DT-116807 Iss 2

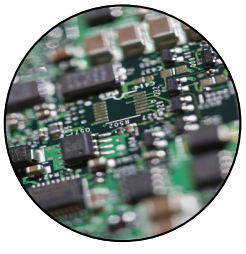
Tequra®
Framework

Product User Guide

Accelerated Test-Solution Development

Enhanced End-User Experience



Simplicity ai

Contents

Introduction

Tequra Framework is a set of components for National Instruments (NI) TestStand, which provides enhanced features over the standard product.

While there are recommended practices for making the best use of the framework, it is not necessary to write TestStand sequences in a specific way to be able to utilise many of the features. The Tequra Framework tools are designed to run with existing code without requiring specific modifications.

This manual is designed for users who are familiar with TestStand development, including the role of a Process Model and the usage of custom step types.

Features Overview

Tequra Framework is made up of the following components:

Table 1: Tequra Framework - features overview

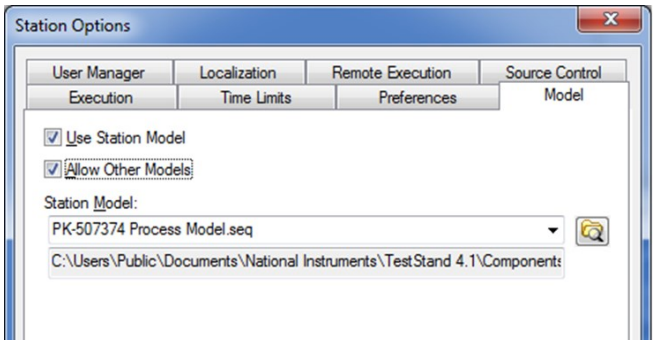
Process Model	Provides infrastructure for Sequential and Batch testing, with modular report generation including PDF, CSV and ATML output. Custom formats are available from Simplicity AI on request.
Test Steps	Enhanced versions of NI Test Steps with additional features. Also includes a Waveform Limit Test step which has no NI equivalent.
Flow Control Step	Adds “Step Selection” to Flow Control steps to provide operators the ability to skip certain tests at runtime, via a simple dialogue box.
User Interface	Enhanced version of NI’s Full Featured User Interface (Operator mode), including a results display and Yield counters.

Installation

Tequra Framework is provided with an installer which copies files to a number of standard TestStand file locations. To install, simply double-click the install.bat file and wait for the process to complete. The installation process should not affect any other customizations or installed components.

The installer does not automatically change the Station Model to use the Tequra Framework model; in order to do this it is necessary to open the TestStand Sequence Editor (or a Full-Featured User Interface), then select Configure >> Station Options >> Model and browse to:
`<TestStand Public>\Components\Models\PK-507374_Process_Model\PK-507374 Process Model.seq`,
as shown in Figure 1.

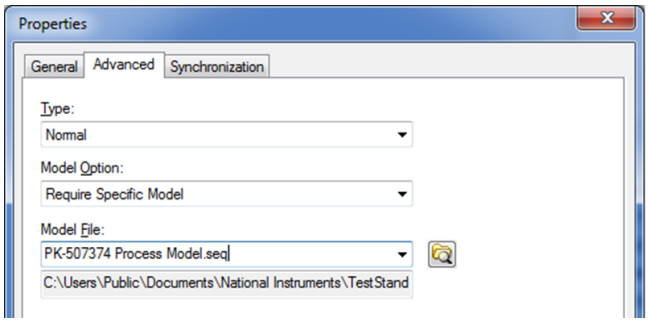
Figure 1: Station Options Model File Selection



If setting Tequra Framework Model as the station model is not possible, then it is possible to set certain sequences files to use this Process Model explicitly. Please note that the recommended approach is to set the Station Model, rather than explicitly specifying the model.

Using the TestStand Sequence Editor (or Operator Interface), open the test sequence file that you would like to modify to explicitly reference the Tequra Framework Model. Select Edit>>Sequence File Properties and select the "Advanced" tab. For the "Model Option", select "Require Specific Model", then click the browse button and select:
`<TestStand Public>\Components\Models\PK-507374_Process_Model\PK-507374 Process Model.seq`
as shown in Figure 2

Figure 2: Sequence File Properties Model File Selection



Process Model

Overview

The Tequra Framework Process Model is based on the NI Process Models to provide maximum compatibility with existing sequence files. In most cases, there should not be a requirement to modify existing sequence files in order to use the Tequra Framework Process Model. However, certain customizations are recommended to take full advantage of the enhanced features. Please refer to the *Modifying Reports* section for further details.

The largest difference from the NI Models is reporting; this has been completely re-written to provide a modular plug-in architecture capable of generating more report formats, including PDF, CSV and ATML. Test results are saved to an intermediate file during an execution, so multiple reports can be generated at the end of a run – or at any point in the future from the same source file. Additionally, this file is designed to allow large amounts of data to be stored, such as all the results from environmental testing which may last for days. Using standard TestStand components would mean that the system could run out of memory or produce reports that could not be loaded in a standard application due to their size.

The NI Process Models provide support for sequential, batch and parallel testing through separate Process Model files. To prevent code duplication and to allow easier switching between modes, the Tequra Framework Process Model provides separate entry points for sequential and batch testing using the same Process Model file. Parallel testing is not currently supported using the Tequra Framework Process Model.

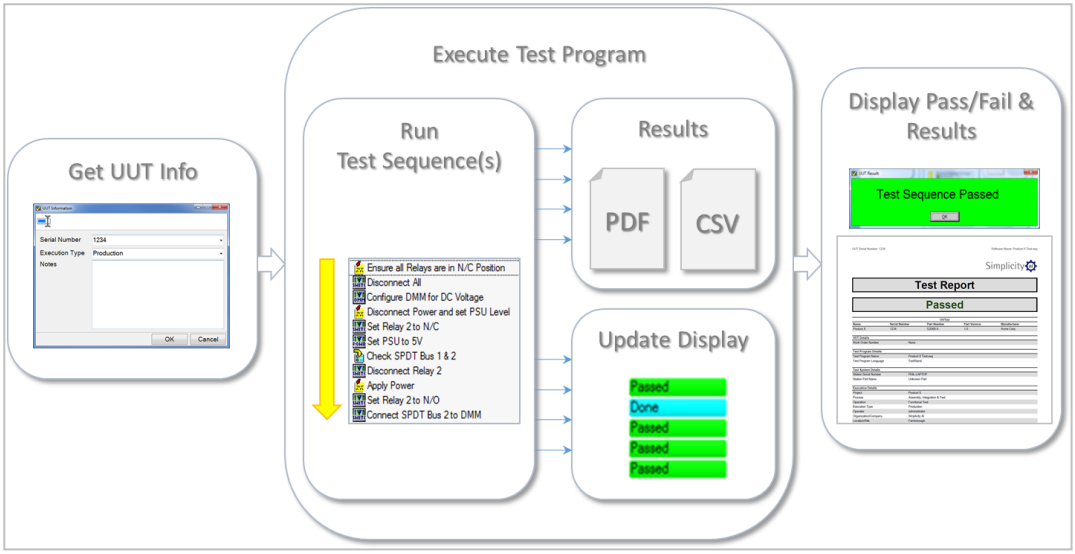


Figure 3: Selection of Features Provided By the Tequra Framework Process Model

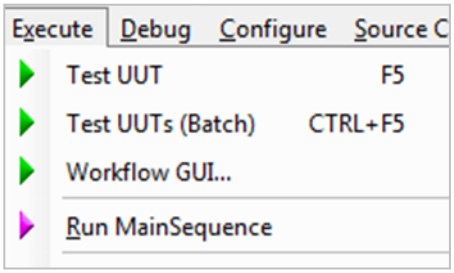
Executing Sequences

When loading up a sequence file in the Sequence Editor or Operator Interface there should be three execution entry points available “Test UUT”, “Test UUTs (Batch)” and “Workflow GUI…”. Note that at present “Workflow GUI” is a beta feature only so is not supported.

In both the Sequence Editor and Operator interface, the entry points are visible on the “Execute” menu as shown in Figure 3.

Figure 4: Execution Entry Points

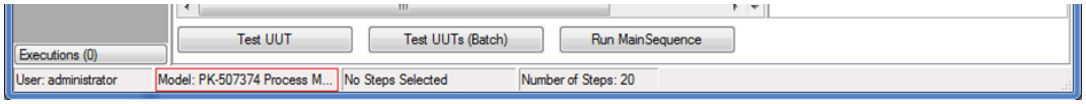
Execute Menu



In the operator interface, large buttons are visible at the bottom of the screen, as shown in Figure 4. Also highlighted in this figure is the indication of the current Process Model – which should be PK-507374.

