TestPro

The Testing Professionals

TAF PRO

User Guide

Version 4.3

This is a confidential document and is for restricted circulation only to the members of TestPro Pty Ltd

All other circulations are only on request subject to approval by TestPro.

DISCLAIMER

The authors have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No Liability is assumed for incidental or consequential damages in connection with or arising out of the use the information or instructions contained herein.

Copyright

Contents

1 Introduction

1.1 TAF Pro User Guide

The TAF Pro User Guide is intended to provide Test Analysts with guidance on how to use TAF Pro to improve productivity by reusing test scripts and automating the execution of testing variations through use of TAF Pro Execution Suites. The User Guide provides information on the product features used every day in preparation and execution of testing scripts, scenarios, data and execution suites.

1.1.1 Audience for the User Guide

The primary audience for the user guide are the testing team members responsible for the development of test scenarios, the development of test data and the execution of Execution Suites.

Other documentation in the TAF Pro distribution provides information for System Administrators or those developing testing scripts.

1.2 Other TAF Pro Documentation

Addition information about the TAF Pro framework is provided in the following documents:

1.2.1 Installation and Configuration Guide

The Installation guide provides information on:

- System pre-requisites to run the TAF Pro software;
- Installing and configuring TAF Pro;
- TAF Pro licences and licensing requirements;
- Creating and configuring User accounts within TAF Pro;
- Linking TAF Pro to the selected scripting and test management tools, such as Rational Quality Manager.

1.2.2 Scripting Guide

The scripting guide provides information for test-script developers to allow them to write and adapt testing scripts that can be integrated with TAF Pro. Scripting is primarily a function of the Test Automation Tool and the documentation provided with the selected tool should be referenced. The TAF Pro Scripting Guide provides information on the functionality within the TAF Pro interface relevant to the development and integration of test scripts.

1.2.3 Release Notes

Release Notes are provided with each software version to provide users with information on changes to the product since the last release.

2 Introduction to TAF Pro

TAF Pro is a hybrid automated testing framework that combines elements of data-driven and keyword-driven frameworks and adopts a role-based approach to test development and execution.

The key objectives of an automation framework are:

- To enable reuse and sharing of automation scripts to improve efficiency and scalability;
- To reduce the test maintenance overhead;
- To provide a non-technical user interface for business users;
- To capture and leverage the domain knowledge of business users into a set of regression tests that can be executed repeatedly without requiring the business representative to be present.

The key functions within TAF Pro are:

- Execution of tests built with Test Automation Tools such as IBM Rational Functional Tester;
- Easing data handling challenges for data being used during testing;
- The linking of tests to form scenarios and suites of associated tests.

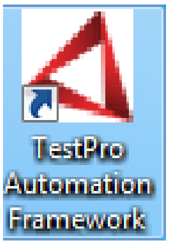
TAF Pro has been developed with a focus on:

- Minimizing the impact of the framework on script development;
- Approaching test execution from a business perspective (as opposed to purely technical testing);
- Providing flexibility in data association and selection.

2.1 TAF Pro Components

When TAF Pro is installed the following options are available from the Start Menu folder. Some options also available via desktop icons by default.

2.1.1 TestPro Automation Framework



This is the primary interface of TAF Pro. It is used for planning tests, configuring data, setting up and executing tests and viewing results. TAF Pro can be initiated by clicking on the desktop icon shown or by selecting from the **TestPro Automation Framework** menu option, **TestPro Automation Framework**.

2.1.2 Command Line Interface

This utility is used to enable direct execution of Execution Suites using batch files linked to Ant.

2.1.3 TAF Pro Configuration Tool

The configuration tool is used to associate the TAF Pro instance with the database

2.1.4 Database Configuration

Information on configuring TAF Pro with a database is contained in the TAF Pro *Installation and Configuration Guide.*

2.1.5 RFT Project Setup

Information on configuring TAF Pro with a database is contained in the TAF Pro *scripting Guide.*

2.1.6 Support

The support utility provides a pro forma for completing support requests. Access to this utility is also available from within the TAF Pro workspace GUI.

2.1.7 View Log

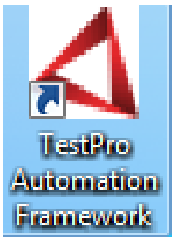
The View Log utility maps to the system log directory providing the facility for the user to open and view current and recent logs created by TAF Pro.

2.1.8 Uninstall TestPro Automation Framework

This utility uninstalls TAF.

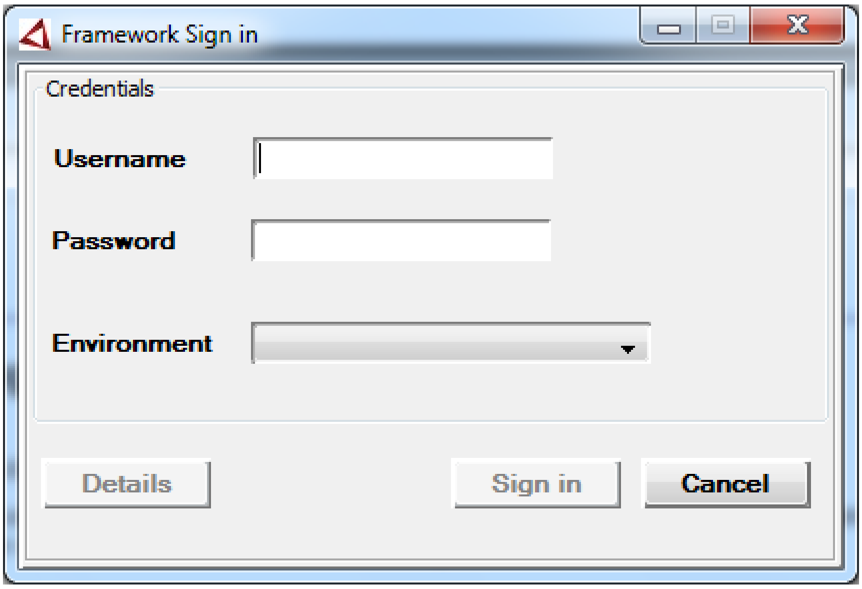
3 Launching TAF Pro Automation Framework

TAF Pro can be launched from the Windows Start Menu at **All Programs ➔ TestPro Automation Framework ➔ TestPro Automation Framework,** or by clicking on the TestPro Automation Framework icon on the desktop.



3.1 Sign in

Once the TAF Pro Automation Framework is launched all users are required to sign in by entering their Username and Password in the Framework Sign In window. Once these are confirmed the user can select the environment they wish to use from the drop down list of environments available to them.

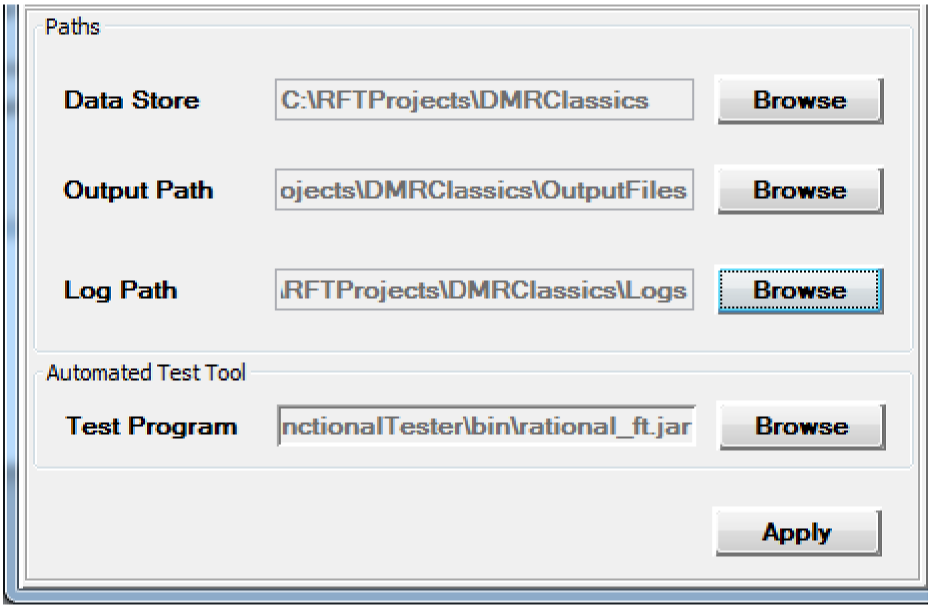


Enter the following details:

1. **Username** – Login name up to 15 characters
2. **Password** – User Password minimum 6 characters
3. Select the required environment from the drop down list.
    - 3.1. If this is the first time the user has accessed the environment the environment details must be configured by selecting Details and proceeding as described in the next section.
    - 3.2. Otherwise select Sign In

3.1.1 Configure Environment Details

Each user environment has four paths set for access and storage. These will be specified when the user first connects to the environment. The paths can be local or networked, however in a multi-user situation the Data Store path needs to be networked to provide access for other users. The four paths are shown below.



Paths

Data Store: C:\RFTProjects\DMRClassics

Output Path: ojects\DMRClassics\OutputFiles

Log Path: \RFTProjects\DMRClassics\Logs

Automated Test Tool

Test Program: nctionalTester\bin\rational_ft.jar

The TAF Pro Administrator of your organisation can provide you with the details of the environment you will connect to.

Data Store: This is the location of the TAF Project containing the test scripts used for testing in this environment. The path should point to the Project level folder that would be shown as the primary container in the automation tool.

Output Path: This is the root level folder for all output from this environment. In a small project this may be a single folder under the project directory as shown. In a larger project separate folders can be used for different environments and users.

Log Path: This is the root level folder for all logs from this environment. In a small project this may be a single folder under the project directory as shown. In a larger project separate folders can be used for different environments and users.

Test Program: This is the path to the application used for developing scripts. The example shown is Rational Functional Tester.

Once the Paths have been configured click **Apply** and then select **Sign In** from the top half of the panel.

3.1.2 The Default Environment

The Default environment paths point to standard paths that are used for system-wide logs and output files. These paths should not be used for users' stand-alone environments.

4 The TAF Pro Interface

This chapter provides functional advice for navigating and using the TAF Pro Interface.

- In the following text, TAF Pro Window names, button names and drop down items will be in **bold** type.
- Information text will be in plain type.
- Text entered by the User, including menu item and button selections, as well as response text made by TAF Pro will be in ***Bold and Italics.***

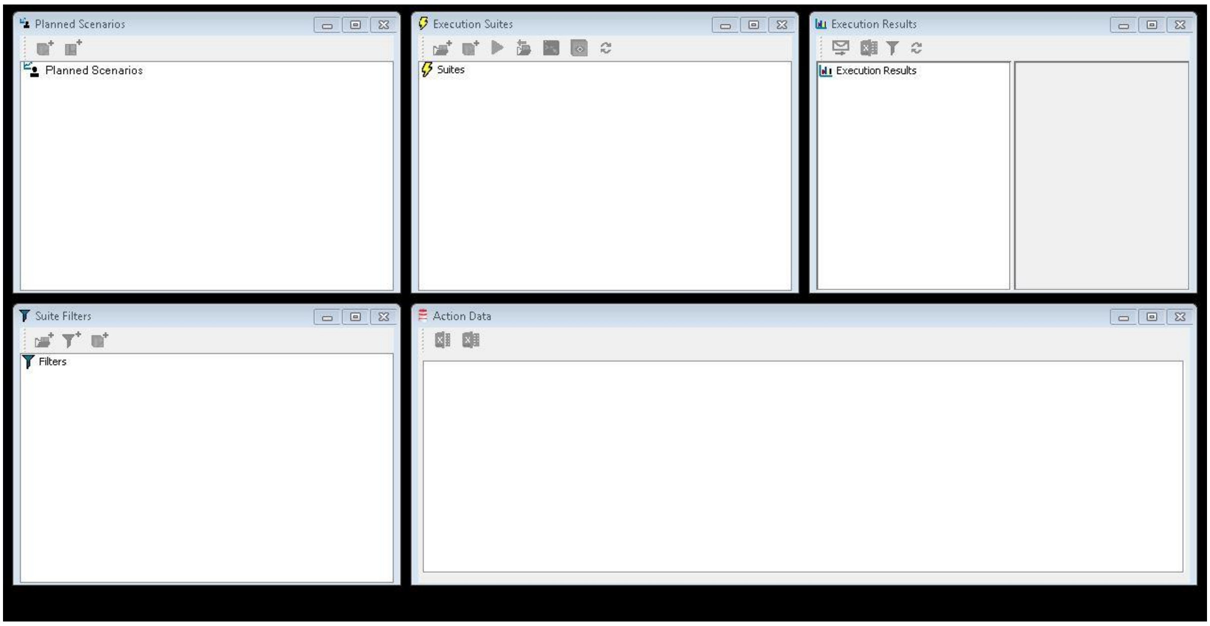
The sub- sections in this chapter provide the following information:

- Section 4.1 provides a summary of the main items in the TAF Pro interface including a description of the Menu Functions;
- Section 4.2 lists Mouse operations to enable the user to navigate the TAF Pro interface;
- Section 3.3 provides a list of the Menue Bar options in TAF Pro.

A detailed description of all the Menu Functions is provided in Section 15.2 – Appendix B.

4.1 Main Window

The Main Window consists of a frame containing the five functional panels in TAF Pro. The frame contains the Menu Bar, Key Functional Icons and the Status bar. The central workspace contains the five functional panels.

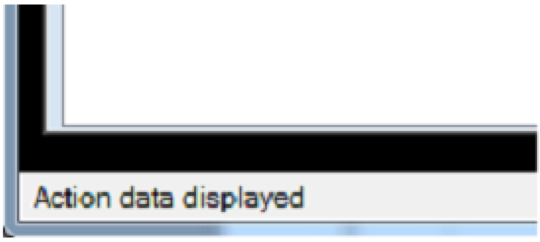


4.1.1 Functional Panels

The five functional Panels in TAF Pro are used to focus on the main activities:

Panel	Description
Planned Scenarios	Planned Scenarios are used to organise Actions into Test Cases and Scenarios that can be combined to create Execution Suites. A key element of getting the most out of TAF Pro is creating Actions that are reusable. Planned Scenarios contain one or more Test Cases and Test Cases contain one or more Actions. A Scenario or Test Case without any content is considered incomplete.
Execution Suites	The Execution Suites panel is used to create and execute Execution Suites. Execution Suites consist of one or more Planned Scenarios containing Test Cases and Actions. Each Action in an Execution Suite needs to be linked to a Data Group which will provide the data required to determine the test completed. An Execution Suite can be executed when all Scenarios are complete and all Actions are linked to Data Groups.
Execution Results	Displays the results available after running an Execution Suite. The user can drill down to review the input and output data for each line of the test as well as seeing any system messages sent from the Test Scripts. Results are colour coded green for success, red for failure and orange for a warning.
Suite Filters	The filters that have been defined for the Execution Suites are catalogued in this window. They are grouped under the Execution Suite name. Filters are used to create subsets of the Data Group lines during Test Execution.
Action Data	The Action Data panel displays the data contained for the currently selected Data Group or Action. Selection of the Action or Data Group can occur from several places in TAF Pro. The data shown will either be the first few lines, all data in the Data Group, or the data corresponding to a filter if the filter is selected.

4.1.2 The Status Bar

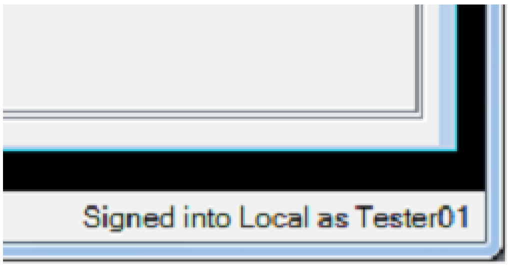


Action data displayed

The Status Bar is located at the bottom of the workspace. It provides two key sets of information for TAF Pro users

- ***Progress Status*** (Status Bar, Left Bottom corner) which gives notification of progress and outcome of actions performed.

- ***Environment Details*** (Status Bar, Right Bottom corner) which show the user logged in and the environment accessed for the session.



Signed into Local as Tester01

4.1.3 Key Functional Icons

TAF Pro provides functional Icons for both the workspace and the individual workspace panels. Icons are opaque when they are not available for use. Icons become active when the relevant tree node is selected within the functional panels.

To identify the function of an icon, place the mouse on the icon and the icon description will be displayed.

4.2 Navigating the Interface

This table below summarises the main mouse click actions required to select and activate TAF Pro functions:

Mouse Operation	Mouse Function
Double click on the navigation tree in any window	Expands a collapsed tree item to display subordinate items. Collapses an expanded tree item at the level selected.
Right click on a highlighted item in any TAF Pro window	Displays a list of operations available on the selected item
Left click on a highlighted available operation	Executes the function associated with the highlighted operation
Control Click	Selects multiple items in a table list displayed from any TAF Pro window
Drag and Drop	Select the item and hold down the mouse to move items within panels where appropriate. TAF Pro uses logic relating to its own functions to prevent users dropping items inappropriately.
Resizing Windows	TAF Pro incorporates standard Windows functionality to resize any active window. Individual panels can be minimized and the remaining windows reconfigured to meet the users current requirements.
Maximise, Minimize , Restore	TAF Pro uses the standard maximise, minimise and restore buttons allowing users to modify the workspace to their requirements. Double clicking anywhere on the window Menu bar switches the window between the maximise and the restore display.

4.3 Menu Bar Options

The **Menu Bar** displays the Menu Items as outlined in the table below. The table shows the menu name and description and any sub-menu structure. These will be explained in detail in subsequent sections of the User Guide.

