManager/auto\_tests

Each Test Run is assigned a directory named after “Autotest\_{timestamp}”. Within this directory is a number of subdirectories, one for each of the test cases selected in the Autotest Manager. The Automated Test Report is saved here as test\_log.html

Within each of the test case directories there are 5 standard folders message interaction folders (messages\_for\_validation, simulator\_saved\_messages, transmitter\_sent\_messages, transmitter\_source, validator\_reports).



### Execution of Client Negative Infrastructure or Business Acknowledgement Tests

When selecting the test scenarios for a client system under test for sendCDA bundles, it should be noted that there are only positive testcases present. The negative testcases are executed in a different manner.

In order for suppliers to exhibit correct response behaviour to incorrect inbound business/infrastructure acks, the bundle: Client Negative Acknowledgement Tests should be chosen. This will proactively transmit business acknowledgements to the SUT

This covers negative tests for the following bundles

* Correspondence
* HSCI
* Telehealth
* Child Screening

The following diagram shows the messaging transaction for sendCDA messages:

Application Client

Application Host

sendCDADocument-v-2.0

Infrastructure Acknowledgement

HTTP 200

HTTP 200

HTTP 200

Infrastructure Acknowledgment

HTTP 200

Business Acknowledgment

### Repeat an execution of a previously run Test Run

A re-run of a previously executed Test Run is possible:

Identify the timestamp named .tst file in the tst directory which was created when the previous execution was first run.

Use the following command:

java –jar {path}TKW.jar –autotest {path}tkw.properties {path}20121112153431278.tst

## Single Run Tab

The single run mode allows the use of TKW directly through a GUI rather than through the command line. The functionality of TKW is described in “A Guide to the Toolkit Workbench” document.

The Transport, Domain and Directory locations define the parameters of the ITK interactions.

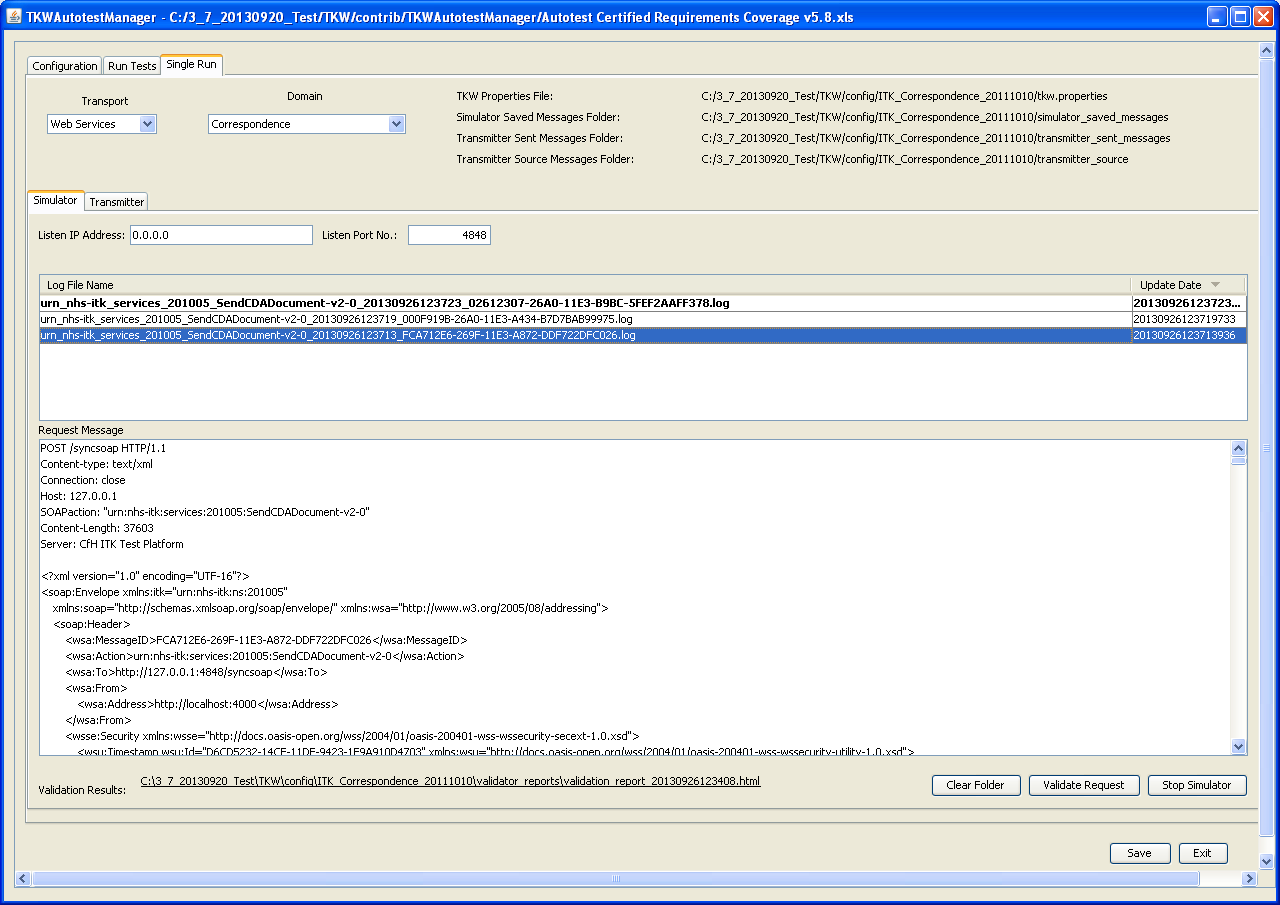
***Selected Transport:*** Currently there is only one selectable transport type which is Web Services. One Transport can be selected per run