Figure 5: Help File

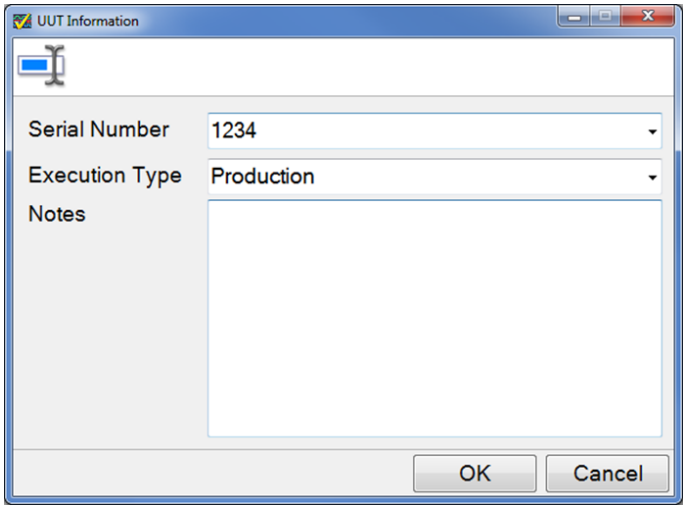
Operator Interface



Test UUT

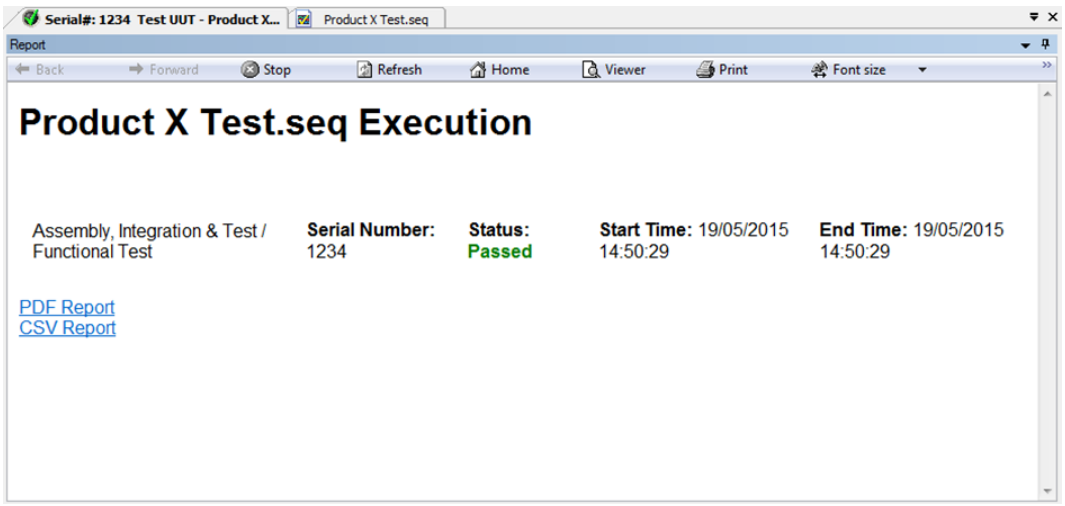
Clicking the Test UUT button fires off a single execution designed for testing one UUT. The Process Model will gather UUT information via an operator dialog box prior to executing the test sequence, as shown in Figure 6.

Figure 6: UUT Information



During the execution results will be gathered and stored in a local database file, then on completion a Pass/Fail result banner will be displayed, reports will be generated and an execution summary will be displayed as shown in Figure 6. This contains hyperlinks to the generated reports (which may be customized using Configure >> Logging Options).

Figure 7: Execution Summary



If the “Loop Sequential Executions” option is set in Configure >> Model options, then the UUT Serial Number dialogue box will be displayed again, ready for the next UUT.

Test UUTs (Batch)

Clicking the Test UUTs (Batch) button fires off multiple executions simultaneously, each running the same sequence file. Depending on the number of test sockets configured using Configure >> Model Options, the UUT Information dialog will be shown for each test socket for the operator to enter UUT details. Most process model operations (such as result logging) are duplicated, however it is possible to use certain shared callbacks which execute at the beginning and end of the batch test.

Please note that it is imperative that the client test sequence file is written in such a way to support parallel testing. For example, if the same sequence is fired off in two parallel executions and both try to access the same instrument at the same time, then they will interfere with each other and lead to errors or inconsistent results. For shared instrumentation, appropriate use of the TestStand synchronisation step types (see TestStand documentation) is required. To determine which test socket a particular execution belongs to, the `RunState.TestSockets.MyIndex` property can be queried. For example, in a system configured to run two UUTs in a batch, two executions would be started. The first would have a test socket index of 0 and the second would be 1.

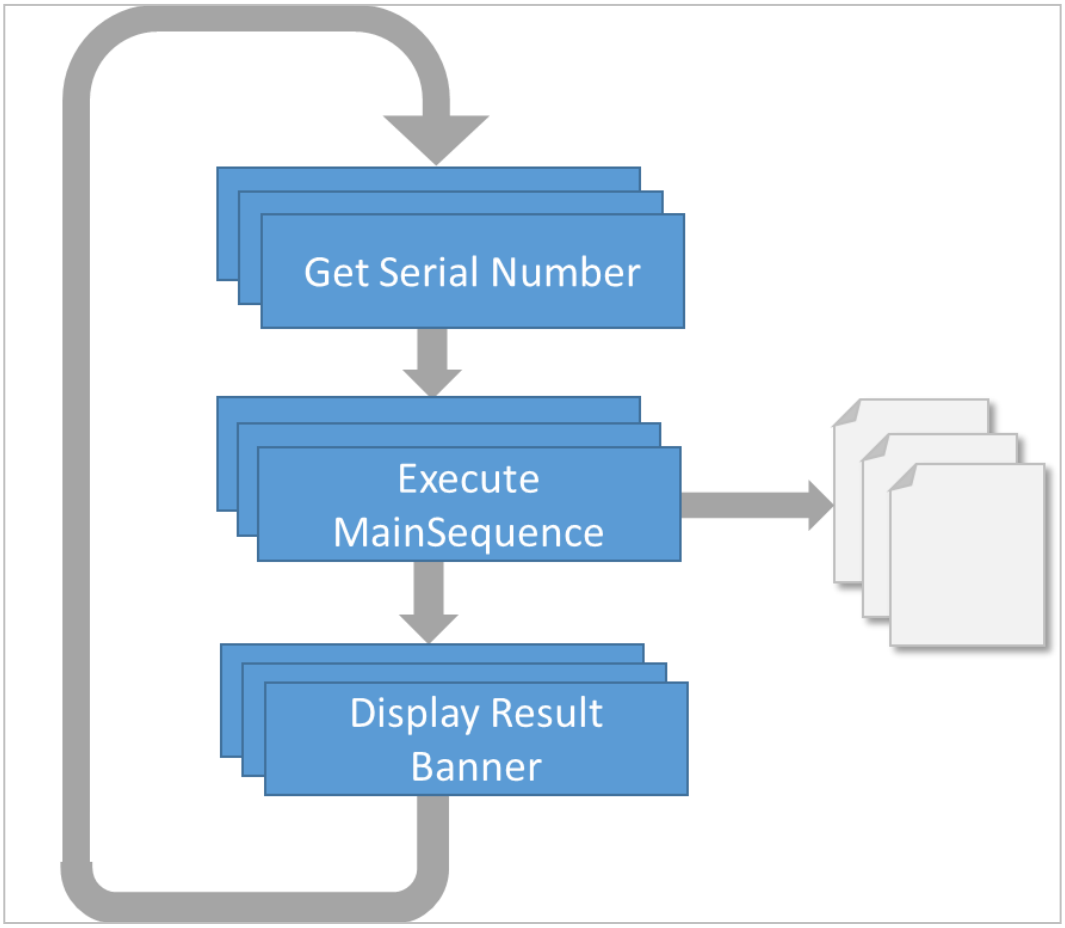


Figure 8: Batch Execution Enables Multiple Executions to be Started Simultaneously

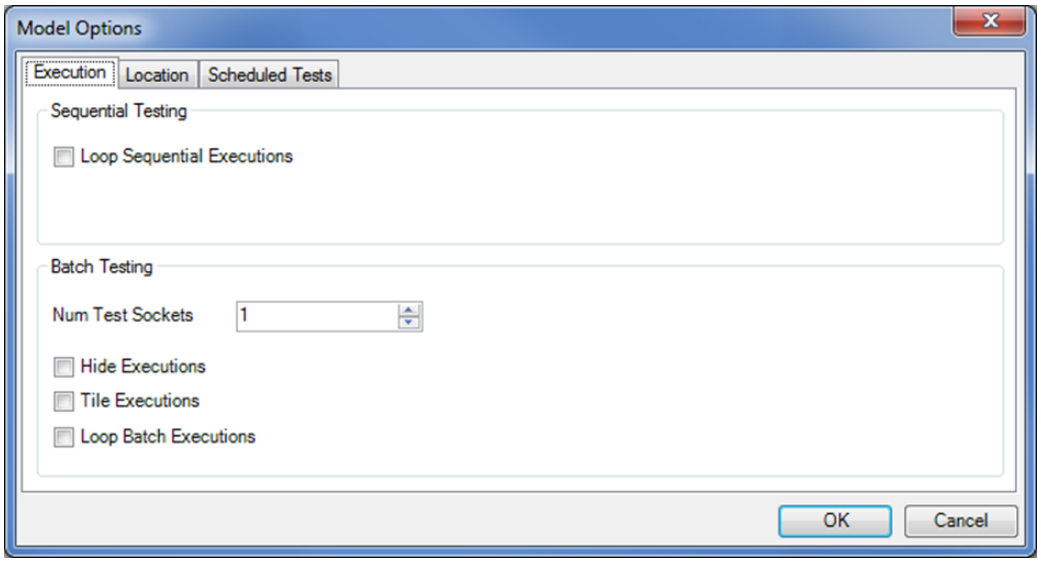
Customization

Model Options

Model options can be accessed using Configure >> Model options and will display the window shown in Figure 9.