Main Menu	**Sub Menu**	**Description**
File (Alt+F)	Import → Scenarios	Import Planned Scenarios from Spreadsheets
	Import →Test Cases	Import Test Cases from Spreadsheets.
	Export → Scenarios	Export Planned Scenarios to Spreadsheets.
	Export → Test Cases	Export Test Cases to and from Spreadsheets.
	Exit	Exit TAF Pro
View (Alt+V)	Scripts **(Alt – V – R)**	Provides the facility to view a list of actions grouped by application, and the script in which they are written. The full path of the script within the automation tool project is provided. Clicking on the path shows the associated Data Groups which can then be viewed to show the data.
	Scenarios **(Alt – V – S)**	Displays the list of Scenarios currently planned in this users environment
	Test Cases **(Alt – V – T)**	Displays the list of Test Cases currently created in this users environment
	Actions **(Alt – V – A)**	This view shows the Actions, and the test cases, scenarios andExecution Suites in which they are used.
	Data Groups **(Alt – V – D)**	Shows the Data Groups, and the actions, test cases, scenarios and Execution Suites they are associated with. Selecting a Data Group allows the user to view the data and export it to a spreadsheet.
Data (Alt+D)	Create Data Template → Create Action Data Template	Create a Data Template for a single Data Group. The template is used to create a standardised input format for data.
	Create Data Template → Create Execution Data Template	Create a Data Template for an Execution Suite Data Group consisting of all the Data imported across an entire Execution Suite. The template is used to create a standardised input format for data.
	Import → Import Action Data	Import Action Data from a CSV or XLS file or from the central repository.
	Import → Import Execution Data	Import Execution Data from a CSV or XLS file or from the central repository.
	Export → Export Action Data	Export Action Data from an existing Data Group or the shared repository to a spreadsheet or to the shared repository.

Main Menu	**Sub Menu**	**Description**
	Export → Export Execution Data	Export Execution Data from an existing Data Group or the shared repository to a spreadsheet or to the shared repository.
Scripts (Alt+S)	Register	Registers a script from the Automated Testing Tool in this TAF environment. Registered scripts contain actions that can be used in Execution Suites.
	Convert Project	Converts an Automated Testing Tool project to include scripting information that is used by TAF Pro.
	Convert Script	Converts a single Automated Testing Tool script to include scripting information that is used by TAF Pro.
Window (Alt+W)	Planned Scenarios	These five options are designed to be used when the functional panels are maximised. They allow the user to switch between each of the panels.
	Execution Suites	
	Execution Results	
	Suite Filters	
	Action Data	
	Show All	These two options are designed to be used primarily when the functional panels are in the standard mode with several panels open. These two options will hide and restore all panels at once. In Full Screen mode the option operates on the open panel only.
	Hide All	
User (Alt+U)	Change Password	The current user can change their password.
	User Preferences	Users can set preferences
	Administration → SMTP Services Setup	The Administrator can set up access from TAF Pro to an SMTP mail server to allow Email to be sent from TAF Pro.
	Administration → Environments Setup	The Administrator can set up Environments and manage users' access to TAF Pro and the environments.
Help (Alt+H)	Documentation → User Guide	Links to a PDF copy of the User Guide.
	Documentation → Scripter Guide	Links to a PDF copy of the Scripter Guide.
	Documentation → Installation Guide	Links to a PDF copy of the Installation Guide.
	Documentation → Release Notes	Links to a PDF copy of the Release Notes
	Log an Issue	Provides a utility for logging issues with TAF Pro Support Team. This requires that email access has been set up.
	About	Presents License and Patent information about TAF Pro and its components

A detailed description of all the Menu Functions is provided in Section 15.2 – Appendix B.

4.4 Exit TAF Pro Application

To exit TAF Pro, either close the TAF Pro window or choose ***File →Exit.***

5 Quick Guide to Test Execution

This chapter provides a high level reference for all steps needed to successfully build and run an execution suite. A detailed description follows in the following chapters; the numbers in brackets following each item refer to the Section of the **TAF Pro User Guide** where this information can be found.

1. Create Automated Test Tool Script from initial recording:
    a. Ensure script is located in the test scripts;
    b. Ensure the super class for the new script is correct;
    c. Ensure the script uses the TAF Pro template.

Refer to the documentation for your Testing Tool for information on configuring the tool and developing scripts. Refer to the **TAF Pro Scripting Guide** for information on developing scripts that can operate within the TAF Pro framework.

2. Register the script in TAF Pro:

    Registering the script enables TAF Pro to pick up actions and dpStrings as defined in IBM Rational Functional Tester script

    a. Sign in to TAF Pro (3.1)
    b. Register Script (6.4.2)
3. Create Action Data Template in TAF Pro:
    a. Sign In to TAF Pro (3.1)
    b. Create Action Data Template (7.3.1)
4. Import Data to TAF Pro:
    a. Open the Action Data Template in Excel
    b. Add data and Save as a CSV file (7.3.2)
    c. Import the Action Data (7.4)
5. Create Scenarios
    a. Create (6.3.1) or Import (6.3.3) a Planned Scenario
    b. Create (6.4.1) or Import (6.4.4) Test Case(s) for the scenario.
    c. Add Action(s) to Test Cases (6.5.1)
6. Create Execution Suite

An Execution Suite is comprised of 1 or more scenarios executed in order to provide a testing outcome.

   a. Create Execution Suite(8.2)
   b. Add existing Scenarios in desired order (8.3.1)
   c. Attach Data Groups to Actions (8.2.3)

7. Regenerate Scenario if required (8.5.6)
8. Execute Scenario (10.2)
9. Verify Results
   a. View the Test Results (11)
   b. Identify issues highlighted in RED (11.2)
   c. Export results if required (11.8)
10. Filter Execution

    Create filters to refine execution and only run records matching specific criteria

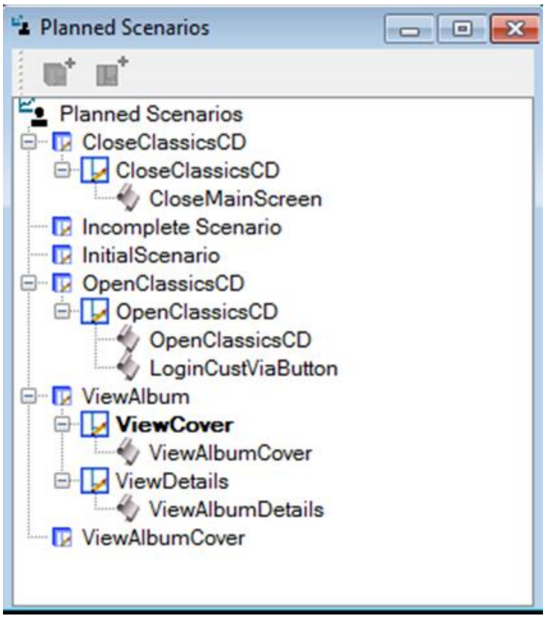
    a. Create Filter (9.2)
    b. Run Execution Suite and ensure the desired filter is selected (10.2.1)
11. If you change a script that is already in TAF Pro you need to Restart TAF Pro
    a. To pick up script changes (not including new dpStrings as that requires re-registration of the script).

For a detailed explanation on the above steps, refer to the User Guide sections listed against the items above.

6 Planned Scenarios

6.1 Understanding Scenarios

Effective use of an automated testing tool requires good planning and the Planned Scenarios window in TAF Pro is used for this purpose.



In TAF Pro Scenarios are grouped together in an Execution Suite to create an executable test. As part of the planning process we are hoping to identify components that are reusable and thus assist in creating a complete Test Program as quickly and effectively as possible.

In the diagram on the left we can see the **Planned Scenarios** window displayed with a few scenarios fully expanded.

The top level object **Planned Scenarios** is the root node that contains a list of all the scenarios in this user's environment.

There are three required levels within a scenario:

Scenario: The scenario identifier object that consists of a Name and a Description. The name must be less than 45 characters but should give a clear indication of the purpose of the Scenario. It may also include some coding to indicate its position in the test plan. The description should provide additional information that makes the function of the scenario clear.

Scenarios must contain one or more Test Cases

Test Case: The test case identifier object also consists of a Name and a Description. The name must be less than 45 characters but should give a clear indication of the purpose of the Test Case. The Test Case contains a set of steps called **Actions** undertaken as part of a scenario. Again we are looking for reusability when creating Test Cases.

Test Cases must contain one or more Actions to be complete.

Actions: Actions are steps that have been coded using the Test Automation Tool. They are imported to TAF Pro using the script registration process. Only Actions that have been registered in this environment can be added to a Test Case in the environment.

The illustration above shows three scenarios that might be grouped together in an **Execution Suite**. These are *OpenClassicsCD*, *ViewAlbum* and *CloseClassicsCD*. The *ViewAlbum* scenario contains two test cases *ViewCover* and *View Details*. The scenario *OpenClassicsCD* contains a single **Test Case,** using the same name, which contains two **Actions** *OpenClassicsCD* and *LoginCustViaButton*

It is easy to imagine with this simple set how our test program can be expanded using these scenarios, test cases and actions in different configurations with other actions to build additional Execution Suites.

6.2 Managing Scenarios

In TAF Pro Scenarios are managed for the entire environment. All users registered for the Environment see the same set of scenarios. In larger teams appropriate management practices regarding changes and naming conventions will need to be in place.

Scenarios can be created in an environment in two ways. They are either built in the environment or imported into the environment as part of an Execution Suite using the Execution Suites window. The Execution Suite import process is described in the chapter on Setting up Tests.

If a user creates, renames, modifies or removes a scenario in the environment other users will be unable to see the changes until they **Refresh** the **Planned Scenarios** or **Execution Suites**.

6.3 Building Scenarios

Scenarios are sets of actions that are performed sequentially and are data independent. When thinking about how to construct a scenario it is important to think about reusability. This depends on the application and how testing is carried out. If every test requires the user to log in and logout then these actions can be contained in a Test Case, if however the user logs in once and then performs several small tests that occur in no particular order then it would be appropriate to have a separate Scenario for log in and logout.

TAF Pro actions are data driven so the impact of an action can be altered by changing the data used. This can be achieved by selecting a different data group or using a filter to restrict the data used in the data group. An action can only be used once per Test Case, however it can be used in different Test Cases in a single scenario. Every time it is used a different data group or filtering selection can be made.

It is also important to consider how the scenario finishes. When the Execution Suite is constructed the user has the option to skip to the next scenario if a scenario fails or to just terminate that test line. The structure of the scenario should maximise the opportunity to continue the test if a single scenario fails. An action that takes the user to a common starting point is a useful one to have whenever possible.

The process for building a scenario is as follows:

12. Create or Add the scenario to the Planned scenarios window;
13. Create or add Test Cases to the scenario;
14. Add Actions to the Test Cases.

6.3.1 Create New Scenario

A new scenario is one that does not currently appear in the Planned Scenarios window or the list of available scenarios.

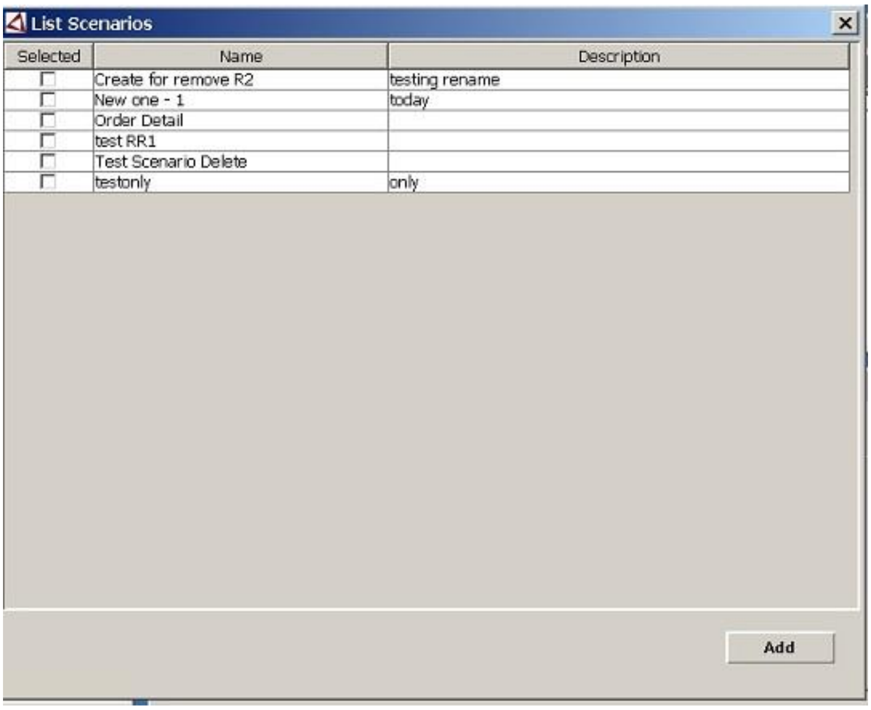
To create a new scenario:

1. In the Planned Scenarios window right click on Planned Scenarios and select New Scenario;
2. Enter the Scenario name and a short description of its purpose in the dialog box;
3. Click OK to create a scenario;
4. The new scenario is automatically added to the planned scenario tree;
5. Click OK on the Create Successful message.

6.3.2 Add Existing Scenario

Any scenarios that have previously been created but are not currently listed in the Planned Scenarios window can be selected using the Add Option;

1. Within the Planned Scenario window, click at the root level;
2. Right click on the root node > select Add Scenario
   OR click on the "Add Scenario" button in the tool bar;
3. If unused scenarios are available the List Scenarios window will be displayed.



Selected	Name	Description
□	Create for remove R2	testing rename
□	New one - 1	today
□	Order Detail	
□	test RR1	
□	Test Scenario Delete	
□	testonly	only

4. Select the checkboxes for the scenario(s) to be added then click on the ADD button
   OR click on a single scenario to be added > Right Click and select Add.
   The scenario(s) is/are now visible on the planned scenario tree.

6.3.3 Import Scenarios

Scenarios can also be imported using the File → Import → Scenarios option described in Section 15.2.1.

6.3.4 Refresh Scenario

The refresh scenario option updates the scenario to reflect changes made by other users. This is performed automatically when the user logs in.

To refresh the Scenarios manually:

1. In the Planned Scenarios window, right click on Planned Scenarios and select Refresh;
2. Any changes made to the stored version of the scenario will be returned from the database and the Scenario will be expanded to the Test Case level.

6.3.5 Remove Scenario

TAF Pro provides a facility to **Remove** a scenario. The Removed scenario name and description are retained in the list of scenarios for this environment. The structure of the scenario is NOT retained. The change affects all users of the environment and this option should therefore be used with caution.

Scenarios cannot be Removed if they are used in an Execution Suite by any user of the environment.

6.4 Managing Test Cases

All existing Test Cases are included in the list of existing Test Cases. Existing Test Cases can be added to the scenario as described in the Add Test Case section. A new Test Case is one that does not appear in that list.

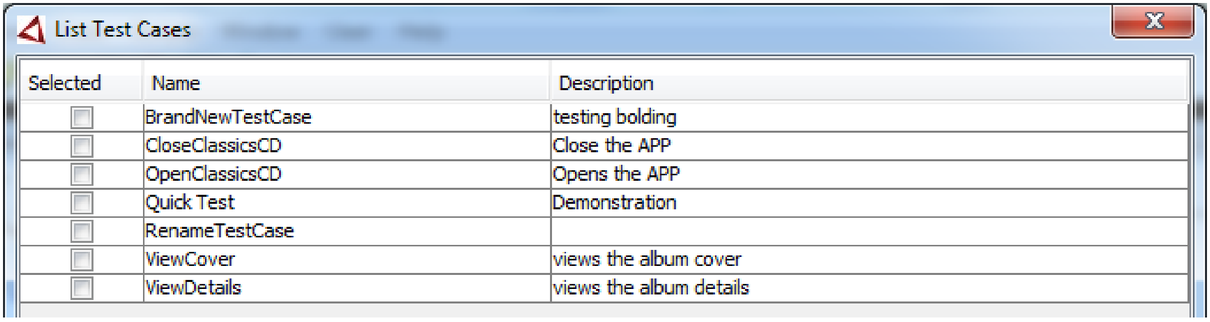
6.4.1 Adding a New Test Case

1. Click on the Scenario name and then right click and select New Test Case;
2. Enter the Test Case name and a short description of its purpose in the dialog box;
3. Click OK;
4. The Test Case is automatically added to end of the scenario selected.

The first time a test case is added to any Scenario it is identified in bold typeface. This is to indicate that it is the primary or first time the test case is used. If a secondary version of the Test Case needs to be modified the user should consider renaming it before altering it.

6.4.2 Add an Existing Test Case

1. In the Planned Scenarios window,
   - Click on the Scenario name and then right click and select Add, or
   - Use the Add Test Case button in the tool bar.
2. The available Test Cases are displayed showing the Test Case name and description.



List Test Cases

Selected	Name	Description
□	BrandNewTestCase	testing bolding
□	CloseClassicsCD	Close the APP
□	OpenClassicsCD	Opens the APP
□	Quick Test	Demonstration
□	RenameTestCase	
□	ViewCover	views the album cover
□	ViewDetails	views the album details

3. There are two ways to add the Test Cases –
   - Select all Test Cases to be added and click on the Add button, or,
   - Click on a single Test Case, Right Click and select Add.
4. Selected Test Cases will be visible under the selected scenario. Test Cases are displayed in the order added and will disappear from the List Test Cases panel.
5. Repeat to select other Test Cases if required.
6. When finished close the window.