***Selected Domain***: Shows the Domain selected. One Domain can be selected per Test Run

The directory/locations for TKW Properties file, Simulator Saved Messages, Transmitter send messages and Transmitter source messages folders are automatically populated from the Bundle selection.

### Simulator Tab

TKW Simulator acts as a programmable endpoint for ITK messaging, responding to requests based upon a ruleset defined within a properties file.

******

***Listen IP Address***: This is the address which simulator will listen

***Listen Port Number***: This is the port which simulator will listen – numerals only.

Errors in the address/port will be reported at the bottom of the screen. The listen address/port details input here overrides the values in the tkw.properties file.

***Log File Name panel:*** reports any messages received by the simulator. This is a view of the Simulator Saved messages folder. New unread messages logs appear in bold. Message logs can be dragged and dropped out of the panel. All drag and drop operations are copy operations of the log files.

***Request Message Panel:*** if the request message in the log file selected in the log file name panel above is well formed, the Request Message panel will pretty print this message and display it.

***Start Simulator button:*** Pressing this button will start the simulator based on the details input at the top of the Single Run screen. A separate window appears displaying the “Java standard out” from the simulator instance.

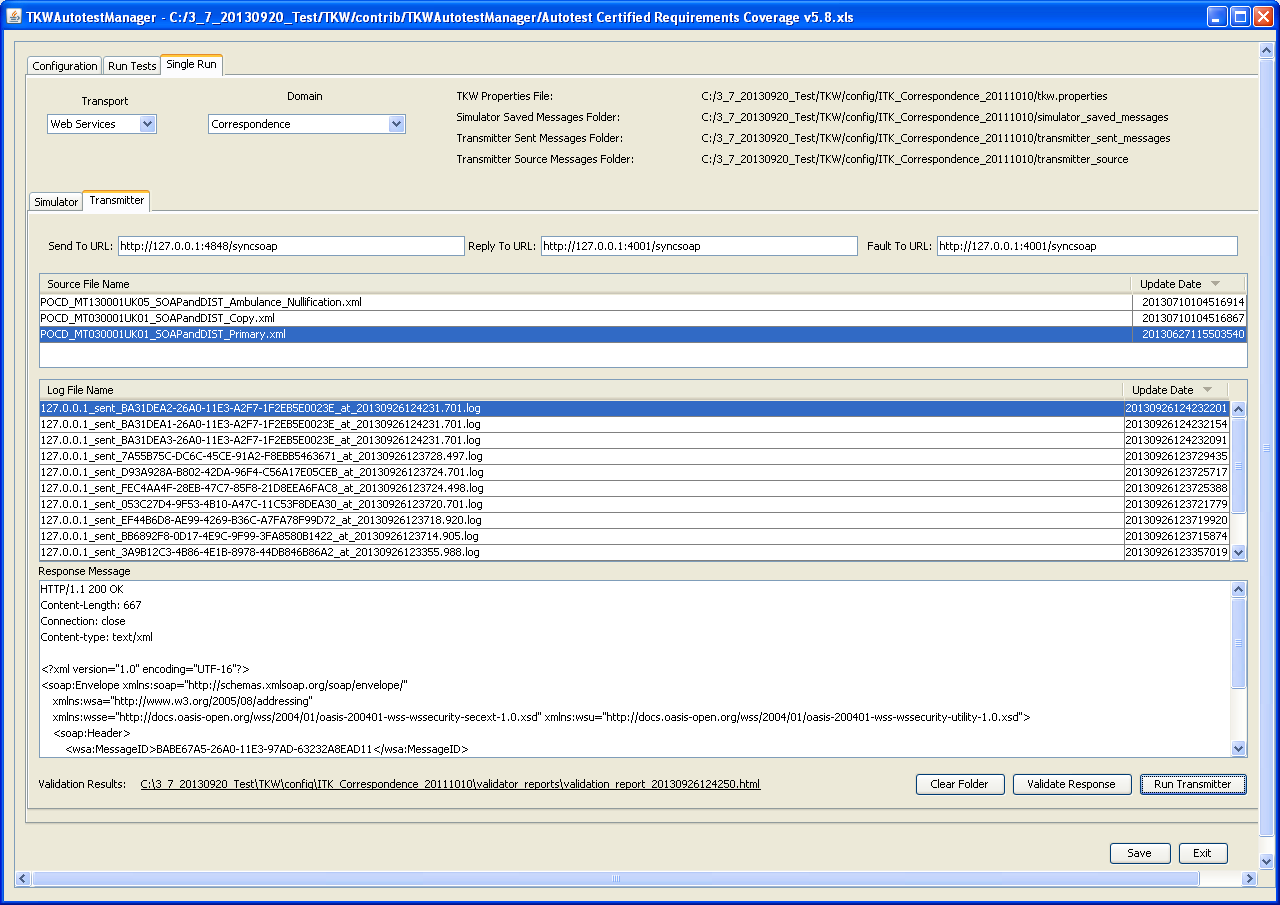
***Stop Simulator button:*** Pressing this button will stop the currently running simulator.

***Validate Request button***: Pressing this button will validate the currently selected log file, opening a new “java standard out” window. A hyperlink to the results of the validation will be displayed at the bottom of the window.

***Clear Folder Button:*** This button clears down the contents of the Simulator Saved Messages directory.

### Transmitter Tab

TKW Transmitter acts as an ITK message transmitter, sending requests based upon a ruleset defined within a properties file.