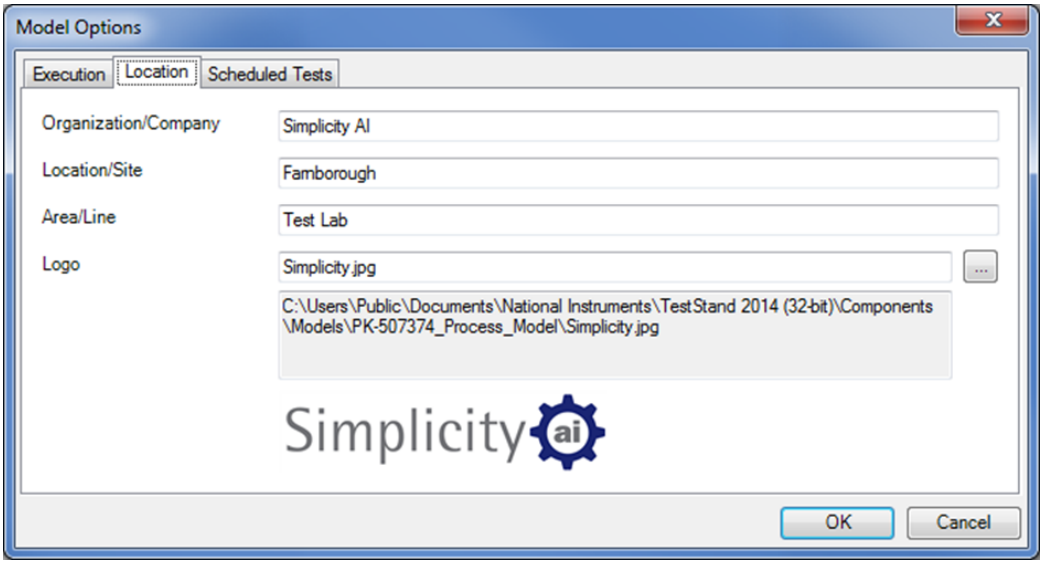
The “Execution” tab allows configuration of Sequential and Batch testing modes, with similar options to NI’s built-in Process Models. For sequential testing, this includes the ability to loop executions allowing operators to test units continuously without having to click “Test UUT” for each new unit. For batch testing, it is possible to set the number of test sockets as well as choosing how executions will be displayed.

Figure 9: Model Options Execution



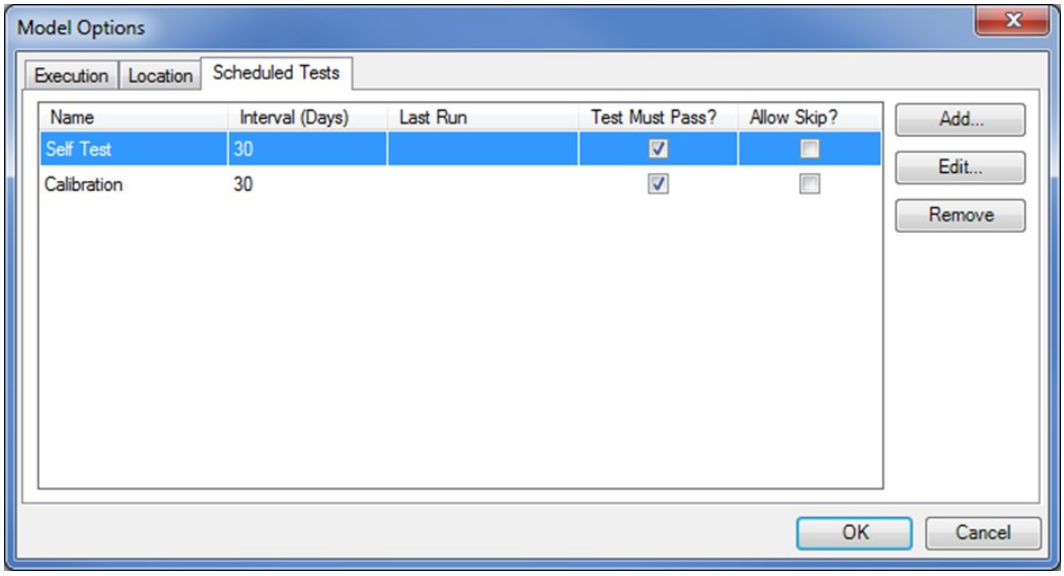
The “Location” tab, as shown in Figure 10, provides the ability to set the location of the Station as well as setting your company logo. Depending on the report type, this may be displayed on the first page of the report.

Figure 10: Model Options Location



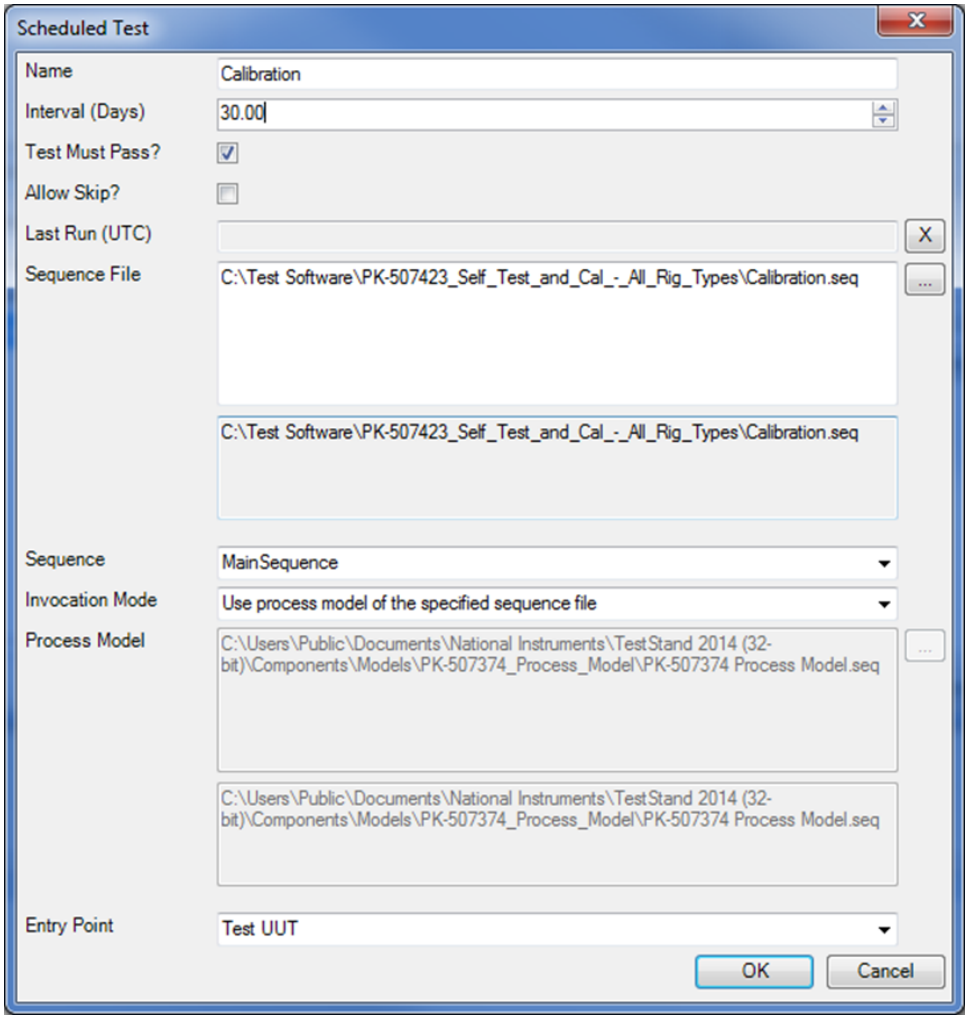
The “Scheduled Tests” tab, as shown in Figure 11, allows sequences to be configured such that they have to be run on a defined schedule. Typical use cases could include Self-Test, Calibration or Maintenance procedures.

Figure 11: Model Options Scheduled Tests



Adding or editing a scheduled test shows the dialog box as shown in Figure 12. From here it is possible to select the sequence to run, how often it should execute and whether to run the sequence directly or invoke via a Process Model entry point.

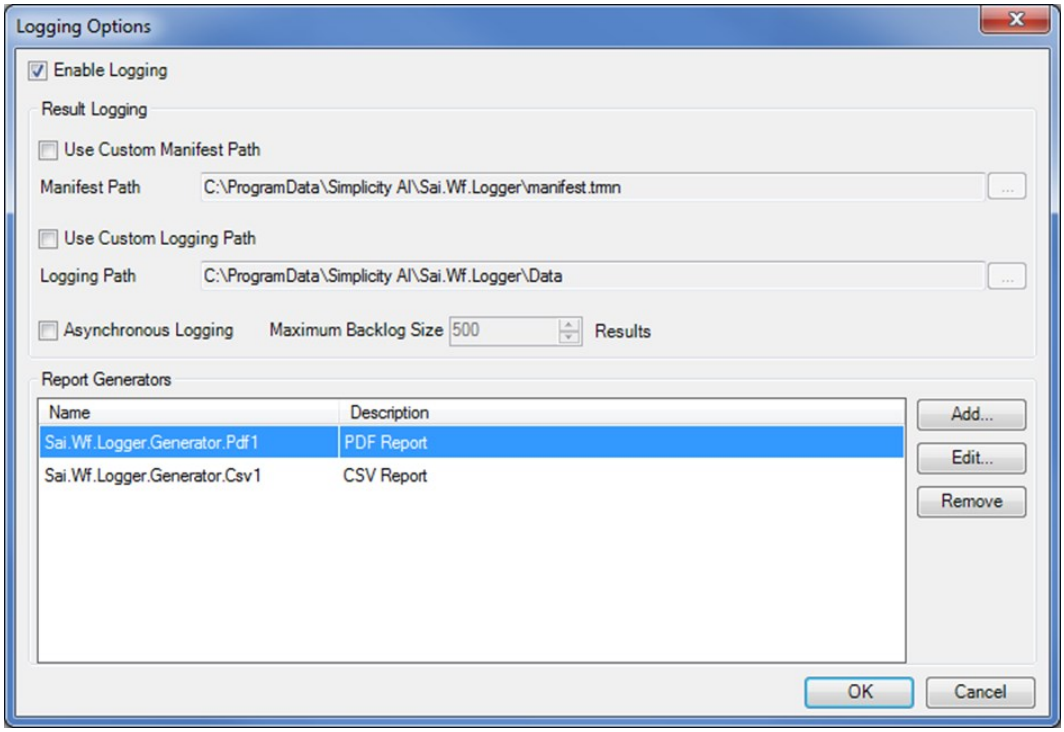
Figure 12: Scheduled Test Configuration



Logging Options

Options relating to result logging and report generation can be accessed via Configure >> Logging Options, as shown in Figure 13.