6.4.3 Insert Test Case

Test cases can be inserted at a specific position within a scenario. The process is similar to that of adding a Test Case however the Test Case will be added at a selected point in the scenario:

1. Click on the Test Case that will be below the one to be inserted
2. Right click and select Insert Test Case
3. Add the Test Case(s) as described in the previous section.
4. The new Test Case is inserted above the test case selected.

6.4.4 Import Test Case

Test Cases can also be imported using the File → Import →Test Cases option described in Section 15.2.1 of this User Guide.

6.4.5 Remove Test Case

To remove a Test Case from a Scenario:

1. Select the Test Case, right click and select Remove Test Case.
2. The Test Case is removed from the Scenario but remains in the List of Test Cases.

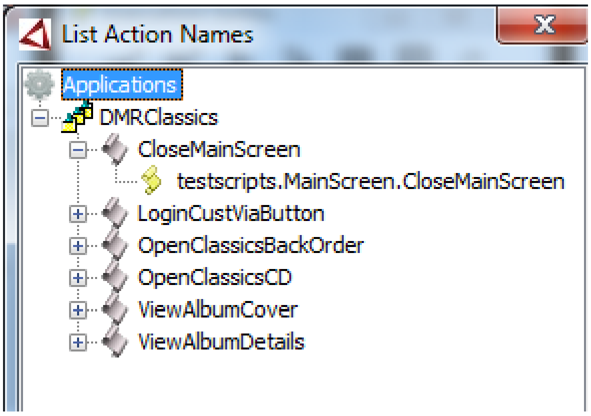
6.5 Managing Actions

6.5.1 Add an Action to a Test Case

The available **Actions** are steps in the test process that have been developed by the scripting team and registered in this environment.

Any registered Action can be added to a Test Case however the Test Case will only function if the set of actions follows a logical flow in the application.

1. Click on the Test Case, right click and select **Add Action;**
2. Expand the Applications tree to locate the required action,



3. Right click on the selected action and select Add;
4. The Action is added to the Test Case;
5. Add more actions or close the window.

6.5.2 Insert an Action

Insert Action allows the user to insert an Action at the required position in a set of Actions:

1. Click on the Test Case , right click and select **Insert Action**;
2. Complete the steps for Add action.

6.5.3 Remove Action

Remove Action allows the user to remove the selected Action from the Test Case. The Action is retained in the List of Actions:

1. Click on the Test Case, right click and select Remove Action;
2. The Action is removed from the Test Case.

6.6 Renaming Scenarios

Scenarios can be renamed using the Rename option. The rename applies to all users once they login or refresh their scenarios. The rename will update any Execution Suites and Filters that use the scenario. The rename will not alter the scenario name in existing Execution Results.

Renaming a scenario will not change any Build Files used by the Command Line Interface. Any existing builds using the scenario will fail as the old scenario name no longer matches the name of the scenario. This cannot be pre-determined as the CLI executes at run time and can only check the execution suitability at that time.

To rename a Scenario:

1. In the Planned Scenarios window, click on the Scenario to be renamed;

2. Right Click and select Rename Scenario;
3. Enter the New Name and the reason for the Rename in the dialog box;
4. Click OK to rename a scenario;
5. Close the rename Window.

6.7 Renaming Test Cases

Test Cases can be renamed using the Rename option. The new name must not exist in the list of available test cases. The rename only applies to the specific instance of the Test Case. The rename will update any Execution Suites and Filters that use the Test Case but will not update any naming in the execution results.

1. In the Planned Scenarios window, click on the Test case to be renamed;
2. Right Click on the Test Case and select Rename Test Case;
3. Rename the test case and add the Reason for the change (mandatory field);
4. Click the OK button;
5. A message will appear stating “Planned Test Case is successfully renamed”;
6. Click OK to close the message window;
7. Click OK to close the Rename window.

6.8 Using Drag and Drop to Re-Sequence Test Cases or Actions

TAF Pro provides a drag and drop facility to allow re sequencing of Test Cases or Actions. Test Cases and Actions can be moved within their existing containers or moved to other containers using this option. Only movements to a suitable location are allowed, some locations are not permitted because the appropriate container is considered uncertain. It is not possible to move an item to the end of a list via Drag and Drop. If this is required the item should be moved to second-last position and then the item in last position moved up.

7 Managing Data

7.1 Data group Use in TAF Pro

TAF Pro testing is data driven using data stored in **Action** Data Groups and **Execution** Data Groups. The process of creating templates for Action Data and Execution Data as well as importing and exporting the data is described in Section 15.23 of this User Guide.

7.1.1 Data Templates

Data Groups are constructed around a template that defines the fields that are required to exist in the Data Group. These are created from the known information relating to registered scripts. The template only needs to be changed when a new data pool variable is created in the Test Tool script. Other modification of the scripts simply require the script to be reregistered either using the **Scripts→ Register** function or the automatic registration as part of Execution Suite Execution.

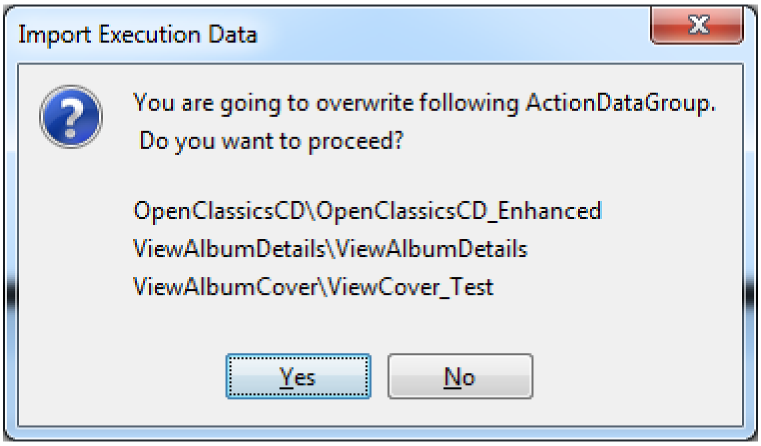
7.1.2 Action Data Groups

Action Data Groups relate to a single Action and can be used in any Execution Suite. It is feasible to have a single data group that covers every possible scenario relating to that action but it is easier to manage multiple data groups to cover different testing scenarios. Filters can be used to determine the data lines that are included in the run when an Execution Suite is executed.

Action Data groups can be displayed from various locations in TAF Pro in the Action Data window.

7.1.3 Execution Data Groups

Execution Data Groups are used by a specific Execution Suite and contain data relating to all the actions in that Execution Suite. When the Execution data is imported it over-writes the data in the existing data groups associated with the Execution Suite.



Import Execution Data

You are going to overwrite following ActionDataGroup.
Do you want to proceed?

OpenClassicsCD\OpenClassicsCD_Enhanced
ViewAlbumDetails\ViewAlbumDetails
ViewAlbumCover\ViewCover_Test

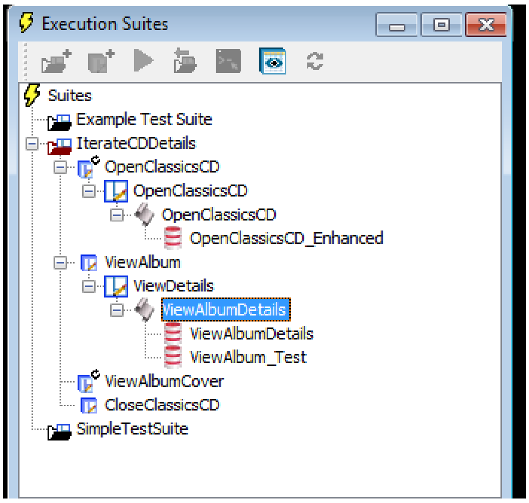
The **Import Execution Data** popup will warn you when existing data is to be over written.

7.1.4 Understanding how data groups are used in a Test.

When a test is executed TAF Pro builds rows by taking one line from each data group based on any filters selected. Rows are added by taking the lines sequentially. When all the lines in a Data Group have been completed, the selection restarts at the beginning of that Data Group. This continues until all lines in every Data Group have been used at least once. The testing results for each row is listed in the Results and the test can be rerun using the row number.

7.2 Associating Data Groups with Execution Suites

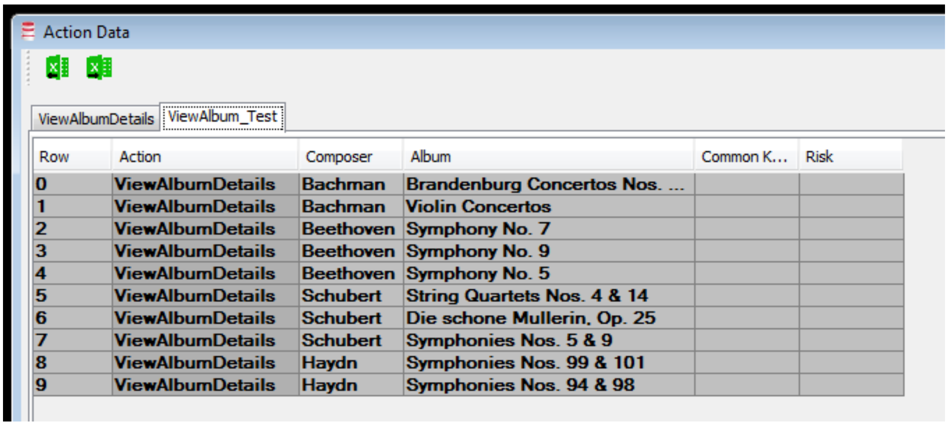
An Execution Suite is built by combining Scenarios with Data Groups to create a sequence of actions that test the application. Only completed Scenarios can be included in the Execution Suite and the Execution Suite is not complete until every Action in the Execution Suite is linked to at least one Data Group.



```
Suites
  Example Test Suite
  IterateCDDetails
    OpenClassicsCD
      OpenClassicsCD
        OpenClassicsCD
          OpenClassicsCD_Enhanced
    ViewAlbum
      ViewDetails
        ViewAlbumDetails
          ViewAlbumDetails
          ViewAlbum_Test
    ViewAlbumCover
    CloseClassicsCD
  SimpleTestSuite
```

In the example shown the ViewAlbumDetails Action is associated with two separate data groups. When the Execution Suite is run the two data groups will be used as if they were one with the first data group processed in its entirety before the second one is started.

Right Clicking on the Action name in the Execution Suites window provides the option show Action Data. When this is selected the Action Data is displayed in the Action Data window. If the Action is associated with more than one Data Groups then each Data group is shown under a separate tab as shown.



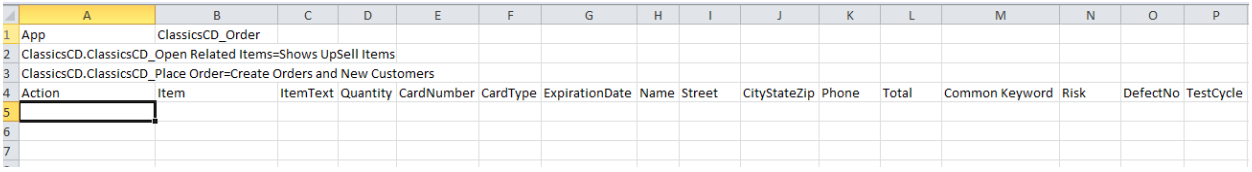
Row	Action	Composer	Album	Common K...	Risk
0	ViewAlbumDetails	Bachman	Brandenburg Concertos Nos. ...		
1	ViewAlbumDetails	Bachman	Violin Concertos		
2	ViewAlbumDetails	Beethoven	Symphony No. 7		
3	ViewAlbumDetails	Beethoven	Symphony No. 9		
4	ViewAlbumDetails	Beethoven	Symphony No. 5		
5	ViewAlbumDetails	Schubert	String Quartets Nos. 4 & 14		
6	ViewAlbumDetails	Schubert	Die schone Mullerin, Op. 25		
7	ViewAlbumDetails	Schubert	Symphonies Nos. 5 & 9		
8	ViewAlbumDetails	Haydn	Symphonies Nos. 99 & 101		
9	ViewAlbumDetails	Haydn	Symphonies Nos. 94 & 98		

7.3 Data Templates

7.3.1 Understanding the Action Data Template

TAF Pro provides capability to generate templates that users can populate with test data and import into the framework for use with their test cases. The Create Data Template function will generate a CSV file that can be used with a spreadsheet to create rows of test data that can be imported back into TAF Pro.

On creation the Action Data Template contains default rows containing data that should not be altered. These are determined from the registered script. The last row contains the headers for the data columns.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	App	ClassicsCD_Order														
2	ClassicsCD.ClassicsCD_Open Related Items=Shows UpSell Items															
3	ClassicsCD.ClassicsCD_Place Order=Create Orders and New Customers															
4	Action	Item	ItemText	Quantity	CardNumber	CardType	ExpirationDate	Name	Street	CityStateZip	Phone	Total	Common Keyword	Risk	DefectNo	TestCycle
5																
6																
7																

The template will have **n + 2** rows where **n = number of 'Actions'** that a script contains. The rows of information created in the template must not be removed. They contain important information that associates with the action when imported into TAF Pro. The Header Rows can be explained as follows:

- Row 1 => Project Folder (*App*) and Script Name (*ClassicsCD_Order*)
- Row 2 => Application (*ClassicsCD*) .Action Identifier 1(*ClassicsCD_Open Related Items*) and Description (*Shows Upsell Items*)
- Row i => Action Identifier n (i = n + 1)
- Row i+1 =>Script fieldnames or dpString statements from the script. (Column headers)

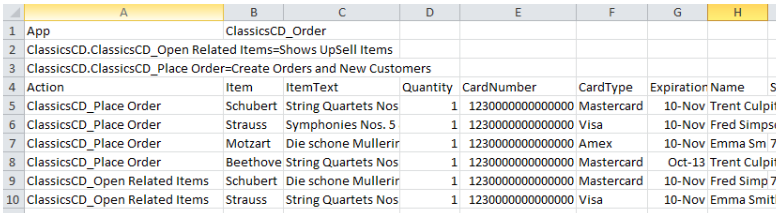
Note: if the script does not require any data input, the row with the Script fieldnames will only contain 3 data fieldnames **Action, Common Keyword, Risk**. These are defaults added by TAF Pro to allow additional filtering capability within the test execution. The spreadsheet above shows these columns as A, M, and N respectively.

The user can add as many additional columns as required. This feature is enables the user to select data lines to execute using Filters. The spreadsheet above shows these columns at the end of the spreadsheet in columns O and P, for example.

7.3.2 Adding Data to the Action Data Template

The rows generated as part of the Data Templates and described above must be retained.

The user can then add data according to the available actions to fill the Spreadsheet. Column A (Action) must be set to one of the actions within the script and copied from row 2+n for each different action contained in the script as shown below.



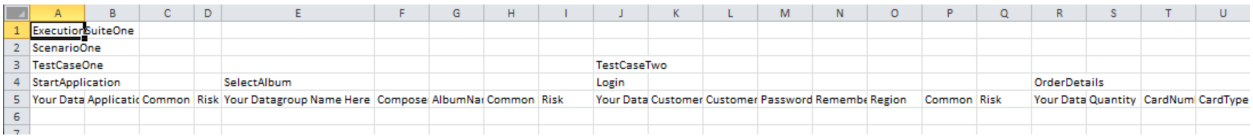
	A	B	C	D	E	F	G	H	
1	App	ClassicsCD_Order							
2	ClassicsCD.ClassicsCD_Open Related Items=Shows UpSell Items								
3	ClassicsCD.ClassicsCD_Place Order=Create Orders and New Customers								
4	Action	Item	ItemText	Quantity	CardNumber	CardType	Expiration	Name	S
5	ClassicsCD_Place Order	Schubert	String Quartets Nos	1	1230000000000000	Mastercard	10-Nov	Trent Culpit	
6	ClassicsCD_Place Order	Strauss	Symphonies Nos. 5	1	1230000000000000	Visa	10-Nov	Fred Simps	
7	ClassicsCD_Place Order	Motzart	Die schone Mullerir	1	1230000000000000	Amex	10-Nov	Emma Sm	7
8	ClassicsCD_Place Order	Beethove	String Quartets Nos	1	1230000000000000	Mastercard	Oct-13	Trent Culpit	
9	ClassicsCD_Open Related Items	Schubert	Die schone Mullerir	1	1230000000000000	Mastercard	10-Nov	Fred Simp	7
10	ClassicsCD_Open Related Items	Strauss	String Quartets Nos	1	1230000000000000	Visa	10-Nov	Emma Smit	

When the required data is added, save the Spreadsheet in the .csv format.

Where no data is required, the Data Group Template must be updated with the action and DEFAULT added as the first data row as shown below.

7.3.3 Understanding the Execution Suite Data Template

The function to create an Execution Data Template results in a CSV file that contains certain default values, as shown in the example below. The rows of information created in the template must not be removed. They contain important information that associates with the action when imported into TAF Pro. The Execution Data template is similar to the Action Data template but there are significant differences noted below.