******

***Send to URL***: Fully qualified endpoint to which the message will be sent. Replaces any instances of \_\_SENDTO\_\_ substitution tag in message

***Reply to URL***: Fully qualified endpoint to which an asynchronous message request fault should be sent. Replaces any instances of \_\_FAULTTO\_\_ substitution tag in message. (Only required for aynchronous requests.)

***Fault to URL***: Fully qualified endpoint to which an asynchronous message response should be sent. Replaces any instances of \_\_REPLYTO\_\_ substitution tag in message. (Only required for aynchronous requests.)

***Source File Name panel:*** Lists messages to be sent to an endpoint. This is a view of the Transmitter Source Messages folder. Messages can be dragged and dropped into the panel. All drag and drop operations are copy operations.

***Log File Name panel:*** reports any response messages received by the transmitter. This is a view of the Simulator Saved messages folder. Message logs can be dragged and dropped out of the panel. All drag and drop operations are copy operations of the log files.

***Response Message Panel:*** if the request message in the log file selected in the log file name panel above is well formed, the Response Message panel will pretty print this message and display it.

***Run Transmitter button:*** Pressing this button will transmit all the messages in the Source File Name Panel based on the details input at the top of the Single Run screen. A separate window appears displaying the “Java standard out” from the transmitter instance.

***Validate Response button***: Pressing this button will validate the currently selected log file, opening a new “java standard out” window. A hyperlink to the results of the validation will be displayed at the bottom of the window.

***Clear Folder Button:*** This button clears down the contents of the Transmitter Sent Messages directory.