Figure 13: Logging Options



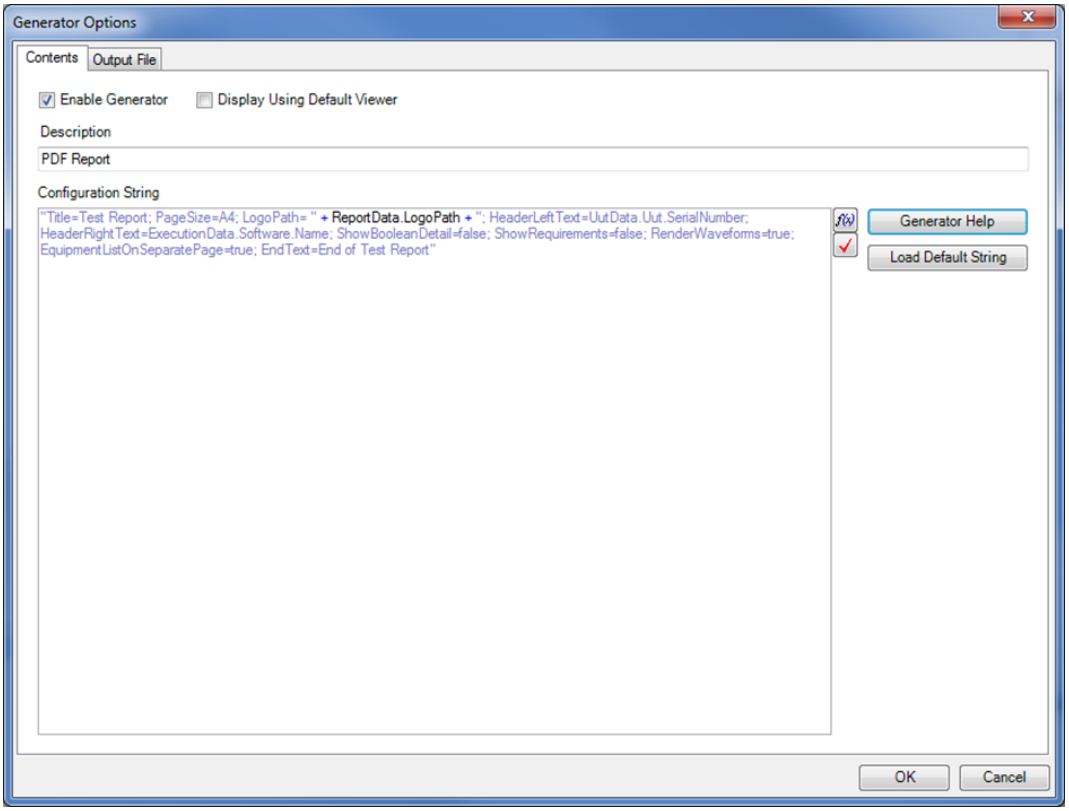
From this screen it is possible to customize the location of various log files. All executions are tracked using a manifest file, which keeps a record of the overall result and the location of reports generated. At present, no user tools are available to inspect the manifest – this is purely as a support aid as this file can be sent to Simplicity AI technical support if required to provide information about station usage. For each Execution (i.e. run of a sequence file using a Process Model entry point) a result file (*.trdb) is generated in the Logging directory. This file is updated “on-the-fly” during an execution and used to generate multiple reports at the end of the execution.

It is possible to enable “Asynchronous Logging” which will improve run-time performance by writing results on a different thread.

Report generators may be added to provide multiple output files for each execution. Each report can be customized by clicking the “Edit” button which displays the Generator Options, an example for a PDF report is shown in Figure 13. From this window it is possible to customize the generator via a configuration string, which contains different options depending on the selected generator. Help is available by clicking the “Generator Help” button; however in the majority of cases, the default values should be sufficient.

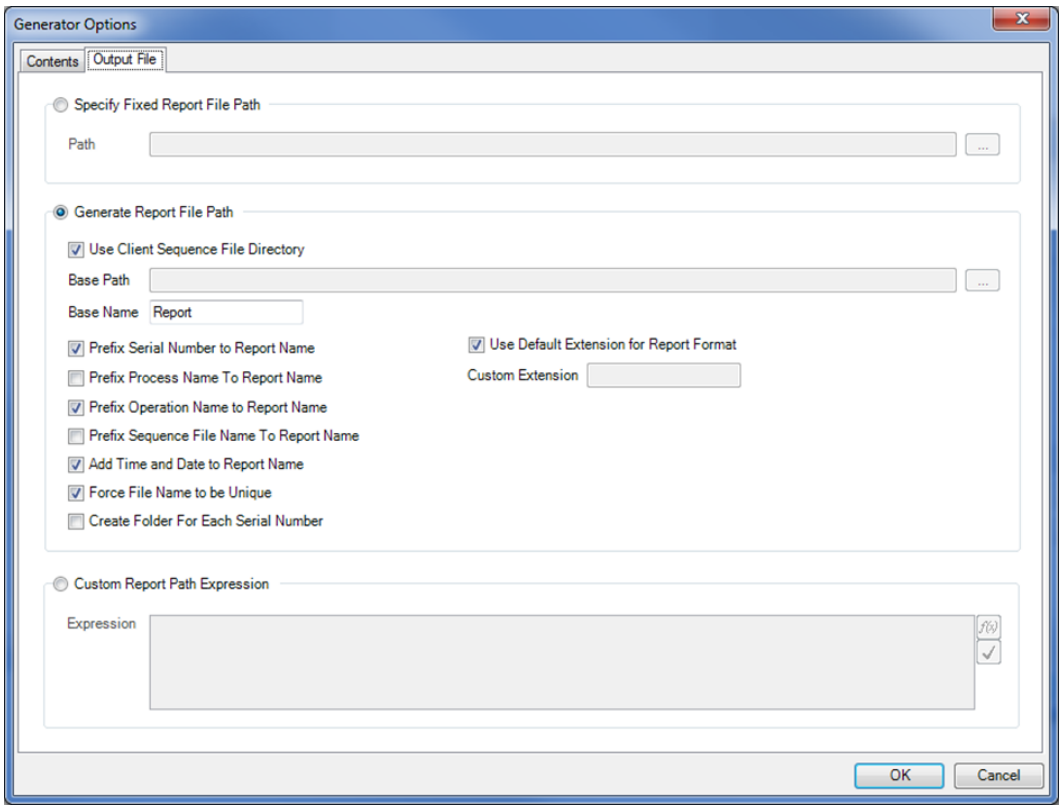
Checking the “Display Using Default Viewer” option will cause the report to be displayed at the end of an execution, in addition to the execution summary displayed in the report view.

Figure 14: Generator Options Contents PDF



Using the “Output File” tab, it is possible to configure the target path in a similar fashion to the NI report generators – as shown in Figure 15.

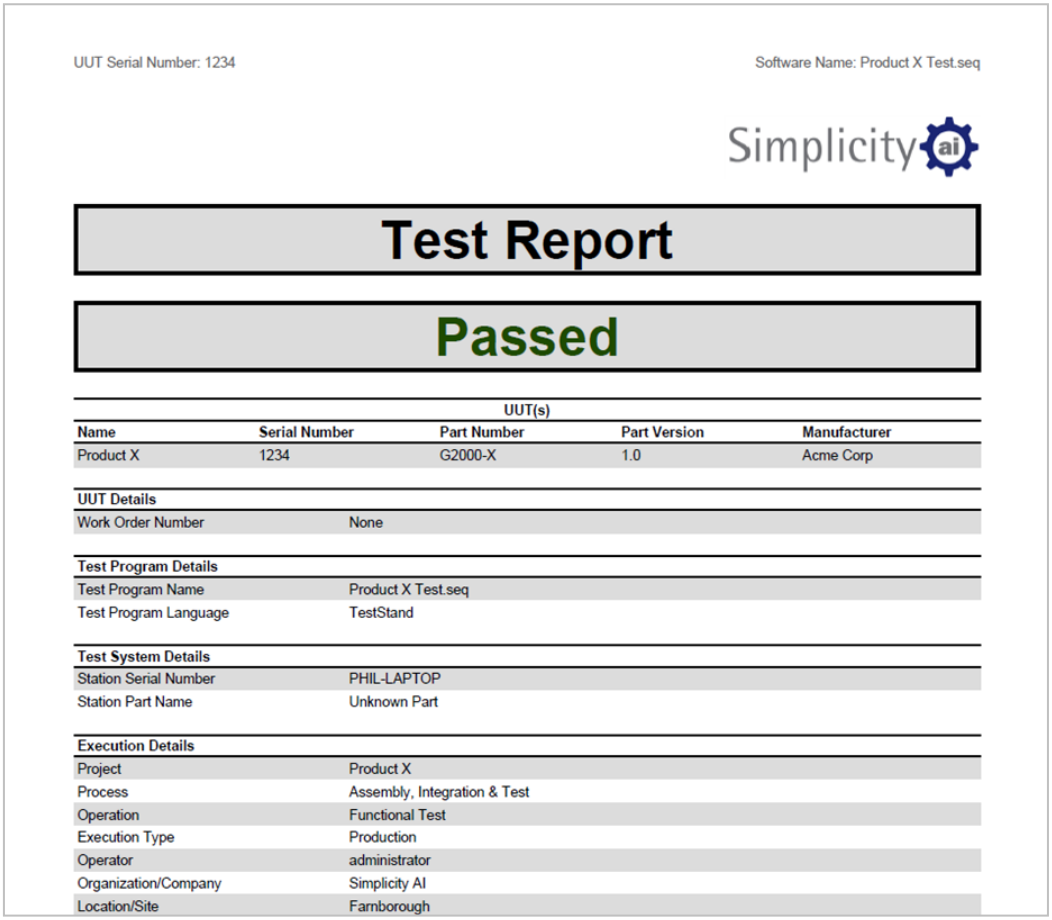
Figure 15: Generator Options Output File



Modifying Reports

The Tequra Framework Process Model provides a number of ways to configure output reports. An example PDF report is shown in Figure 16, which has been customized.