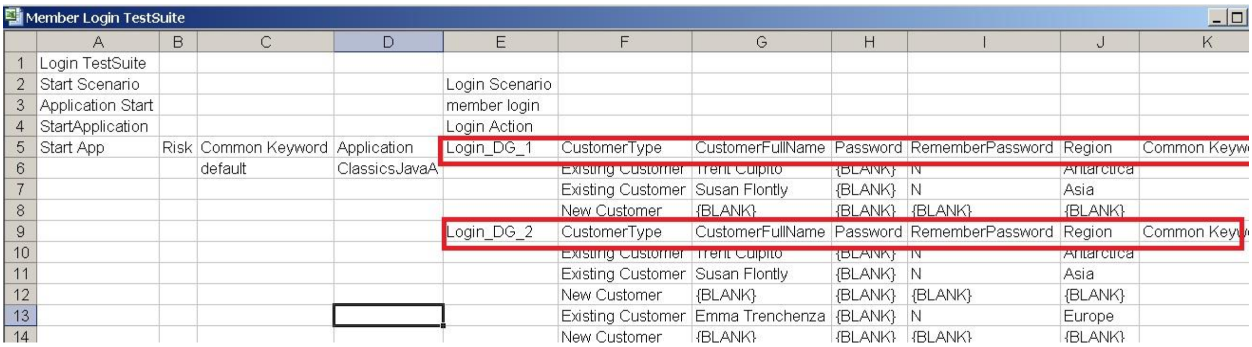
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	ExecutionSuiteOne																				
2	ScenarioOne																				
3	TestCaseOne									TestCaseTwo											
4	StartApplication				SelectAlbum					Login								OrderDetails			
5	Your Data	Applicatio	Common	Risk	Your Datagroup Name Here	Compose	AlbumNa	Common	Risk	Your Data	Customer	Customer	Password	Remembe	Region	Common	Risk	Your Data	Quantity	CardNum	CardType
6																					

The Header Rows can be explained as follows:

- Row 1 => Execution Suite Name (*ExecutionSuiteOne*)
- Row 2 => Scenario Names within the execution Suite (*ScenarioOne, ScenarioTwo*)
- Row 3 => Test Case Names within the Scenario. (*Test CaseOne, Test CaseTwo, etc.)*
- Row 4 => Action Name within each Test Case. *(StartApplication, SelectAlbum,etc*)
- Row 5 => For each action, columns for the Data Group Name, followed by the Data Group fields. The Data Group Name column is aligned with the action name.

7.3.4 Adding Data to the Execution Suite Data Template

The 5 rows generated as part of the Data Template and outlined above must be retained. In row five the “Your Data Group Name Here” column header(s) should be changed to show the Data Group used. If more than one Data Group is used for an action an additional set of columns should be appended to the end of the previous Data Group for that action, as shown below. These columns should have the same contents in row five as the initial group with only the Data Group name changed.

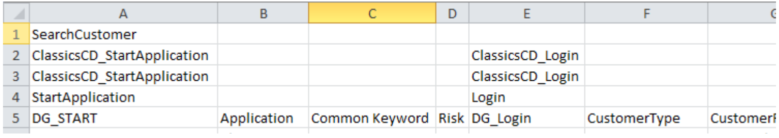


Member Login TestSuite

	A	B	C	D	E	F	G	H	I	J	K
1	Login TestSuite										
2	Start Scenario				Login Scenario						
3	Application Start				member login						
4	StartApplication				Login Action						
5	Start App	Risk	Common Keyword	Application	Login_DG_1	CustomerType	CustomerFullName	Password	RememberPassword	Region	Common Keyw
6			default	ClassicsJavaA		Existing Customer	Trent Culpito	{BLANK}	N	Antarctica	
7						Existing Customer	Susan Flontly	{BLANK}	N	Asia	
8						New Customer	{BLANK}	{BLANK}	{BLANK}	{BLANK}	
9					Login_DG_2	CustomerType	CustomerFullName	Password	RememberPassword	Region	Common Keyw
10						Existing Customer	Trent Culpito	{BLANK}	N	Antarctica	
11						Existing Customer	Susan Flontly	{BLANK}	N	Asia	
12						New Customer	{BLANK}	{BLANK}	{BLANK}	{BLANK}	
13						Existing Customer	Emma Trenchenza	{BLANK}	N	Europe	
14						New Customer	{BLANK}	{BLANK}	{BLANK}	{BLANK}	

Figure 7-II

,



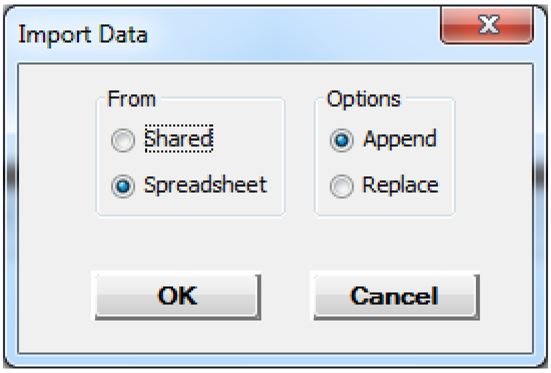
	A	B	C	D	E	F	G
1	SearchCustomer						
2	ClassicsCD_StartApplication				ClassicsCD_Login		
3	ClassicsCD_StartApplication				ClassicsCD_Login		
4	StartApplication				Login		
5	DG_START	Application	Common Keyword	Risk	DG_Login	CustomerType	CustomerF

Once the execution data template is set up, save the template in .CSV format. It is recommended that execution data templates are stored in a sub folder such as “OutputFiles\DataTemplates” using a prefix such as ‘testSuite.” followed by the Execution Suite name to identify that they are different from Action Data Templates, eg. “OutputFiles\DataTemplates\testSuite.SearchCustomer.csv”.

To create a new Execution Data Group:

1. Open the Execution Template using Excel;
2. Save the Template to your Data Groups folder providing a defining suffix. It is recommended that all Execution Data Group csv files are stored in a sub folder such as “OutputFiles\Data Groups” using just the action name and a descriptive tag to identify the Data Group eg. “OutputFiles\Data Groups\testSuiteSearchCustomer_NSW.csv”;
3. Add Data to the various fields in the new file;
4. Save the Execution Data Group File in .CSV format.

7.4 Importing Action Data from Spreadsheet



Importing data from Spreadsheet refers to importing data into TAF Pro from a local data source. Once the data sheet has been built (see above) the user can import the contents of it to TAF Pro to provide the Action with data as follows:

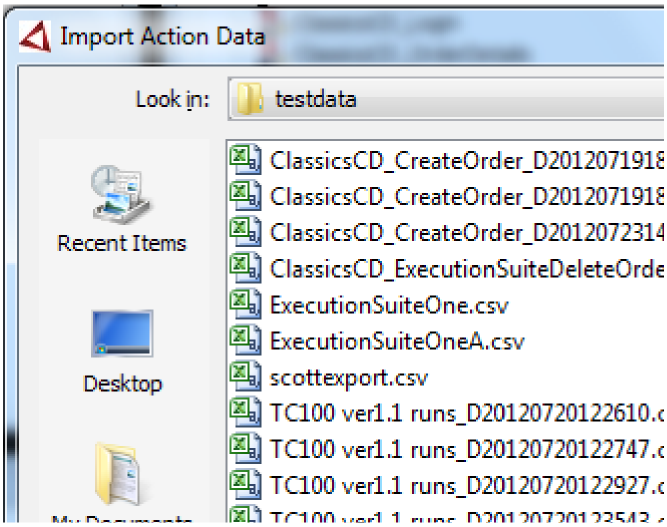
1. From the Menu Bar select Data → Import → Import Action Data;
2. The **Import Data** window is displayed;
3. Click on From Spreadsheet;

4. Select either of the following options -

    a. ***Append*** – to create the data group first time or add more rows of data to an existing data group, or;

    b. ***Replace*** – to create the data group first time or replace any previous data that may exist in the selected data group.

5. Click ***OK;***
6. The **Import Action** window is displayed (see below);
7. ***Select*** the data file to be imported;

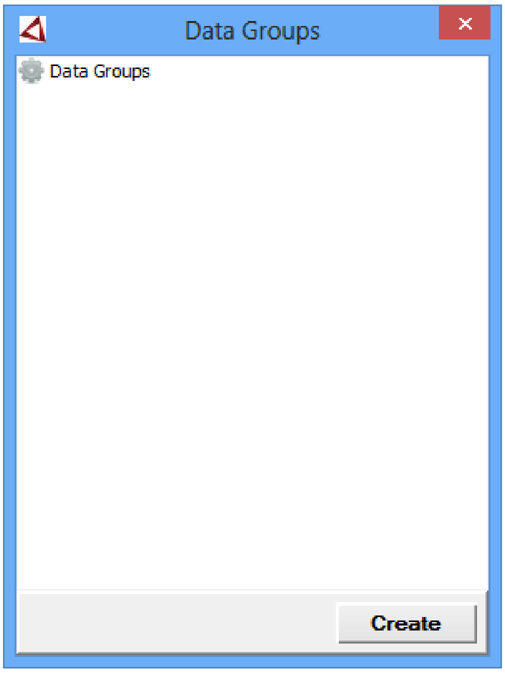


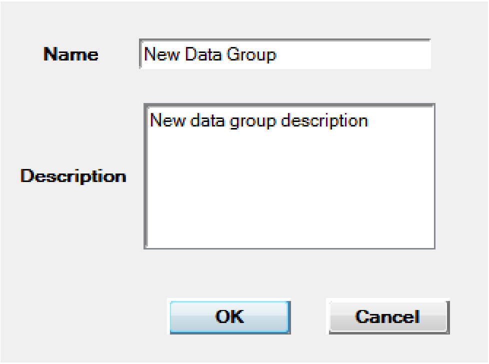
Import Action Data

Look in: testdata

ClassicsCD_CreateOrder_D2012071918
ClassicsCD_CreateOrder_D2012071918
ClassicsCD_CreateOrder_D2012072314
ClassicsCD_ExecutionSuiteDeleteOrde
ExecutionSuiteOne.csv
ExecutionSuiteOneA.csv
scottexport.csv
TC100 ver1.1 runs_D20120720122610.c
TC100 ver1.1 runs_D20120720122747.c
TC100 ver1.1 runs_D20120720122927.c
TC100 ver1.1 runs_D20120720123543.c

8. Click **Open;**
9. The **Data Groups** window is displayed (see below) Click on Data Groups root to display the data groups;
10. To choose an existing data group -
    - ***Click*** the data group from those listed;
    - ***Right click*** and select ***Select;***
11. To create a new Data Group click **Create** or right click on an existing Data Group and select **New**;
    - The **New Data Group** window is displayed (see below);
    - Enter ***Data Group name*** and ***Description;***
    - Click ***OK.***





Name: New Data Group

Description: New data group description

12. A message for successful completion of importing data should be displayed in the status bar.

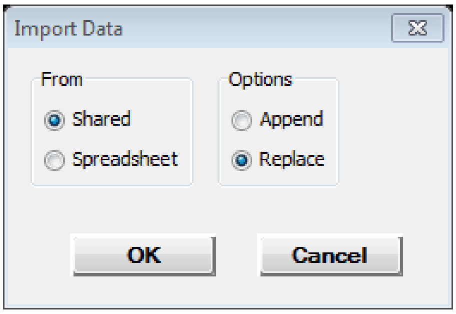
Alternatively, importing action data can also be done from the **Action Data** window as follows:

1. Select the Action/data group under the relevant test suite from the **Execution Suites** window;
2. The **Import** and **Export** buttons on the Action Data console will be enabled, and the selected data group will be highlighted;
3. Click on the ***Import Action Data*** button & follow above steps 2 to 10 to import the Action Data.

7.5 Importing Action Data from Share

Once data is imported to TAF Pro, it can be shared with all users of the environment. Refer to Exporting data to the Share below. When data has been created and added to the Share for an environment all users are able to then import that data to TAF Pro via the following steps:

1. From the Menu Bar select ***Data → Import;***
2. The Import Data Window is displayed;
3. Click on ***From Share***;
4. Select either -
   - ***Append*** – to create the data group first time or add more rows of data to an existing data group, or ;
   - ***Replace*** – to create the data group first time or replace any previous data that may exist in the selected data group.
5. Click ***OK;***
6. The Data Group window is displayed;
7. Click on the Data Groups icon to display the data groups;
8. Click the data group from those listed;
9. Right click and then click ***Select;***
10. Message for successful completion of importing data should be displayed in the status bar.



Import Data

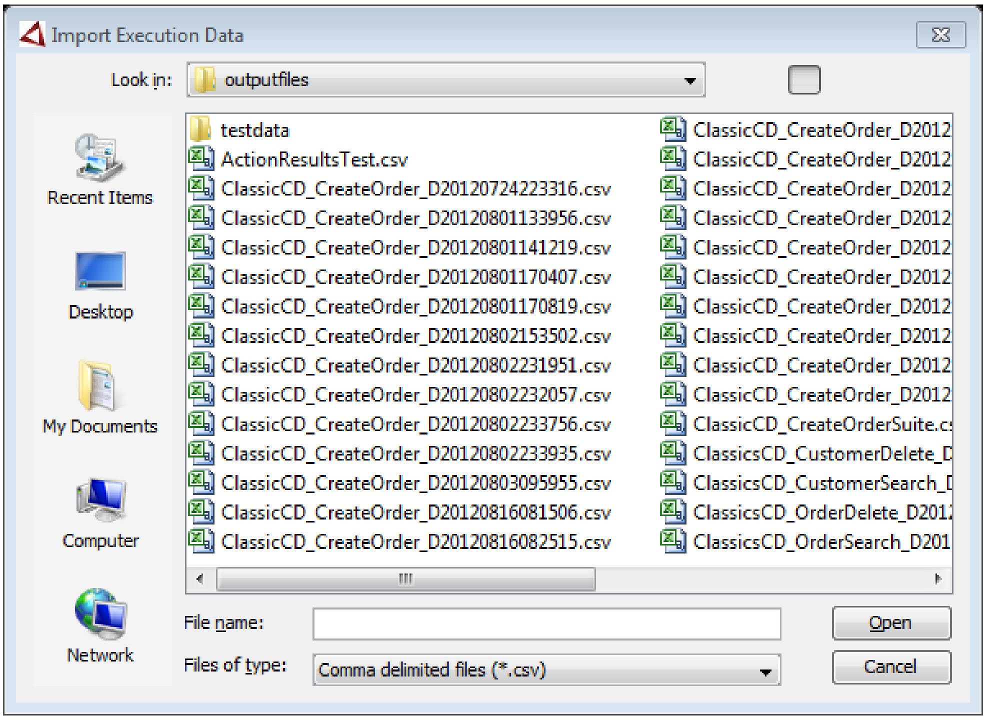
From: ● Shared ○ Spreadsheet

Options: ○ Append ● Replace

7.6 Importing Execution Data

The Import function allows the user to populate TAF Pro data groups from spreadsheets populated outside of the tool. Once the data sheet has been built (see **Adding Data to the Execution Suite Data Template,** above) the user can import the contents of the spreadsheet into TAF Pro:

1. From the Menu Bar select Data Import Import Execution Data;
2. The Import Execution Data window is displayed see below;



Import Execution Data

Look in: outputfiles

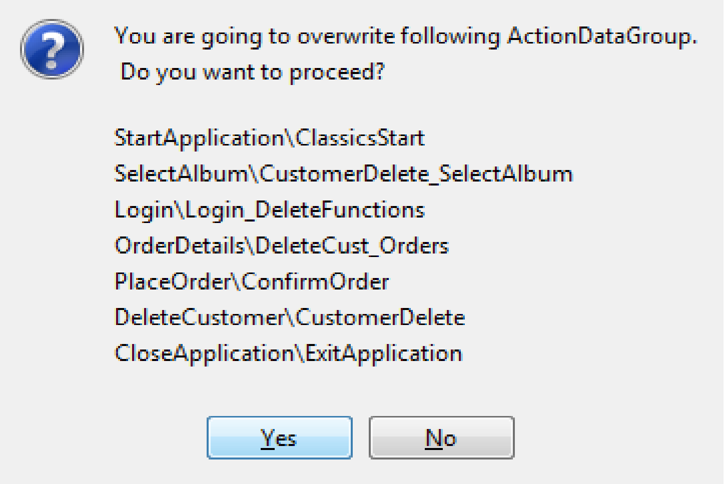
Files of type: Comma delimited files (*.csv)

Recent Items
Desktop
My Documents
Computer
Network
testdata
ActionResultsTest.csv
ClassicCD_CreateOrder_D20120724223316.csv
ClassicCD_CreateOrder_D20120801133956.csv
ClassicCD_CreateOrder_D20120801141219.csv
ClassicCD_CreateOrder_D20120801170407.csv
ClassicCD_CreateOrder_D20120801170819.csv
ClassicCD_CreateOrder_D20120802153502.csv
ClassicCD_CreateOrder_D20120802231951.csv
ClassicCD_CreateOrder_D20120802232057.csv
ClassicCD_CreateOrder_D20120802233756.csv
ClassicCD_CreateOrder_D20120802233935.csv
ClassicCD_CreateOrder_D20120803095955.csv
ClassicCD_CreateOrder_D20120816081506.csv
ClassicCD_CreateOrder_D20120816082515.csv
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrderSuite.c
ClassicsCD_CustomerDelete_D
ClassicsCD_CustomerSearch_
ClassicsCD_OrderDelete_D201
ClassicsCD_OrderSearch_D201
File name:
Open
Cancel

3. Select the data file to be imported and click Open;

If the data already exists for the execution suite the following message is displayed to confirm the data in the execution suite will be replaced.

4. Click **Yes** to continue or **No** to cancel the import;
   On selecting **Yes** the existing data in the execution suite is automatically replaced.
5. A message for successful completion of the Import will be displayed in the status bar.