Figure 16: Example PDF Report



UUT Serial Number: 1234

Software Name: Product X Test.seq

Simplicity ai

Test Report

Passed

UUT(s)				
Name	Serial Number	Part Number	Part Version	Manufacturer
Product X	1234	G2000-X	1.0	Acme Corp

UUT Details	
Work Order Number	None

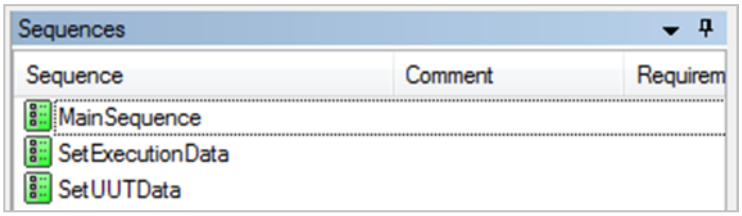
Test Program Details	
Test Program Name	Product X Test.seq
Test Program Language	TestStand

Test System Details	
Station Serial Number	PHIL-LAPTOP
Station Part Name	Unknown Part

Execution Details	
Project	Product X
Process	Assembly, Integration & Test
Operation	Functional Test
Execution Type	Production
Operator	administrator
Organization/Company	Simplicity AI
Location/Site	Farnborough

The logo graphic, “Organization/Company”, “Location/Site” and “Area/Line” can all be configured using Configure>>Model Options (Location tab). Other fields are specific to the test sequence being executed so can be configured by overriding various Process Model callbacks – open the sequence file to be customised then select Edit>>Sequence File Callbacks and enable “SetExecutionData” and “SetUUTData”. This will add two sequences to your sequence file as shown in Figure 17.

Figure 17: Report Customization Callbacks



In the “SetExecutionData” sequence it is possible to override information about the execution by customizing the expression in the “Set Execution Data” step.

Table 2: Execution Information

Category	Item	Description	Default Value
Project	Name	The project to which the UUT belongs	“Unknown Project”
Process	Name	The overall manufacturing process name	“Unknown Process”
Operation	Name	The name of the operation/ test stage within the manu-facturing process	“Unknown Operation”
Station	Serial Number	Serial Number for the test station	The network name of the test sta-tion PC.
	Part Name	The part name/model for the test station	“Unknown Part”
	Part Number	The part number for the test station	<Optional>
	Part Version	The specific version of the test station part	<Optional>
	Manufacturer	The manufacturer of the test station	<Optional>
Software	Name	The part name/model for the test station	The name of the client sequence file
	Version	The version of the test soft-ware	<Optional>
	Language	The language that the test software was authored in	“TestStand”
	Limits Version	The version of test limits used by the test software	<Optional>
Operator	Username	The username of the test operator	The username of the logged in TestStand user
Miscellaneous	Execution Type	The type of execution. Valid values are “Debug”, “Diagnostics” or “Production”	“Production”
	Persist Results	Reserved for future use	True
	Is Partial	Flag to determine whether a run is a partial execution (also known as “Test Selec-tion”).	False

In the “SetUUTData” sequence it is possible to override information about the Unit-under-test by customizing the expression in the “Set UUT Data” step.

Table 3: UUT Information

Item	Description	Default Value
Serial Number	The serial number of the UUT	Value from Operator entry in the dialogue box shown during the PreUUT callback
Part Name	The part name/model for the UUT	“Unknown Part”
Part Number	The part number for the UUT	“Unknown Part Number”
Part Version	The specific version of the UUT	<Optional>
Manufacturer	The manufacturer of the UUT	<Optional>
Work Order Number	The work order number to which the UUT belongs	“None”

Additional Information may be added to report header by calling sequences in `<TestStand Public>\Components\Models\PK-507374_Process_Model\Logger Support.seq`

This provides the ability to add information to any of the tables (Execution Attributes) as well as adding other relevant information:
Note that these support sequences can only be called from within

Table 4: Execution Attributes

Execution Attribute	Any custom string, number or Boolean to report header
Execution Note	A simple string which will appear in the “Execution Notes” table, including a timestamp of when the note was added.
Equipment	Test equipment details, including the ability to specify serial numbers and calibration dates. These items will appear in the “Equipment” table.
Child Unit	Child Unit/Sub assembly information. For electronics testing this could include PCB components, power supplies etc. These items will appear in the UUT(s) table at the top of the report.
Related Software	Related software information, this could be development environment versions, driver version etc. These items will appear in the “Related Software table”.

PreMainSequence Callback or MainSequence Callback, as the logger reference will not be valid before this point.

In the event that completely custom reports are required, Simplicity AI can develop these to customer requirements on a fixed-cost basis. The reporting framework is completely modular so it is simply a case of adding a new generator DLL to the Process Model directory, this would not involve a custom build of Tequra Framework.

Test Steps

Overview

NI TestStand provides a set of Step Types which facilitate many common tasks, such as checking a numeric value against limits. However, there are some limitations making it necessary to write custom code for certain test scenarios. Tequra Framework includes an enhanced set of test steps including waveform testing, enhanced string testing and the ability to add data tags to specific results, which is particularly useful for characterisation testing. These steps are designed to be drop-in replacements for the NI steps, such that it is possible to change the type of a step to an NI version to the equivalent Tequra Framework version, with no loss of configuration data.

All Tequra Framework steps provide the ability to define a Test Name expression, a Measured Parameter expression, a Test Point expression, Test Parameters (test conditions) and Meta Data. Full documentation for each step type can be found by opening the Edit window and selecting “Help”, as shown in Figure 18. Certain steps also support defining multiple properties from a “configuration data” variable; this is to provide integration with Tequra Requirements – allowing for easy passing of parameters from requirements files, as well as automatic code generation. All Tequra Framework Test steps provide excellent compatibility with Tequra Analytics, allowing data to be passed to a server for aggregated analysis and archiving.

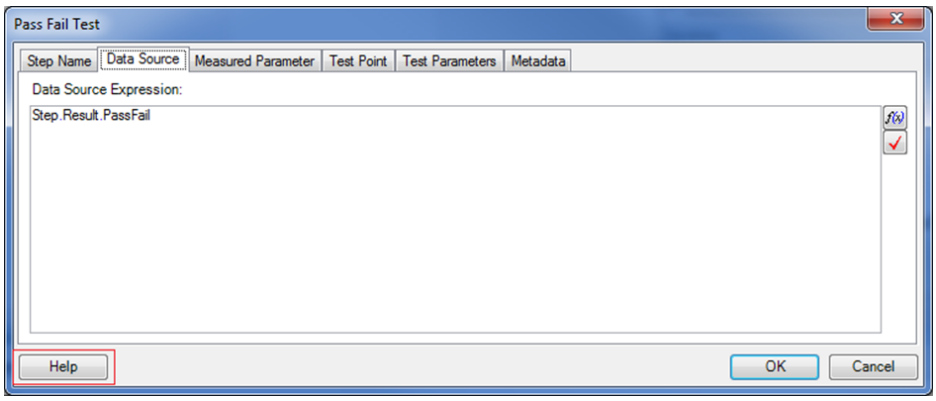
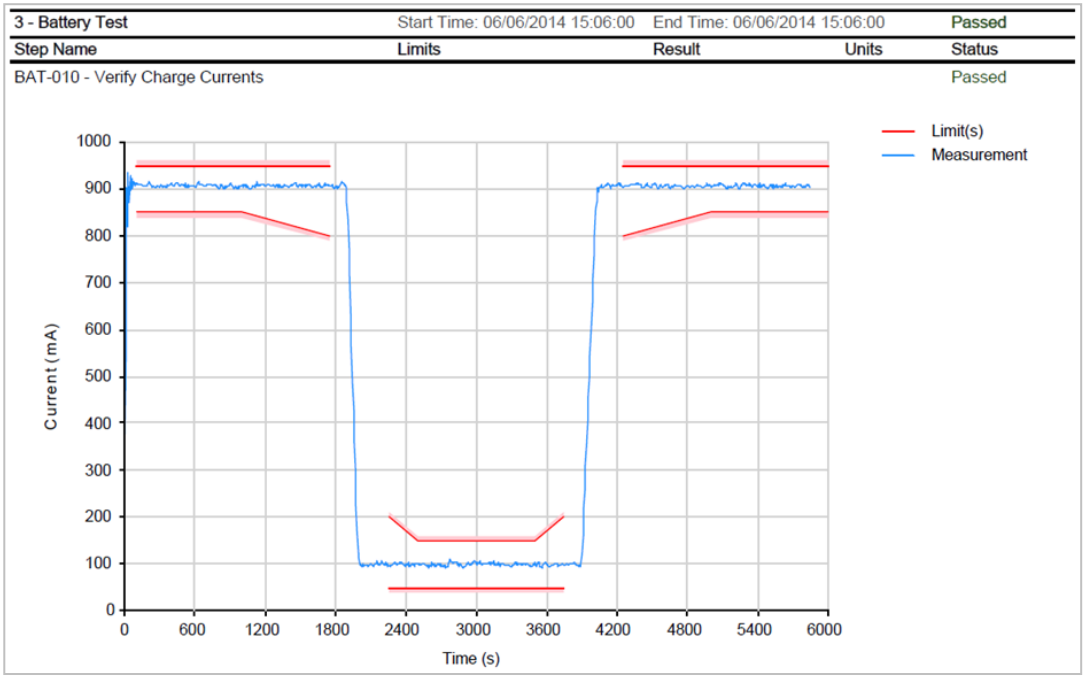