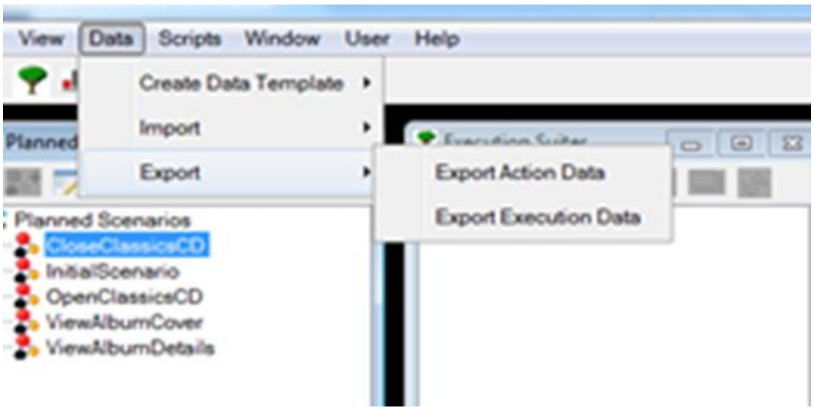


You are going to overwrite following ActionDataGroup.
Do you want to proceed?

StartApplication\ClassicsStart
SelectAlbum\CustomerDelete_SelectAlbum
Login\Login_DeleteFunctions
OrderDetails\DeleteCust_Orders
PlaceOrder\ConfirmOrder
DeleteCustomer\CustomerDelete
CloseApplication\ExitApplication

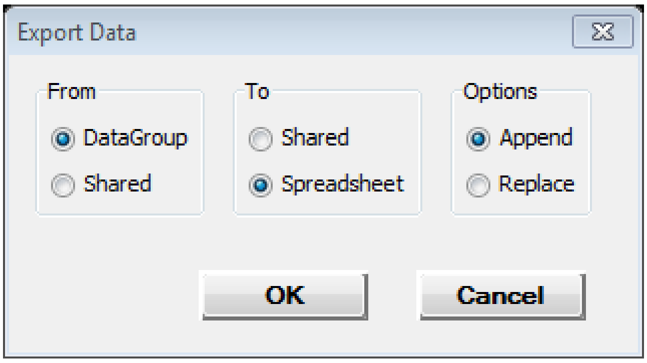
7.7 Exporting Data

The ***Data* → *Export*** function allows the user to export data from TAF in the form of a spreadsheet. Data export can be done at both Action Data and Execution Data levels. The user can export action data to the Shared environment to provide the ability to share action data among users within the same environment through ***Export from TAF Pro Action Data Group to Share.*** Export also provides the ability to send data to spreadsheet through ***Export from Share to Spreadsheet*** to enable mass updates to be done.



7.7.1 Exporting Action Data to Spreadsheet

1. From the Menu Bar select **Data →Export → Export Action Data;**
2. The **Export Data** window is displayed;
3. Choose either From DataGroup or From Share;
4. Choose To Spreadsheet;
5. Select either:
   - ***Append*** – to create the data group first time or add more rows of data to an existing data group, or;
   - ***Replace*** – to create the data group first time or replace any previous data that may exist in the selected data group.
6. Click ***OK;***
7. The **Data Groups** window is displayed;
8. Click on the Data Groups icon to show the data groups;
9. Click on the required data group;
10. Right Click and select ***Select;***
11. The **Export Action** window is displayed (see below);

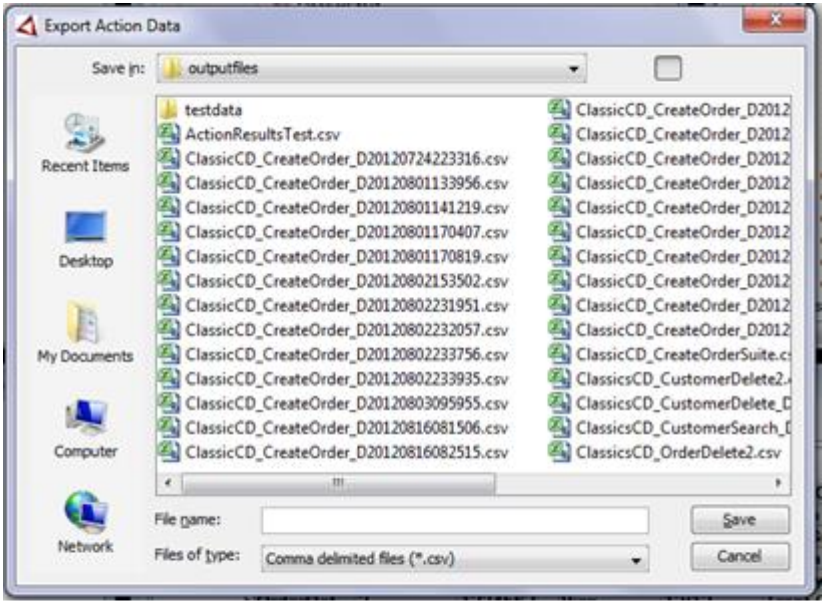


Export Data

From
● DataGroup
○ Shared

To
○ Shared
● Spreadsheet

Options
● Append
○ Replace


Export Action Data
Save in:
outputfiles
Recent Items
Desktop
My Documents
Computer
Network
testdata
ActionResultsTest.csv
ClassicCD_CreateOrder_D20120724223316.csv
ClassicCD_CreateOrder_D20120801133956.csv
ClassicCD_CreateOrder_D20120801141219.csv
ClassicCD_CreateOrder_D20120801170407.csv
ClassicCD_CreateOrder_D20120801170819.csv
ClassicCD_CreateOrder_D20120802153502.csv
ClassicCD_CreateOrder_D20120802231951.csv
ClassicCD_CreateOrder_D20120802232057.csv
ClassicCD_CreateOrder_D20120802233756.csv
ClassicCD_CreateOrder_D20120802233935.csv
ClassicCD_CreateOrder_D20120803095955.csv
ClassicCD_CreateOrder_D20120816081506.csv
ClassicCD_CreateOrder_D20120816082515.csv
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrderSuite.c
ClassicsCD_CustomerDelete2.
ClassicsCD_CustomerDelete_D
ClassicsCD_CustomerSearch_D
ClassicsCD_OrderDelete2.csv
File name:
Files of type:
Comma delimited files (*.csv)
Save
Cancel

12. Enter a new ***File name*** or ***Select*** a file to extend or overwrite;
13. Ensure the file type is Comma delimited files (*.csv);
14. Click ***Save;***
15. Message for successful Export of Action Data should be displayed on the status bar.

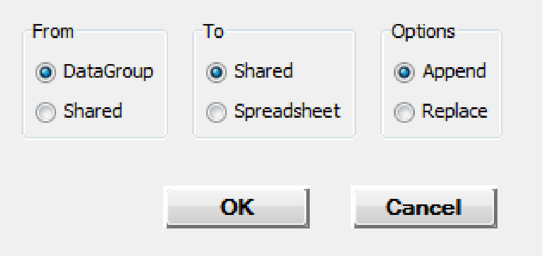
Alternatively, exporting action data can also be done from the Action Data window as follows:

1. User must select the Action/data group under the particular test suite from the **Execution Suites** window;
2. This enables the **Import** and **Export** buttons on the Action Data console, along with displaying the selected data group;
3. User can click on the **Export Action Data** button & follow above steps 2 to 15 for successful export of Action Data.

7.7.2 Exporting Action Data to the Share

As the Execution Suites are unique to each user's environment within TAF Pro, exporting data from Data Group to the Share provides the capability for users within the same inplementation to use the same data for test case actions. This enables a single data set to be shared across users.

1. From the Menu Bar select **Data → Export → Export Action Data;**
2. The **Export Data** window is displayed;
3. Choose **From DataGroup**,
4. Choose **To Shared**;
5. Select either:
   - ***Append*** – to create the data group first time or add more rows of data to an existing data group, or;


From: ● DataGroup ○ Shared
To: ● Shared ○ Spreadsheet
Options: ● Append ○ Replace

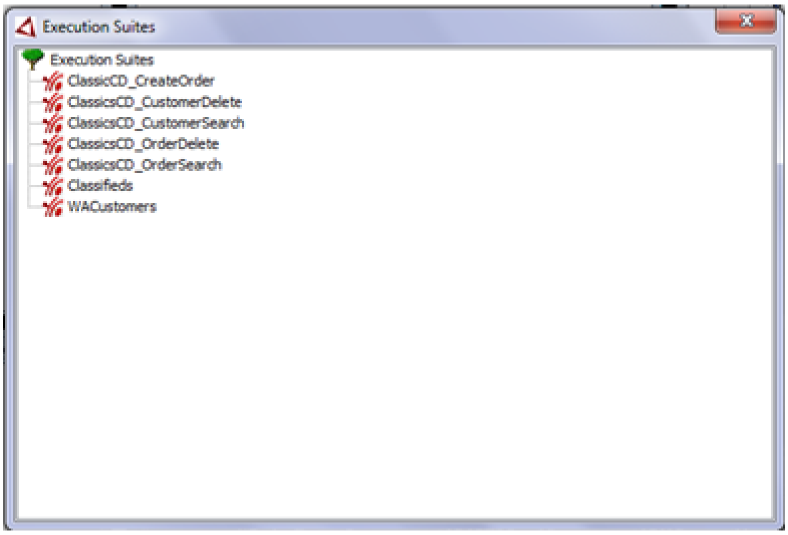
- ***Replace*** – to create the data group first time or replace any previous data that may exist in the selected data group. (Note Replace is only available if the user created the data in the share initially, otherwise append is defaulted).

6. Click **OK**;
7. The **Data Group** window is displayed;
8. Click the data group from those listed;
9. **Right Click** and then click **Select;**
10. The message **Local Data Exported to Server** is displayed.

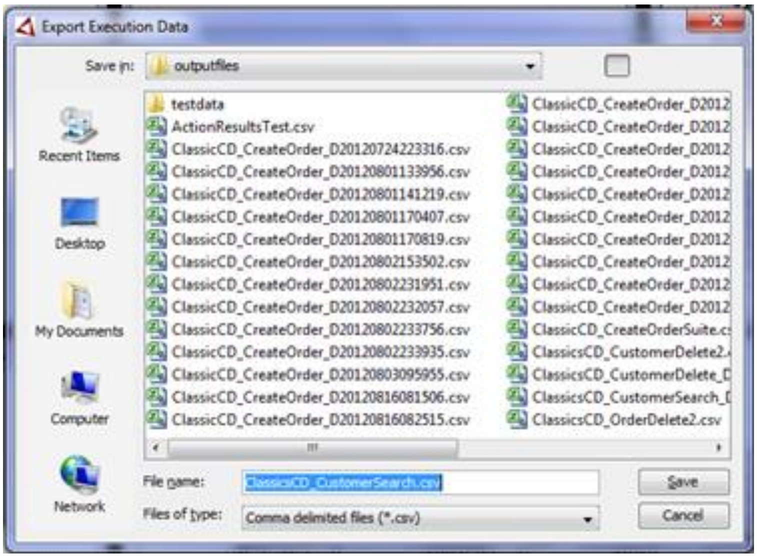
7.7.3 Exporting Execution Data

The data that is available on all actions in an Execution Suite can be exported in one operation. It provides the user with the capability to view all data attached to an execution suite in a single spreadsheet.

1. Select **Data → Export → Export Execution Data;**
2. The **Execution Suites** window is displayed;
3. Click to ***select*** the Execution Suite to export;
4. Right click and select ***Export Execution Data;***
5. The **Export Execution Data** window is displayed as shown below;



Execution Suites
- ClassicCD_CreateOrder
- ClassicsCD_CustomerDelete
- ClassicsCD_CustomerSearch
- ClassicsCD_OrderDelete
- ClassicsCD_OrderSearch
- Classifieds
- WACustomers



Save in: outputfiles

testdata
ActionResultsTest.csv
ClassicCD_CreateOrder_D20120724223316.csv
ClassicCD_CreateOrder_D20120801133956.csv
ClassicCD_CreateOrder_D20120801141219.csv
ClassicCD_CreateOrder_D20120801170407.csv
ClassicCD_CreateOrder_D20120801170819.csv
ClassicCD_CreateOrder_D20120802153502.csv
ClassicCD_CreateOrder_D20120802231951.csv
ClassicCD_CreateOrder_D20120802232057.csv
ClassicCD_CreateOrder_D20120802233756.csv
ClassicCD_CreateOrder_D20120802233935.csv
ClassicCD_CreateOrder_D20120803095955.csv
ClassicCD_CreateOrder_D20120816081506.csv
ClassicCD_CreateOrder_D20120816082515.csv
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrder_D2012
ClassicCD_CreateOrderSuite.c
ClassicsCD_CustomerDelete2.
ClassicsCD_CustomerDelete_D
ClassicsCD_CustomerSearch_I
ClassicsCD_OrderDelete2.csv

File name: ClassicsCD_CustomerSearch.csv

Files of type: Comma delimited files (*.csv)

6. Enter a new ***File Name*** or ***Select*** a file (overwrite) to which the Execution Data will be saved;
7. Click ***Save;***
8. The status bar will report the export information.

8 Setting up Tests

8.1 Execution Suites

TAF Pro uses Execution Suites to produce specific test exercises. This is where Planned Scenarios are linked together to produce a sequence of actions that, in their entirety, can constitute an end-to-end test of an application. Individual scenarios that have been created in the planning window are linked together and associated with specific Data Groups to create a unique test.

Once the Execution Suite is created it can be modified by setting run parameters, changing Data Groups or using filters to create different tests following the same sequence of test cases.

Execution suites can be run from within the TAF Pro application or using the Command Line facility. The steps for executing Execution Suites and generating the command line Build Files required to use the Command Line interface are described in Chapter 12

8.1.1 Understanding Data Group processing

To execute an Execution Suite every action in the suite must be associated with at least one Data Group. A TAF Pro row consists of single line taken from each Data Group. Each time a new row is commenced the next row in the Data Group is used. This continues until every line in the largest Data Group has been processed and the test then stops.

If an action has more than one Data Group associated with it the Data Groups are concatenated. Because Data Groups differ in the number of lines in them TAF Pro reuses the rows cyclically in the smaller Data Groups restarting from the beginning.

If the test uses iteration, TAF Pro looks for matching lines in the relevant Data Group at or after the line that was included in the TAF Pro row.

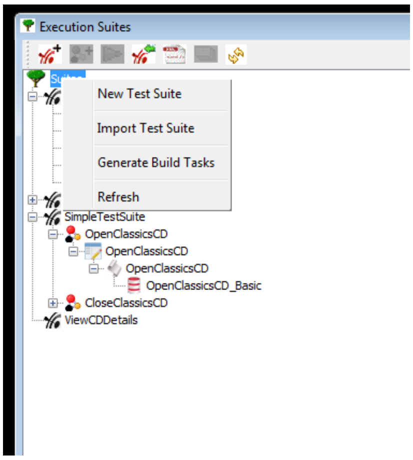
8.1.2 Managing Execution Suites

Execution Suites can be created and run within a single environment or transferred between environments. Execution Suites transferred between environments become independent copies that can be modified in the new environment without altering the original. When transferring an Execution Suite between environments the Planned Scenarios, Test Cases and Actions are always transferred to the new environment. The user has the option to transfer Data Groups and filters when transferring the Execution Suite.

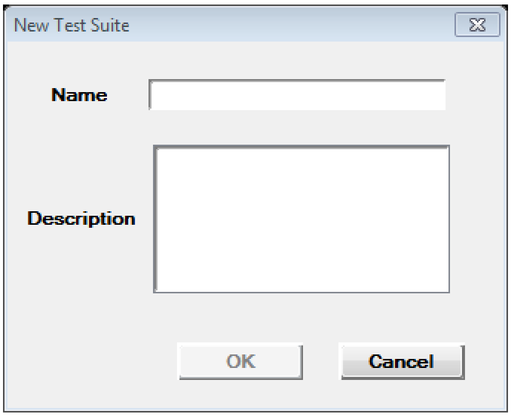
8.2 Creating Execution Suites

8.2.1 Name and Describe the Execution Suite

Execution Suites are created in the Execution Suites window. Select the Add a suite icon or right click on the root node and select **New Execution Suite** to create a new Execution Suite.



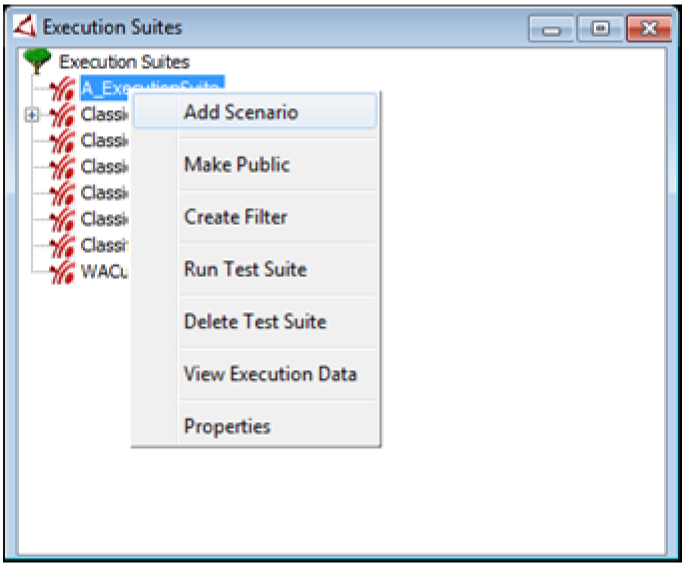
The new Execution Suite window will be displayed



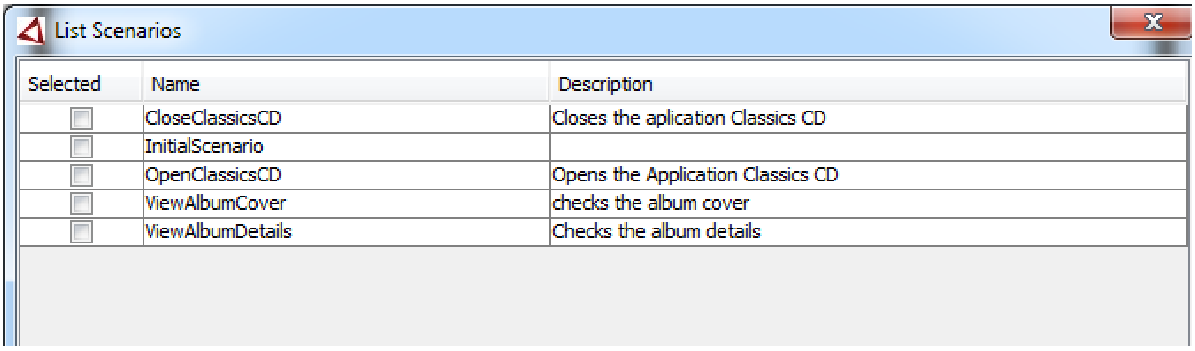
1. Enter Execution Suite Name. The Execution Suite name is restricted to 45 characters. It may be used later to identify Filters and Executions Runs. Consider adding an abbreviation to simplify later naming of filters and runs;
2. Enter Description. The description should provide more complete detail about what the Execution Suite does. There is no need at this point to add detail of filters and run parameters that may be created later as these have their own descriptions;
3. Click ***OK*** to create the new Execution Suite;
4. The New Execution Suite is added to the Tree in Alphabetical order;
5. To confirm successful creation or the Execution Suite an information popup, ***Create successful***, will be displayed and the main window status bar will display the message - ***Successfully created new Execution Suite {Name}***.

8.2.2 Add scenarios

Right click on the Execution Suite and select Add Scenario:



The List Scenarios window will be displayed. All the scenarios listed in the Planned Scenarios window that are not currently used in the Execution Suite are displayed:



Selected	Name	Description
□	CloseClassicsCD	Closes the aplication Classics CD
□	InitialScenario	
□	OpenClassicsCD	Opens the Application Classics CD
□	ViewAlbumCover	checks the album cover
□	ViewAlbumDetails	Checks the album details

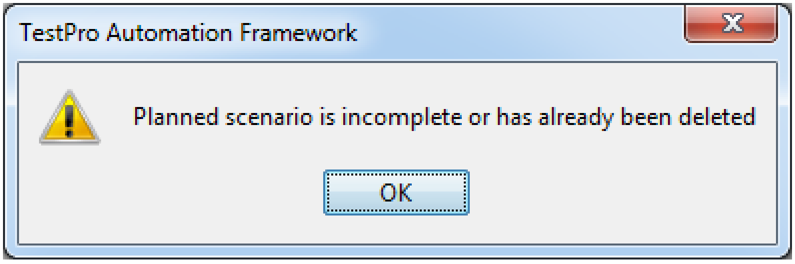
Select the scenarios to be used in the Execution Suite.

Scenarios can be selected individually by clicking the ‘Selected’ box and then the Add button or by clicking on the row and right clicking to get the Add option.

Multiple rows can be selected by clicking multiple selected rows or by using standard Windows Ctrl -> Click or Shift-> functions to select rows. Once the rows are selected click the Add button or Right click and select Add to add them to the Execution Suite.

When the selected Scenarios have been added to the Execution Suite the List Scenarios window remains open displaying the unselected scenarios to allow more scenarios to be selected.

Only completed Scenarios can be added to an Execution Suite. If a scenario is incomplete the following error message will be displayed:



Planned scenario is incomplete or has already been deleted

If multiple scenarios have been selected when this error occurs, none of the scenarios will be added to the Execution Suite.

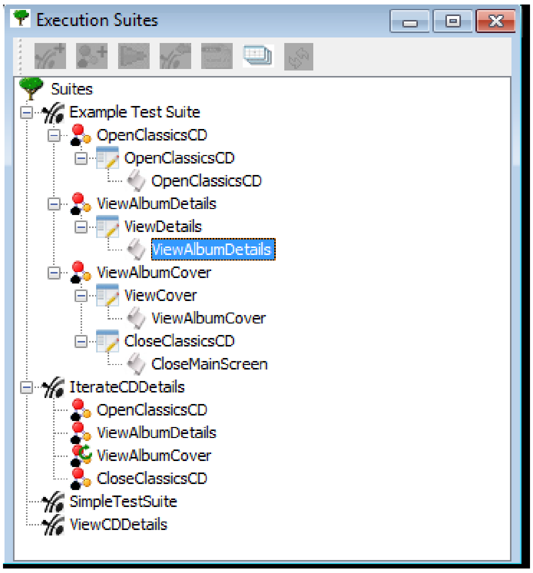
Close the List Scenarios window when all scenarios have been selected.

If the scenarios are not listed in the required order they can be reorganized using the functions described in the Modify Scenarios section.

8.2.3 Add Data Groups

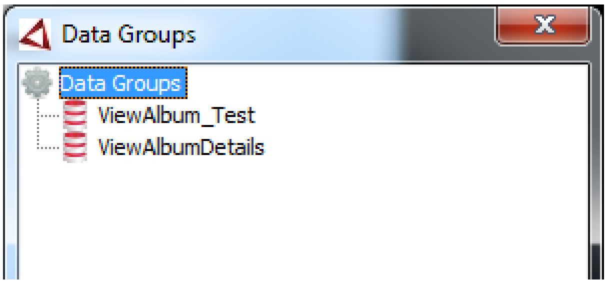
Each Action in a scenario must be associated with a Data Group. The Data Group provides the variable values used in the Action. The Data Group must be based on the current Data Template. Data Templates may change if the Test Tool script is changed:

1. ***Expand*** the Execution suite to the Action level;
2. ***Click*** on the action to highlight it***;***
3. ***Right click*** and select ***Add Data Group;***
4. The available Data Groups for the action selected are displayed is in the Data Groups window;
5. ***Click*** on the Data Group name;
6. ***Right click*** and click ***Select*** to attach the Data Group to the action;
7. The selected Data Group is displayed below the Action & the message “Data Transfer completed” is displayed in the status bar;
8. Multiple Data Groups can be added to an action where multiple data sources are required or available for the test. At Test Execution where multiple Data Groups are added to an action, all data applicable to the test will be exercised as if a single large Data Group had been added. The capability to add multiple Data Groups allows several sources to be used without having to first consolidate the data.



Once Data Groups have been added to all actions the Execution Suite creation is complete.

8.2.4 Use drag and drop to reorder Scenarios and Data Groups.



Drag and drop can be used to reorder Scenarios and Data Groups in an Execution Suite. Highlight the object you wish to move then drag it to its new position and release. You will not be allowed to release the object in an incorrect place. You cannot drag an object to the last position in a list. You must drag it to second last and then move the currently-last object up the list.

8.3 Modify Planned Scenarios

Execution Suites use a set of scenarios created in the Planned Scenarios window. Scenarios are designed to be used in more than one Execution Suite. If you wish to modify a Planned Scenario you should consider whether other users will be effected and whether it is more appropriate to create a new Scenario. This is particularly important if you have multiple environments with shared Execution Suites.

If you modify a Planned Scenario you must **Regenerate** the Scenario in the Execution Suite window to make the Execution Suite up-to-date.

8.3.1 Add a Scenario

A scenario can be added to an existing Execution Suite using the Add Scenario options. These are explained more fully in the previous section. The scenario will be added at the end of the list of scenarios. Once added it will probably be necessary to move the scenario using drag and drop. It will definitely be necessary to add a Data Group as described in the previous section.

8.3.2 Insert Scenario

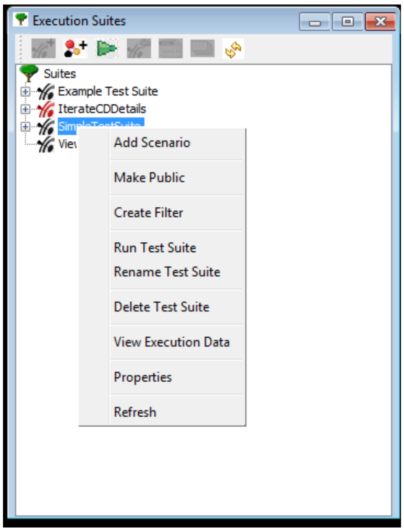
A scenario can be inserted into the Execution Suite by selecting the scenario after the insertion position and selecting Insert Scenario from the options menu. Data Groups will then need to be added to the Action(s) to complete the Execution Suite.

8.3.3 Delete Scenario

Scenarios can be deleted from an Execution Suite by right clicking on the Scenario name and selecting Remove Scenario from the menu. The scenario is not removed from the list of Planned Scenarios.

8.4 Importing Execution Suites

8.4.1 Public and Private Execution Suites



Execution Suites can be moved between environments or between users in the same environment by importing them. Only **Public** Execution Suites can be imported. The user who creates the Execution Suite is required to make it public using the Make Public menu option displayed when the user right clicks on the Execution Suite name.

Public Execution Suites have a red icon to differentiate them from Private ones which have a black Icon.

A Public Execution Suite can be made Private by selecting the Make Private menu option which appears if the selected Execution Suite is already Public.

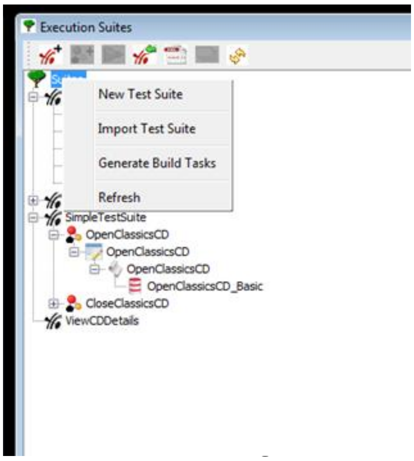
8.4.2 Considerations when Importing an Execution Suite

Importing an Execution Suite will import the Planned Scenarios that it incorporates, plus the Data Groups and filters, if selected. The import Suite Function will automatically register the Automated Test Tool scripts in the local environment

An Execution Suite can be imported multiple times, provided certain limitations are considered:

- If Planned Scenarios already exist they will be updated provided that no test cases have been replaced or renamed. If they have been replaced or renamed the import will fail;
- The Execution Suite Data Groups imported will replace existing Data Groups regardless of the import data settings in your preferences;
- All filters for the Execution Suite are deleted before the import is commenced regardless of whether filters are imported.

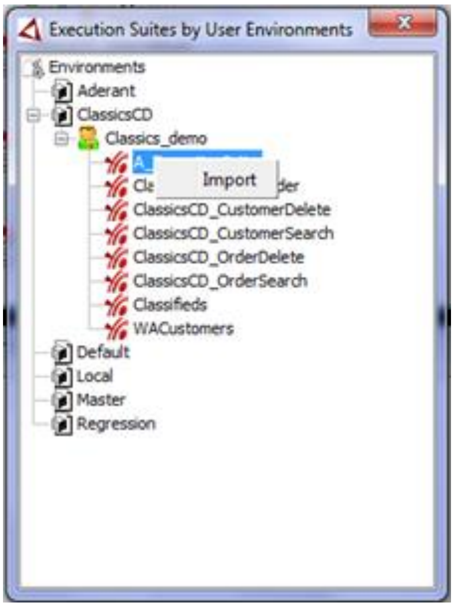
8.4.3 Steps for Importing the Execution Suite



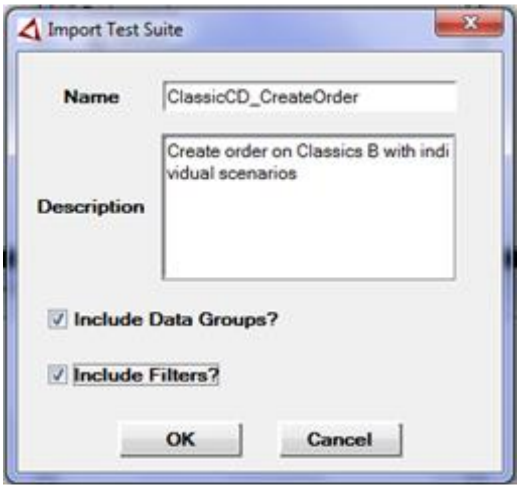
To import an Execution Suite you must know the environment it exists in and the user who has made it public.

1. ***Click*** on the Root Node of Execution Suites Window;
2. ***Right click*** and select ***Import Execution Suite;***
3. The **Execution Suites by User Environment Window** is displayed;
4. ***Click*** on the Environments Symbol to show the list of environments;

5. *__Click__* on the required environment to display the different user names within the environment;
6. *__Click__* on the user name icon to display the list of available Execution Suites. Only the Public Execution Suites owned by that user will be displayed.



7. Click on the required Execution Suite
8. Right Click and select Import
9. The Import Execution Suite window is displayed. The data displayed defaults to the selected Execution Suite Name and description, as shown below.


Import Test Suite

Name: ClassicCD_CreateOrder

Description: Create order on Classics B with individual scenarios

☑ Include Data Groups?

☑ Include Filters?

10. The user can change the Execution Suite Name and Description prior to import and can optionally include Data Groups and any Filters that are on the Execution Suite being copied.
11. Check the Status Bar for the progress and completion of the Import function.

8.5 Additional Functionality available in the Execution Suites window

Various additional Functions and shortcuts to other processes are available from within the Execution Suite window.

8.5.1 Create a new Data Group from an Action node

The user can create a Data Group by selecting the Create Data Group option from the Action node menu.

The process is similar to that followed in the main menu option Data ➔ Import ➔ Import Action Data however the data can only be imported from a spreadsheet not from the Shared data area.

8.5.2 Create a Filter from an Execution Suite node

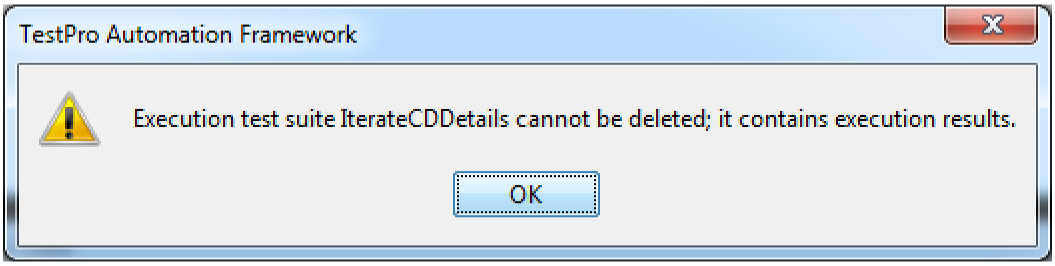
The user can create a Filter by selecting Create Filter from an Execution Suite node.

The Process is similar to the process used in the Test Filters window.

8.5.3 Delete Execution Suite

An Execution Suite can only be deleted when there are no execution results associated with it. To Delete the Execution Suite, right click on the Execution Suite name and select "Delete Execution Suite".

If execution Results are associated with the Execution Suite the following error message will be displayed.


TestPro Automation Framework

Execution test suite IterateCDDetails cannot be deleted; it contains execution results.

8.5.4 Insert Data Group from Data Group node

Additional Data Groups can be added to an Action by selecting the Insert Data Group option from a Data Group node. The Data Group is inserted above the selected node.

8.5.5 Refresh

Refresh is available in many places in TAF Pro. It updates the selected item against the current version in the data base. This ensures that any recent changes not displayed on the screen are integrated into the current process.

8.5.6 Regenerate Scenario

After a Scenario is used in an Execution Suite any changes made to it need to be updated to the Execution Suite. The Regenerate Scenario function has been provided so the user does not have to remove then re-add the Scenario to adopt the changes. Until Scenarios are regenerated, any changes made to them will not be recognized in an existing Execution Suite.

Select the scenario, right click and select the **Regenerate Scenario** option.

8.5.7 Remove Data Group from Data Group node

A Data Group can be removed from an action by selecting **Remove Data Group** on the options menu of the Data Group.

8.5.8 Remove Scenario

A Scenario can be removed by right clicking on the scenario and selecting **Remove Scenario** from the options menu.