Figure 18: Pass Fail Test Edit Dialog

Table 5: Test Steps

Tequra Framework Step	NI Equivalent Step	Description
Pass Fail Test (Extended)	Pass Fail Test	Similar functionality to NI step, but adds standard features listed above.
Boolean Limit Test	-	No NI equivalent – this provides the ability to compare Boolean values (Equal To/Not Equal To True/False). This can provide a clearer output that Pass/Fail step.
Numeric Limit Test (Extended)	Numeric Limit Test	Enhanced version of the Numeric Limit Test Step, which provides a simpler mechanism to set number of decimal places and numeric base. This also truncates values to the specified number of decimal places before comparing against limits to prevent failures caused by the mismatch between display precision and numeric representation.
String Limit Test (Extended)	String Value Test	Enhanced version of the String Value Test with many more comparison types: Equals, Does Not Equal, Begins With, Ends With, Contains, Does not Contain and Regular Expression – all of which can be set to be case sensitive or case insensitive.
Waveform Limit Test	-	No NI Equivalent – this provides the ability to compare a single trace against limits. Limits may be scalar values which apply to the entire trace, or segmented limits which apply to particular sections. Examples of PDF report rendering of various Waveform Limit Tests are shown in Figures 19-21.

Figure 19: Waveform Limit Test Result (PDF Report)



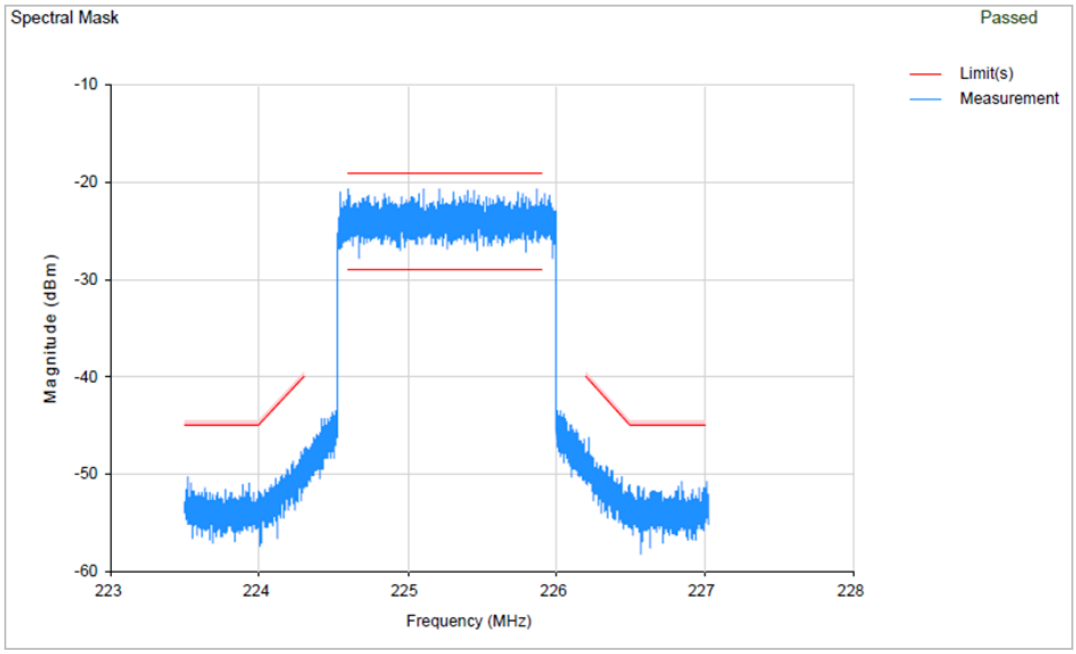


Figure 20: Waveform Limit Test Result (PDF Report)

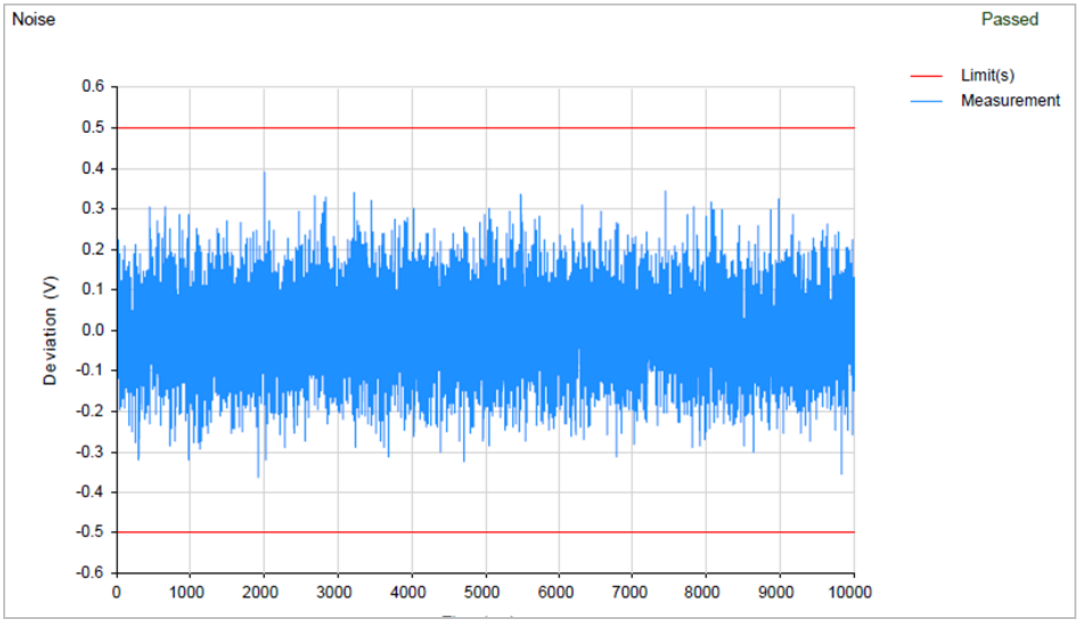


Figure 21: Waveform Limit Test Result (PDF Report)

Flow Control Steps

Step Selection

A common requirement in advanced manufacturing scenarios is for skilled operators to be able to perform partial tests on UUTs, rather than have to run complete test procedures which may be too time consuming to identify specific failures. A typical use case would be a verification check of a particular subset of functionality after a rework operation. This may need to occur multiple times for the rework operation to be successful, after which a complete run may be performed before the unit is returned to the main manufacturing process.

The Tequra Framework Step Selection step is added to the “Flow Control” section of the insertion palette. The step is used in a similar fashion to other Flow Control steps whereby other steps sit between the step and its associated “End” step, as shown in Figure 22. Typically, these steps would be sequence calls which perform particular groups of tests; however the step will work with any step type.

Figure 22: Step Selection Example

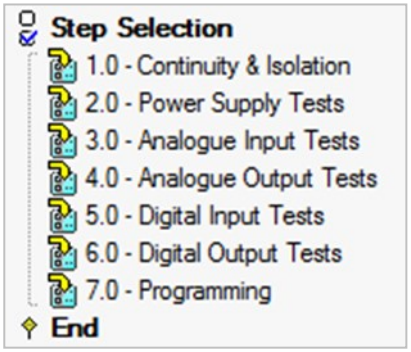
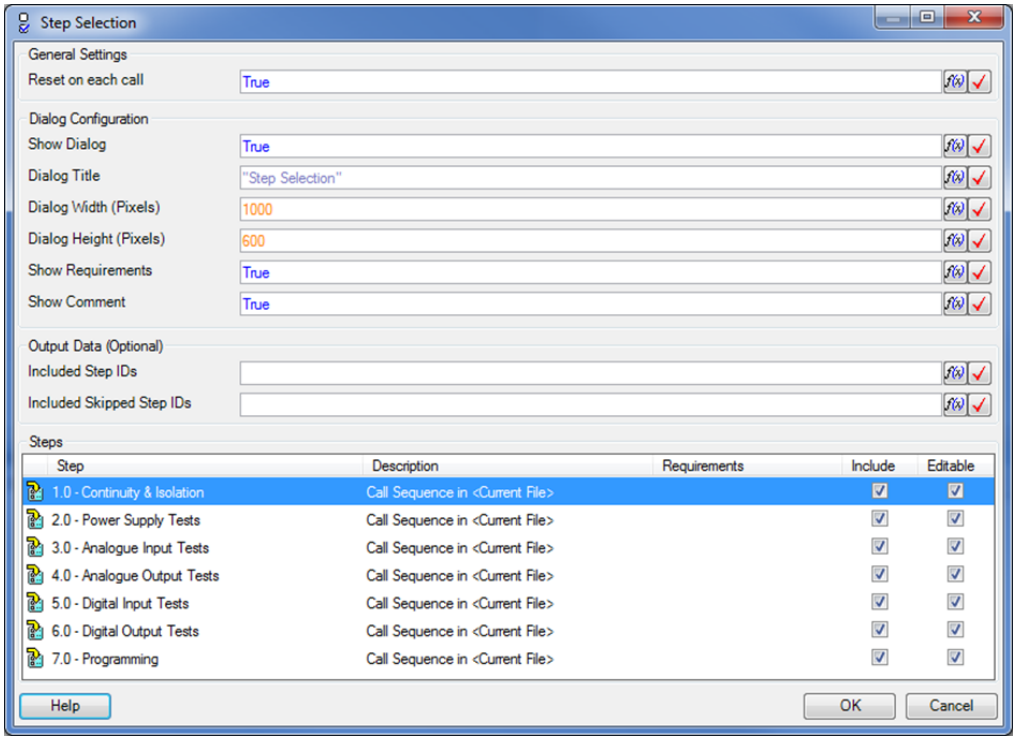
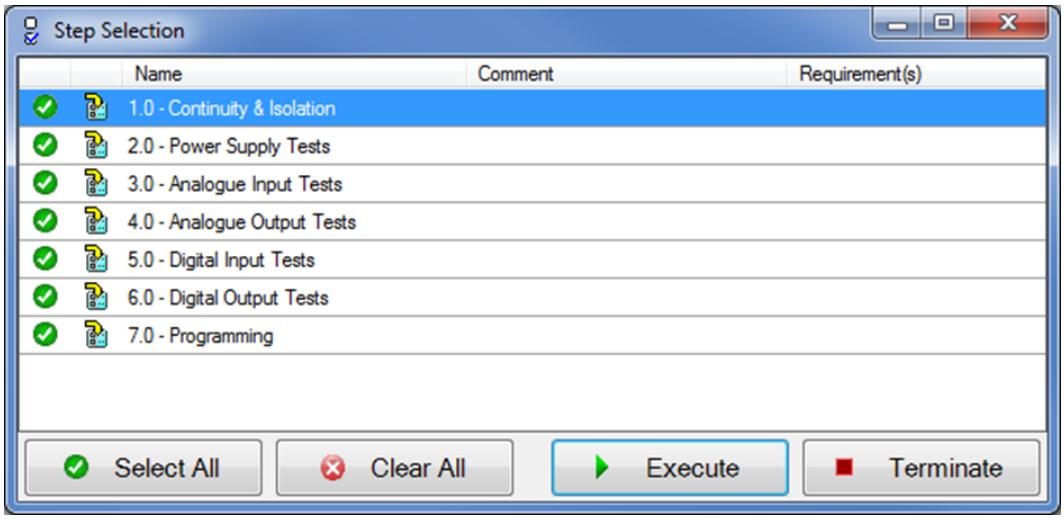


Figure 23: Step Selection Edit Dialog



At run-time, a simple window will be displayed to the operator giving the option to skip particular steps before executing or terminate the sequence. An example is shown in Figure 24. Steps displayed in the dialog will match those which have the “Include” flag set in the configuration. Those with “Editable” set to false cannot be modified by the operator.

Figure 24: Step Selection Run-Time Dialog



The step selection step does not modify the sequence file and will restore default value for each run of an execution.

User Interface

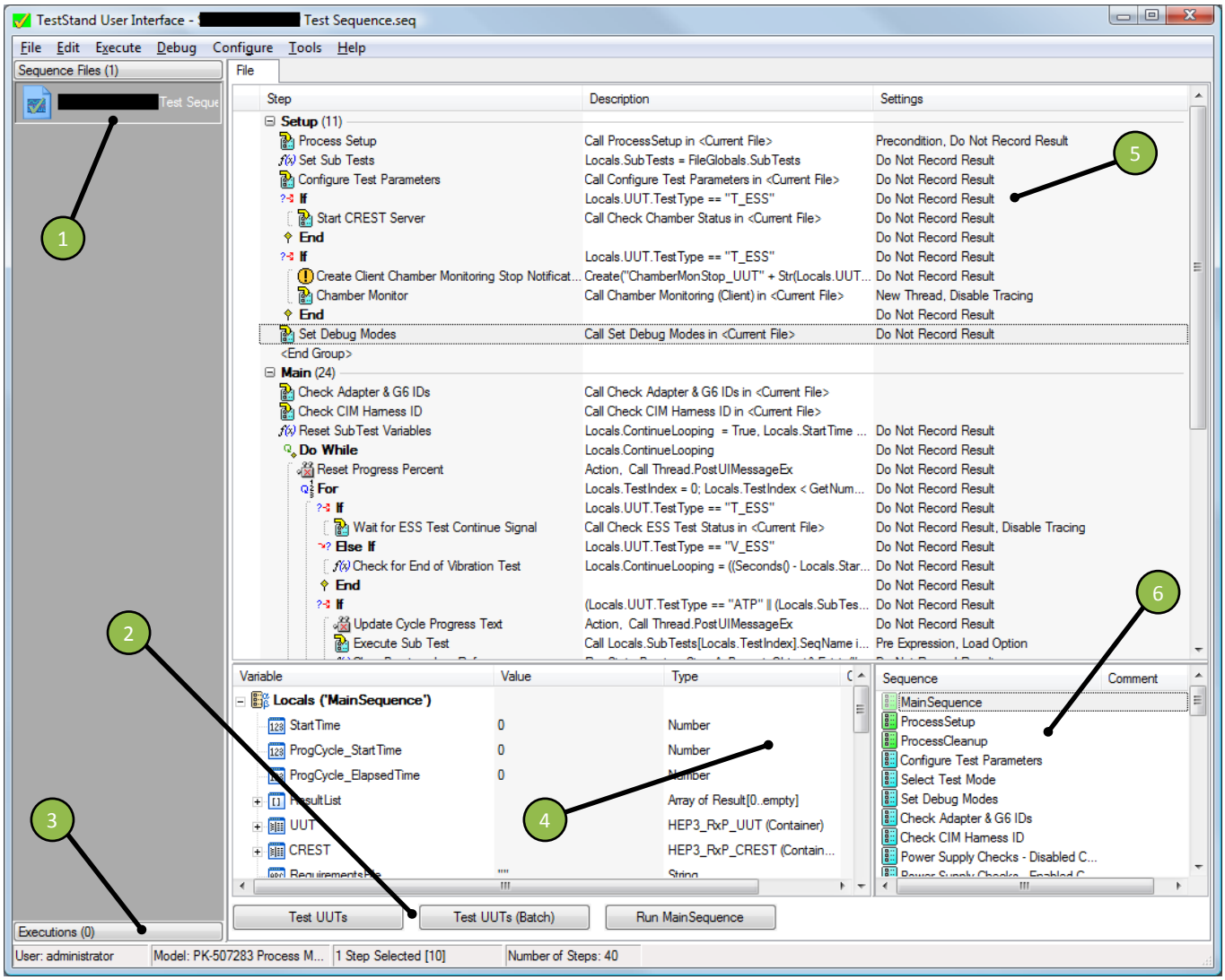
Overview

Typically, the TestStand Operator environment is provided through one of the standard NI Operator Interfaces. As source code is provided for these, it is possible to customize the interface to match specific requirements. Tequra Framework adds a number of attractive features missing from the standard interfaces meaning that customers no longer need to make specific modifications.

The Tequra Framework Operator Interface is installed to `<TestStand Public>\UserInterfaces\PK-507396_Operator_Interface\PK-507396_Operator_Interface.exe`. To allow easy access for operators, it is recommended to create a desktop shortcut to this executable.

Operator Interface—Sequence File

Figure 25: Operator Interface with Sequence File Loaded



1) Sequence file list
2) Execution entry point buttons for process model
3) Execution view tab
4) Variables view
5) Selected sequence file display
6) Sequences file view

Sequence File List

This shows a list of which sequence files are currently loaded. Unless a sequence file is specifically closed, this maintains a history of sequence files executed, so that they are available when the Operator Interface is opened. If the desired sequence file is not loaded when the application starts, it can be opened by using the File>>Open Sequence File option.

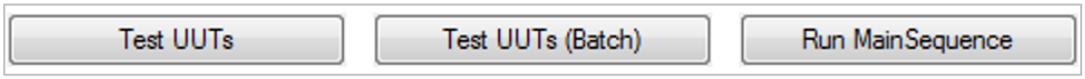
When a sequence file is selected, the contents of the sequence file are shown in the "Selected Sequence File Display". This is for information only and does not provide any useful information for operators.

By toggling the "Executions" view tab and "Sequence Files" view tab buttons it is possible to switch between File view and Execution view modes.

Execution Entry Point Buttons

To start executing a sequence, the Execution Entry Point buttons can be used. When an execution is started, the display automatically switches to the Execution View.