8.5.9 Rename Execution Suite

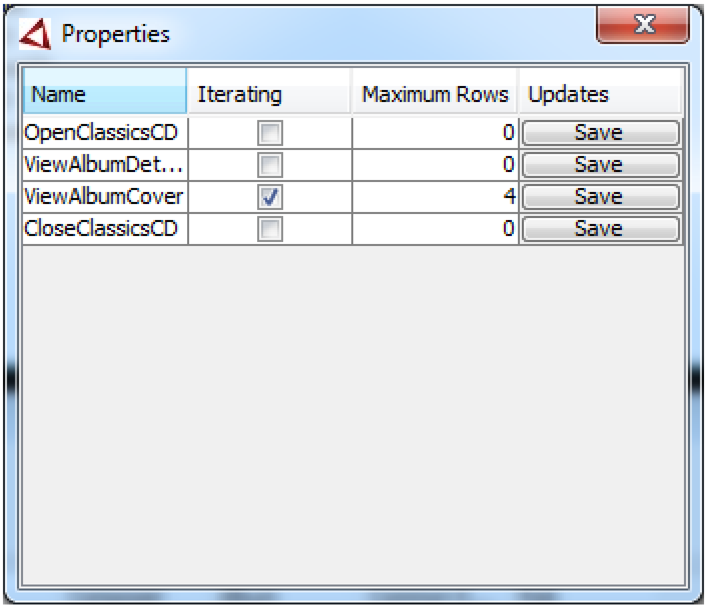
The rename function can be selected by right clicking on the Execution Suite and selecting the **Rename Execution Suite** option. The rename only applies to the current environment. Follow the simple steps to rename the Execution Suite. You are required to provide a reason for renaming the Execution Suite because of the potential loss of continuity to results, Command Line Interface Batch Jobs, Test Filters and Test Results that are linked to the Execution Suite name. Renaming has the following impact on these functions:

- Command Line Interface: Any Build Files that use the Execution Suite will no longer function as they will not locate the Execution Suite;
- Execution Results: Existing Execution Results will continue to be listed under the previous Execution Suite name. New Runs will be be listed under the new Execution Suite Name;
- Filters: The Execution Suite name in the filters window will be changed to the new name;
- Delete Execution Suite: A renamed Execution Suite can be deleted before any Tests are executed.

8.5.10 Run Execution Suite

The user can run an Execution Suite from within the Execution Suites window by selecting the Execution Suite and then right clicking to select **Run Execution Suite** from the options menu. The process is fully explained in the chapter on Executing Tests.

8.5.11 Simple Iteration



Properties

Name	Iterating	Maximum Rows	Updates
OpenClassicsCD	☐	0	Save
ViewAlbumDet…	☐	0	Save
ViewAlbumCover	☑	4	Save
CloseClassicsCD	☐	0	Save

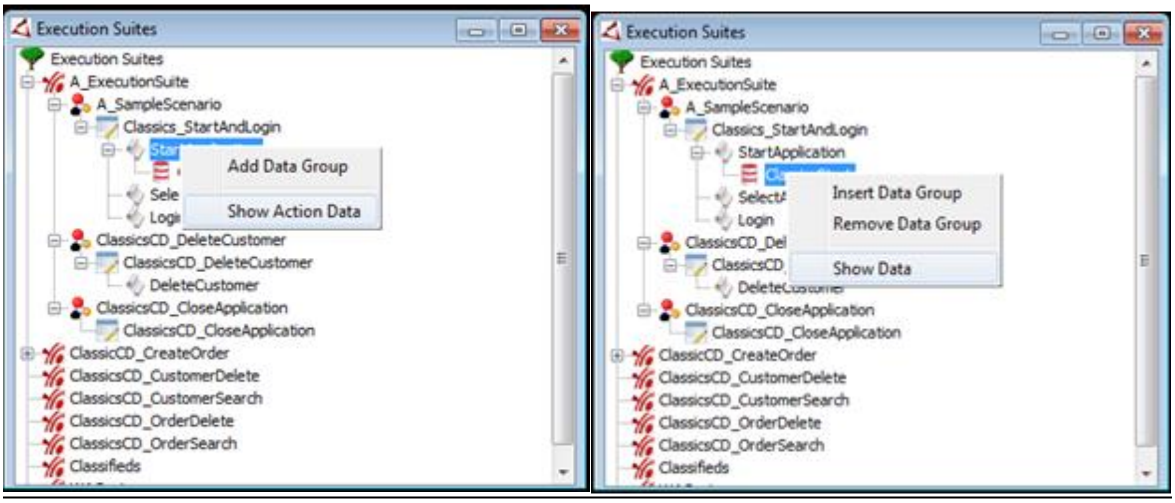
Simple Iteration is available by selecting **Properties** from the Execution Suite options menu. The Properties box shows all the Scenarios in the Execution Suite in order. The Scenarios that iterate are ticked in the Iteration column and the maximum rows to iterate is shown in that column. Changes can be made at this level and must be saved for each scenario that is altered. A full explanation of Iteration is provided in the section “Final Preparation for Testing”.

8.5.12 View Data at the Action Level

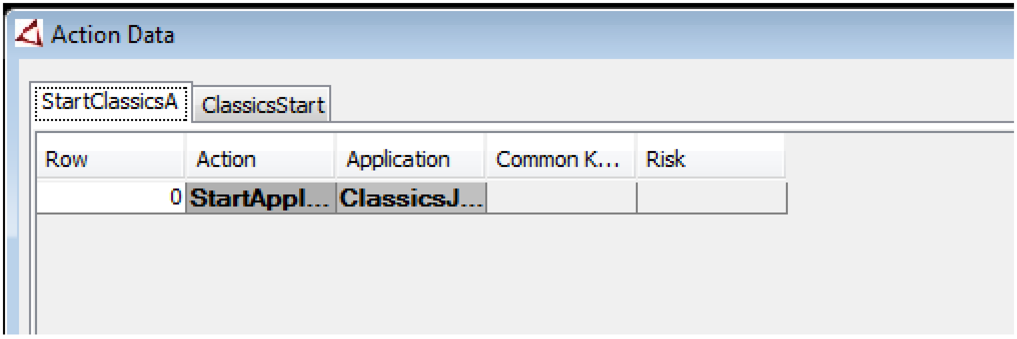
When a Data Group is selected in the Execution Suites window the first few lines of data are displayed in the Action Data window.

The number of lines displayed initially is set in Preferences (See section 15.2.5 - Preferences ➔General ➔ Maximum Action Data - Auto Display).

To see the full data set click the Show ALL icon or right click on the Action or Data Group and select the **Show Action Data** or **Show Data** option, as appropriate.



All the data in the Data Group is now visible in the Action Data window for the action selected. Where there are multiple Data Groups attached to an Action each Data Group is shown under its own Tab View as shown below. Click on the Tab to view each Data Group.



Action Data

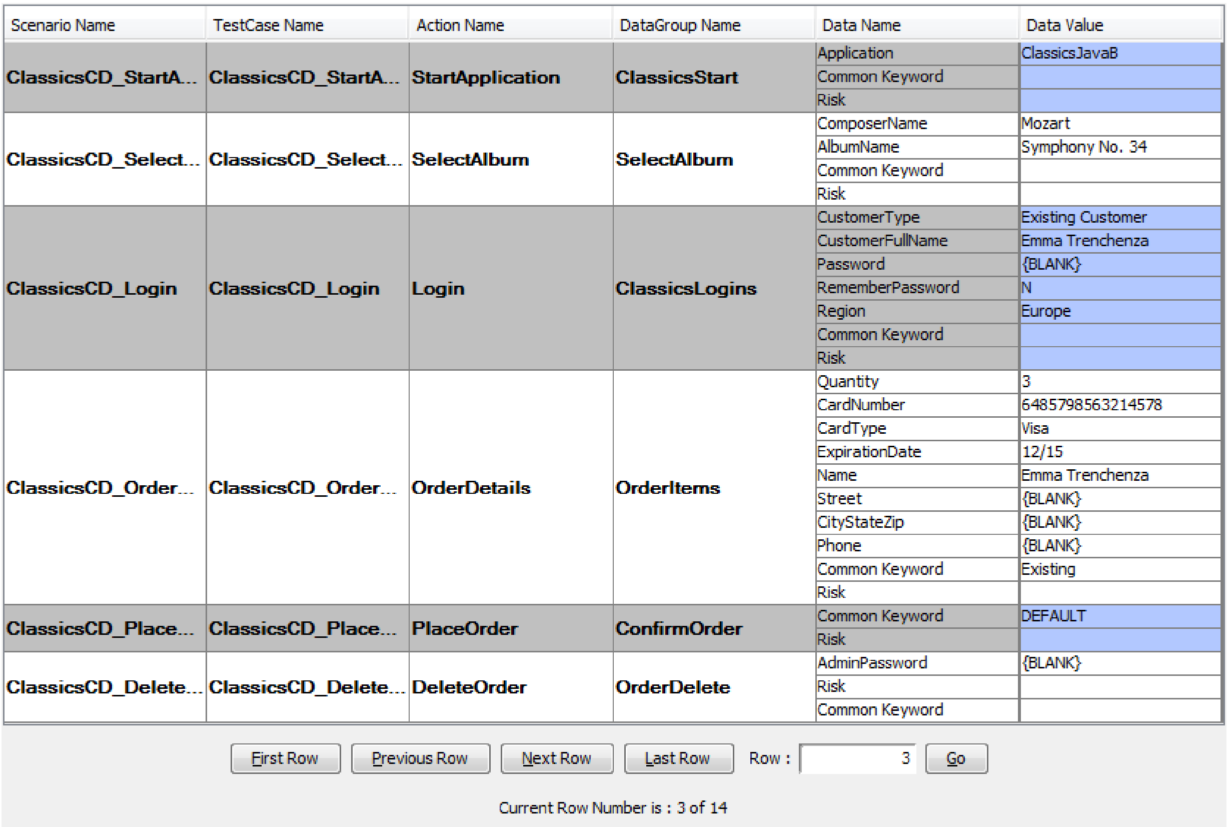
StartClassicsA | ClassicsStart

Row	Action	Application	Common K...	Risk
0	StartAppl...	ClassicsJ...		

8.5.13 View Data at Execution Level

Data within an Execution Suite can be quite detailed. Often with many Data Groups attached to actions the full sequence of data can be hard to determine. View Execution data provides the ability to review the data for all the actions in the execution suites in a single view.

1. ***Click*** on an Execution Suite to select it
2. ***Right click*** and select ***View Execution Data.*** The Execution Data Window is displayed as shown below:



Scenario Name	TestCase Name	Action Name	DataGroup Name	Data Name	Data Value
ClassicsCD_StartA...	ClassicsCD_StartA...	StartApplication	ClassicsStart	Application	ClassicsJavaB
				Common Keyword	
				Risk	
ClassicsCD_Select...	ClassicsCD_Select...	SelectAlbum	SelectAlbum	ComposerName	Mozart
				AlbumName	Symphony No. 34
				Common Keyword	
				Risk	
ClassicsCD_Login	ClassicsCD_Login	Login	ClassicsLogins	CustomerType	Existing Customer
				CustomerFullName	Emma Trenchenza
				Password	{BLANK}
				RememberPassword	N
				Region	Europe
				Common Keyword	
				Risk	
ClassicsCD_Order...	ClassicsCD_Order...	OrderDetails	OrderItems	Quantity	3
				CardNumber	6485798563214578
				CardType	Visa
				ExpirationDate	12/15
				Name	Emma Trenchenza
				Street	{BLANK}
				CityStateZip	{BLANK}
				Phone	{BLANK}
				Common Keyword	Existing
				Risk	
ClassicsCD_Place...	ClassicsCD_Place...	PlaceOrder	ConfirmOrder	Common Keyword	DEFAULT
				Risk	
ClassicsCD_Delete...	ClassicsCD_Delete...	DeleteOrder	OrderDelete	AdminPassword	{BLANK}
				Risk	
				Common Keyword	

Current Row Number is : 3 of 14

3. Use ***First Row / Last Row / Next Row / Previous Row*** keys to ***navigate*** through the execution suite data

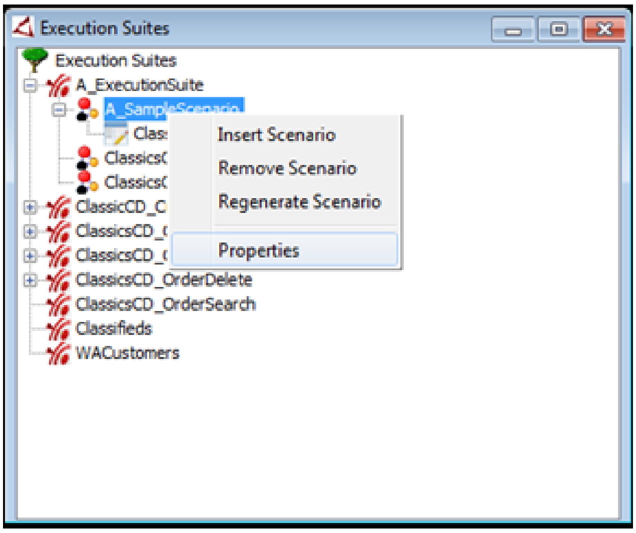
4. The required ***row number*** can as be entered in the ***Row prompt*** to navigate to an exact row to view

The Data is displayed in order of Scenario – Test Case – Action – Data Group Name. The user can move through the Data Group using the navigation buttons or jump to a specific row to check the data.

8.6 Final Preparation for testing

The Scenario Properties panel provides the facility to manage error handling within the scenario and to provide for Iteration within the scenario. The Iteration process is described in the next section.

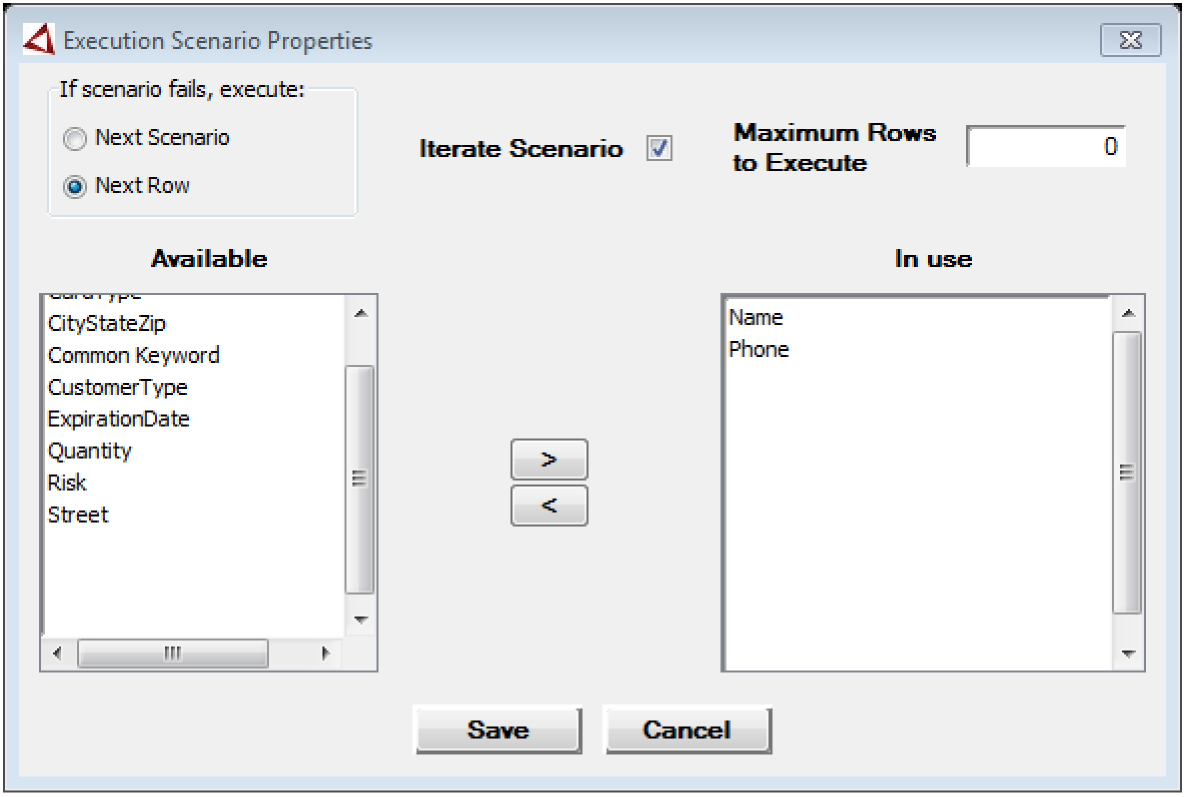
Open the Scenario Properties panel by right clicking on the scenario and selecting **Properties**.



8.6.1 Error Handling

There are two choices available if the test returns an error, either the complete row is terminated and the test continues with the next row, or the test continues with the next scenario. This choice will be affected by how the tests are written and the impact of a scenario failure.

Select the appropriate Radio button to determine the desired action.



8.6.2 Set up Iteration

Iteration is used when there is a requirement to repeat a particular scenario within the test row. As an example, if the test creates an order it may require that several line items are added to the order.

Select the radio button “Iterate Scenario” to confirm that iteration is required and then enter the maximum number of rows to be selected that meet the iteration requirements. As with other situations 0 (zero) means ‘Select all’ which should be avoided in this instance.

Fields displayed in the “Available column” can be selected, by highlighting them and clicking the Right Arrow, to determine which rows will be considered. The fields available consist of data variables in the current scenario plus any values generated during previous scenarios that have been saved using the SetOutput() function for later use.

In the example shown the fields Name and Phone will be used for iteration, as displayed in the ‘In use’ column. Only Data Group lines that contain the same Name and Phone, used in a previous scenario will be selected. If no fields are selected iteration will use rows sequentially regardless of their content. If no rows match the selected data a blank row will be assumed and the test will continue successfully.