Figure 26: Execution Entry Point Buttons



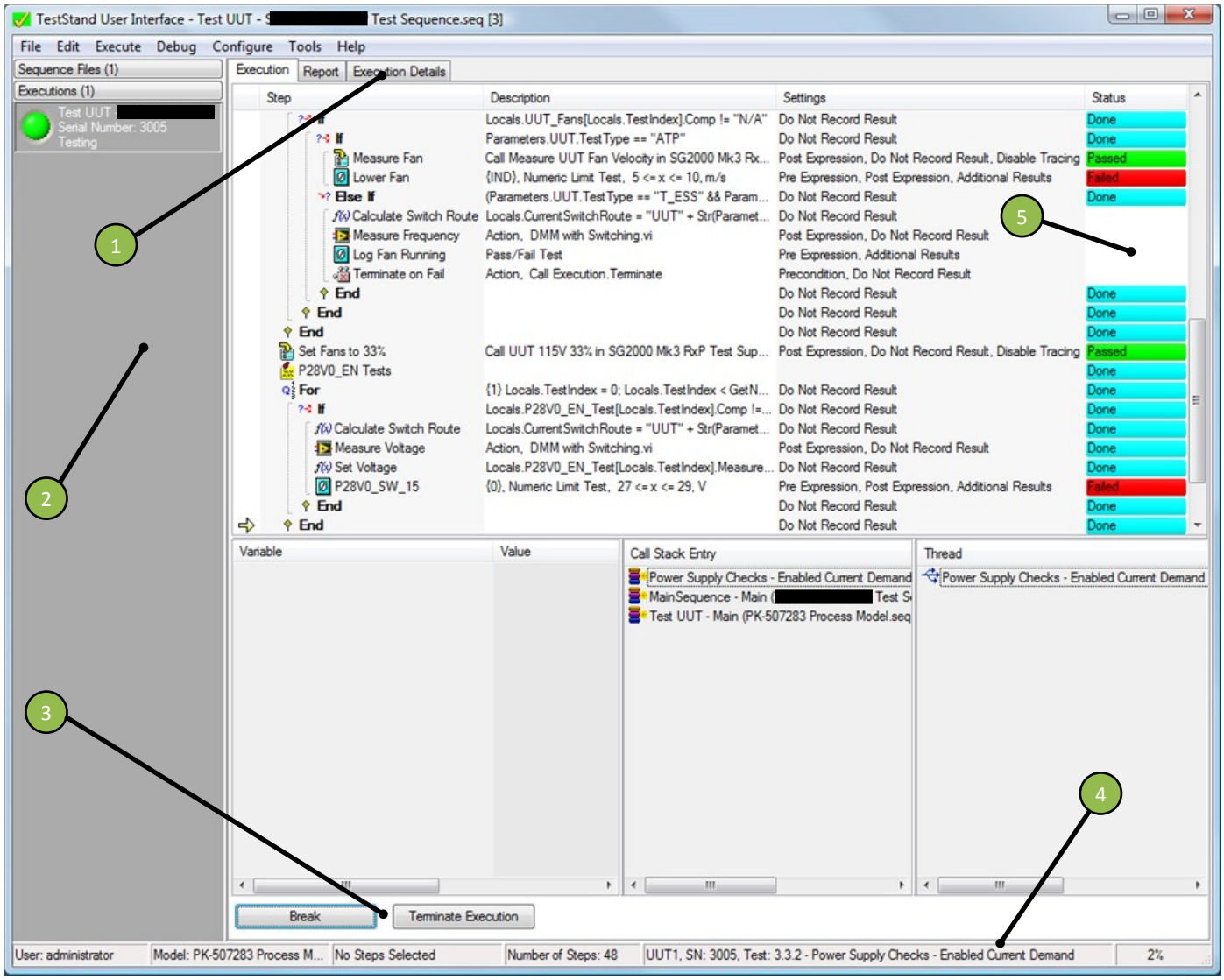
Test UUT
Tests a single UUT, logging all results to a report file.

Test UUTs (Batch)
Tests multiple UUTs in parallel, define the number of test sockets using Configure>>Model Options. This option requires that the test sequence can support parallel execution.

Run <Sequence>
Run the current selected sub sequence within the sequence file. Results will not be logged to a report file. This is here for compatibility purposes and is not designed to be used by test operators during testing.

Figure 27: Operator Interface Sequence File Executing (execution Tab)

Operator Interface—Execution Tab



1) View tabs
2) Execution list
3) Execution control buttons
4) Status/progress bar
5) Execution progress

Execution List

This shows a list of currently active executions (sequence files that are running). Generally there will only be one item active in the list, the currently running sequence. However, if the Test UUTs (Batch) Entry Point has been selected, multiple executions will be shown here (one for each UUT being tested). By selecting the executions in the list, the Execution Progress changes to show the current execution.

By toggling the “Executions” and “Sequence Files” buttons it is possible to switch between File view and Execution view modes.

View Tabs

These tabs allow the operator to change the Execution View:

- Execution: Trace View showing all test steps and the step status as they are executed
- Report: Available for compatibility purposes; however in the current rig configuration this will not display anything until the end of execution
- Execution Details: Updated result summary and Pass/Fail Yields throughout execution

Execution Progress

This window shows the currently selected execution, showing progress as the sequence executes test steps. This is useful to show activity since there may be long delays between logged results if only monitoring the Execution Details tab.

Execution control Buttons

The execution control buttons allow the operator to Pause, Resume and Terminate a sequence in progress.

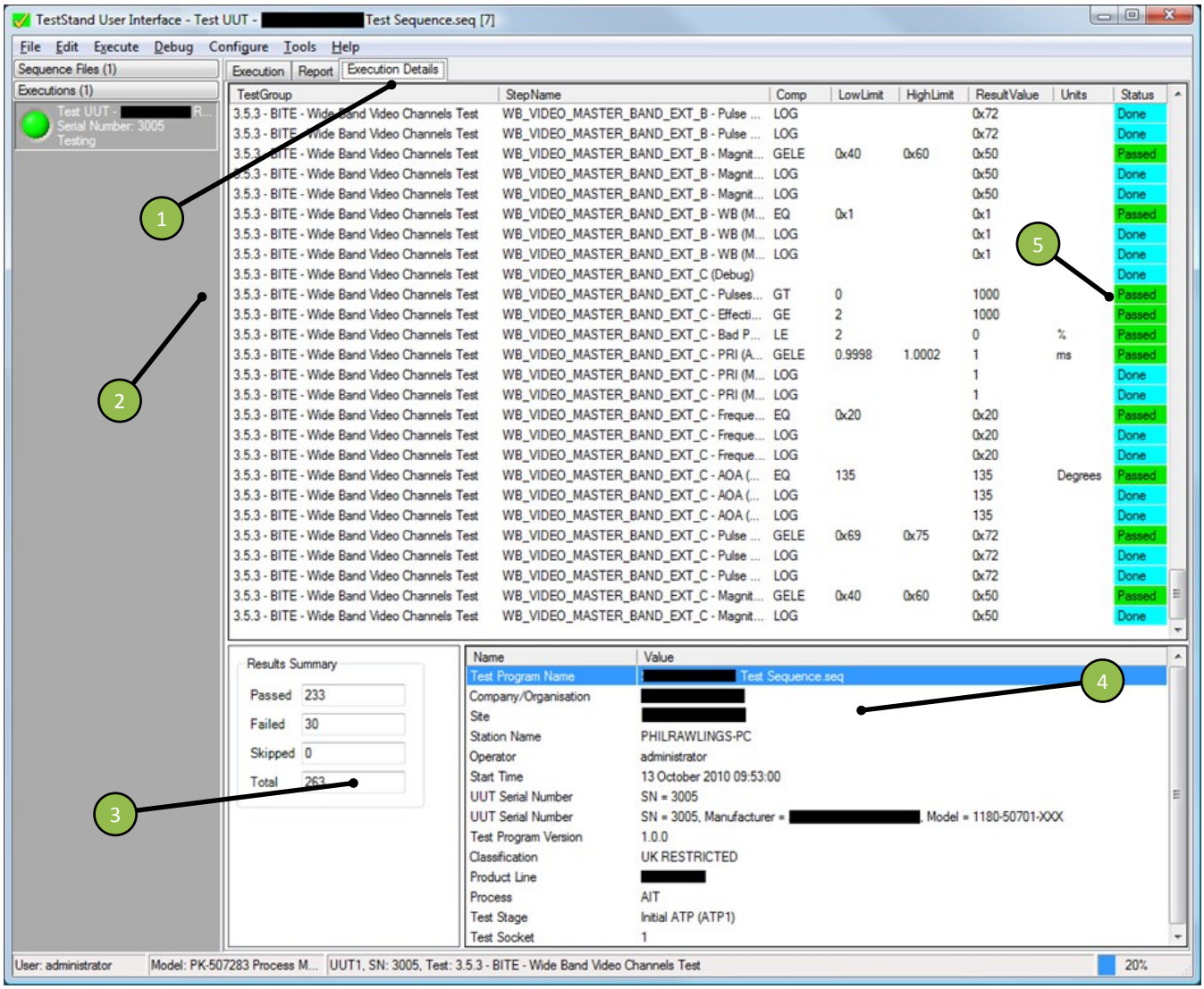
- Break – Pauses the execution
- Resume (Only Visible when an execution is paused) – Resumes a paused execution
- Terminate Execution – Performs a graceful shutdown of the test sequence (closing resources etc.) and marks the result as “Terminated”

Status Bar/Progress

For sequences that supply status information – the status bar can show progress test and a 0-100% progress indicator. This can be updated using TestStand UI Messages for “Progress Percent” and “Progress Text”.

Figure 28: Operator Interface Sequence File Executing (Execution Details Tab)

Operator Interface—Execution Details Tab



1) View tabs
2) Execution list
3) Result yields
4) Additional information
5) Logged results status

Result Yields

This pane shows a summary of Passed, Failed and Skipped results, the yield values are updated as the test progresses.

Execution Progress

This window shows the currently selected execution, showing results acquired as the sequence executes test steps. This shows the following columns:

- Test Group – Sub Test/Section Name
- Test Name – Type/Name of Measurement
- Comp – Comparison Type (generally for Numeric Limit Tests)
- Low Limit, High Limit – Limits values for the test steps
- Result Value, Units – Measured value and associated units
- Status – Result of the test step (Passed/Failed/Done/Error)

Note: to prevent the system running out of resources, this display only shows the last 1000 results for the selected execution. If the operator wishes to view earlier results, they will need to wait until the report is generated at the end of the test. The view window will auto-scroll such that the newest results are added to the bottom of the displayed list.

Additional Information

Summary information about the test in progress is presented in this area (such as Station Details, Operator Information, Test Program Properties, UUT Data etc.). Each property for additional information is named with the value shown in the adjacent column.

The Additional Information section can be populated using the Set Execution Details callback, send information as an object reference to the activeXDataParam parameter. To view an array of data in this field, with the item name on the left and the value on the right, create an array, each element should be named as the property name, and populate the elements with values to display.

Further Information

Product Web Page

www.simplicityai.com/tequra-framework/

Contact Us

Telephone: +44 (0)845 468 2886 or +1 877-717-1190

Email: info@simplicityai.com

Support

Various support, training and development options are available, please contact us for more information.