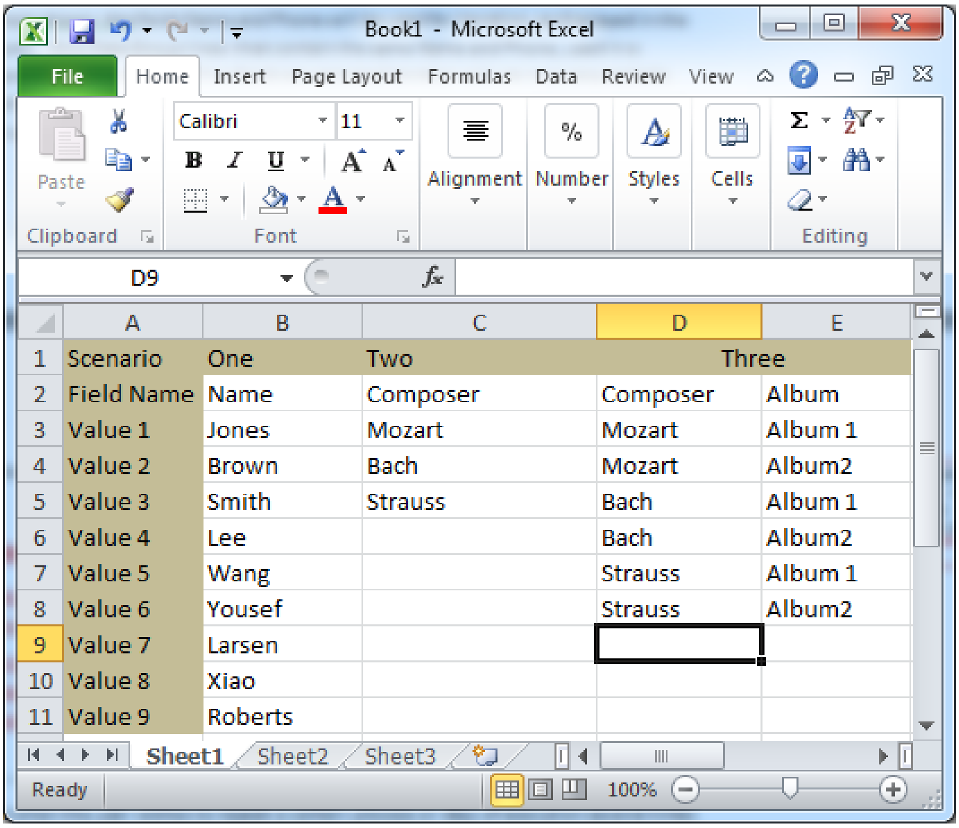
Click **Save** to keep the Settings or **Cancel** to ignore changes.

8.7 Understanding Iteration

8.7.1 Understanding a Standard Run

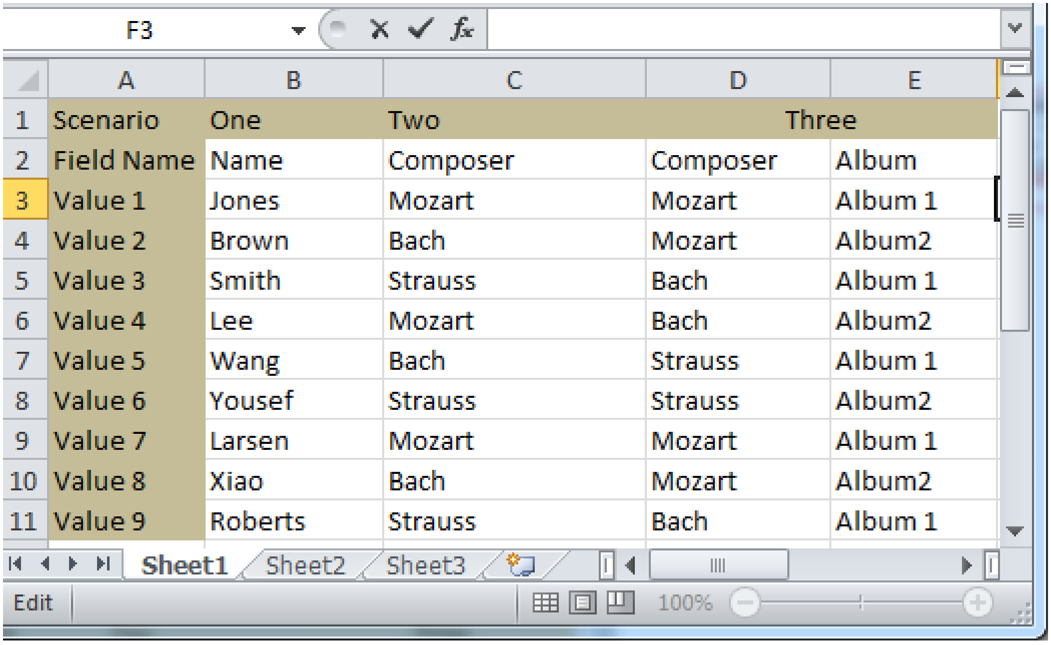
When a TAF run is set up the data to be used is selected from the various Data Groups by applying the filter selected to each data group. If no filter is used all data is selected.

Displayed is a simple set of three scenarios with 9 consumers, three composers and 6 albums



	A	B	C	D	E
1	Scenario	One	Two	Three	
2	Field Name	Name	Composer	Composer	Album
3	Value 1	Jones	Mozart	Mozart	Album 1
4	Value 2	Brown	Bach	Mozart	Album2
5	Value 3	Smith	Strauss	Bach	Album 1
6	Value 4	Lee		Bach	Album2
7	Value 5	Wang		Strauss	Album 1
8	Value 6	Yousef		Strauss	Album2
9	Value 7	Larsen			
10	Value 8	Xiao			
11	Value 9	Roberts			

The number of rows to be built is determined by the maximum number of remaining rows in a Data Group. Every row has data from every Data Group. Smaller Data Groups loop recursively through their lines of data to complete each row. So the following rows would be created from the above data.



	A	B	C	D	E
1	Scenario	One	Two	Three	
2	Field Name	Name	Composer	Composer	Album
3	Value 1	Jones	Mozart	Mozart	Album 1
4	Value 2	Brown	Bach	Mozart	Album2
5	Value 3	Smith	Strauss	Bach	Album 1
6	Value 4	Lee	Mozart	Bach	Album2
7	Value 5	Wang	Bach	Strauss	Album 1
8	Value 6	Yousef	Strauss	Strauss	Album2
9	Value 7	Larsen	Mozart	Mozart	Album 1
10	Value 8	Xiao	Bach	Mozart	Album2
11	Value 9	Roberts	Strauss	Bach	Album 1

During a standard run these nine rows (3 – 11) will be stepped through sequentially and processed individually.

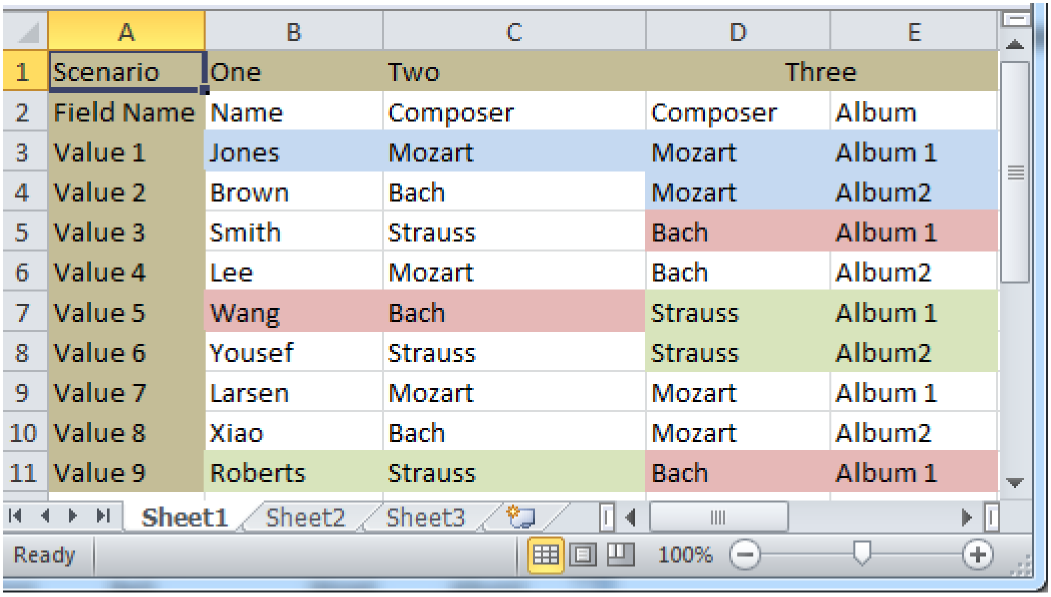
8.7.2 Understanding an Iterating run

If we set this data up so that scenario three iterates twice on Composer then, when we get to scenario three, the selection of the appropriate lines changes as shown in the illustration below.

- User Jones (line 3) selects composer Mozart and then selects the two Mozart albums on lines 3 and 4.

- User Wang (line 7) selects Bach and selects the Bach album on line 11 and then returns to the start of the scenario three data group and selects the first Bach album on line 5.

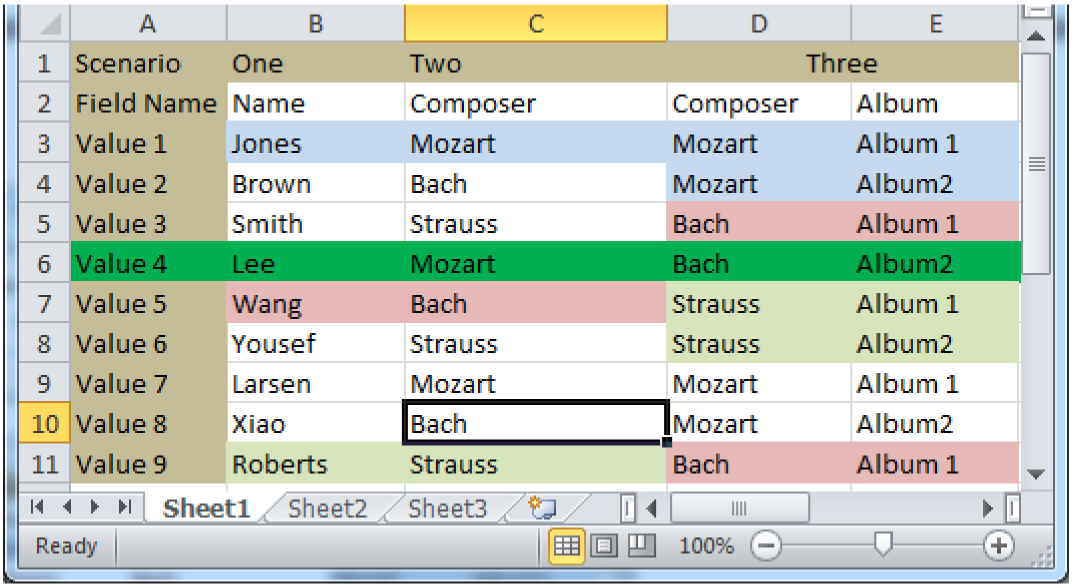
- User Roberts (line 9) selects Strauss and, because there are no Strauss albums in the remainder of the scenario three Data Group, starts searching at the beginning of the scenario and selects the Strauss albums in line 7 and 8.



	A	B	C	D	E
1	Scenario	One	Two	Three	
2	Field Name	Name	Composer	Composer	Album
3	Value 1	Jones	Mozart	Mozart	Album 1
4	Value 2	Brown	Bach	Mozart	Album2
5	Value 3	Smith	Strauss	Bach	Album 1
6	Value 4	Lee	Mozart	Bach	Album2
7	Value 5	Wang	Bach	Strauss	Album 1
8	Value 6	Yousef	Strauss	Strauss	Album2
9	Value 7	Larsen	Mozart	Mozart	Album 1
10	Value 8	Xiao	Bach	Mozart	Album2
11	Value 9	Roberts	Strauss	Bach	Album 1

The start position for each user in each scenario is the original line in the Data group. In the example we have here User Lee commences on line 4 in every Data Group including scenario 3. This is despite the fact that user 1 will have read lines 1 & 2 in scenario 3, user 2 will have read lines 3 & 4 in scenario 3 and user 3 will have read lines 5 & 6 in scenario 3.

The starting values for user 4 are highlighted in green below.



	A	B	C	D	E
1	Scenario	One	Two	Three	
2	Field Name	Name	Composer	Composer	Album
3	Value 1	Jones	Mozart	Mozart	Album 1
4	Value 2	Brown	Bach	Mozart	Album2
5	Value 3	Smith	Strauss	Bach	Album 1
6	Value 4	Lee	Mozart	Bach	Album2
7	Value 5	Wang	Bach	Strauss	Album 1
8	Value 6	Yousef	Strauss	Strauss	Album2
9	Value 7	Larsen	Mozart	Mozart	Album 1
10	Value 8	Xiao	Bach	Mozart	Album2
11	Value 9	Roberts	Strauss	Bach	Album 1

8.7.3 Points to Note

- The Execution Suite will only loop through a data group once to locate appropriate values.
- A blank row is assumed if no values are found.
- Iteration should not be applied to the first scenario.

- Iteration should only be used for the purpose described. The test execution results include a line for each row. Overloading a row by using inappropriate iteration will create test results that are difficult to understand.

9 Using Filters

9.1 Overview

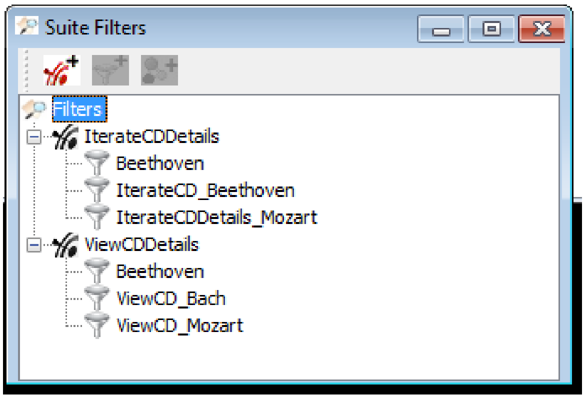
Test Suite Filters are used to limit the number of tests executed when the suite is run to those relating to a particular set of data. This produces a shorter and more targeted test exercise. Test Suite Filters are displayed in the Suite Filters window.

One of the most common uses of filters is to restrict the test run to a particular test environment, browser or user type. Once the test suite is configured it can be used in a wide range of environments. By putting fields for these variables in the Data Group for the commencing action a filter can be built that logs-in the required user to a specific environment using a particular browser. Another example would be a situation where one area of your data has changed and you need to ensure that the changes are reflected in all environments.

Test Suite Filters can be used in all Test Suite Execution situations. The Build Files used for command line execution of tests use filters in the same way they are used in the GUI interface. When using Test Suite Filters in Build Files it is advisable to generate the Build Files and use filters in a dedicated ‘production’ environment that is not modified on a regular basis. Each generated build file for a Test Suite can use a different filter making it easy to build a set of automated tests for different situations.

9.2 Creating a Test Suite filter

9.2.1 Enter filter name and description

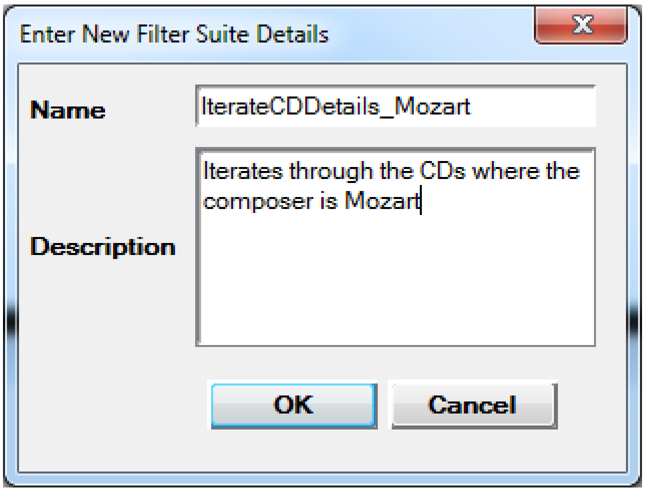


There are four ways to create a new Test Suite Filter:

1. Click on top node of the Filters tree (Filters), click on **Add a Suite** icon once it is activated then, select one of the suites in List Test Suites window, right click on the suite and select **Add**;

2. Right click on top node of the Filters tree (Filters), select the **Create Filter** option. then, select one of the suites in **List Test Suites** window, right click on the suite and select **Add;**
3. Right Click on an existing Test Suite listed in the Suite Filters window and select **Create Filter;**
4. In the Execution Suites window Right Click on a Test Suite and select **Create Filter.**

Any of these will cause the **Enter New Filter Suite Details** window to display.



Name: IterateCDDetails_Mozart

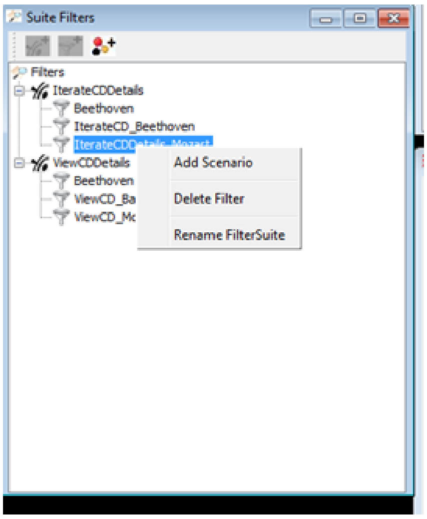
Description: Iterates through the CDs where the composer is Mozart

Enter the name of the Filter. Typically add an extension to the name or abbreviation of the Test Suite.

Enter a detailed description of what the Filter does. This should provide another user with a sufficient description so that they do not need to examine all the details to find out.

Click OK to complete the description. The filter will appear in the Suite Filters window as shown above.

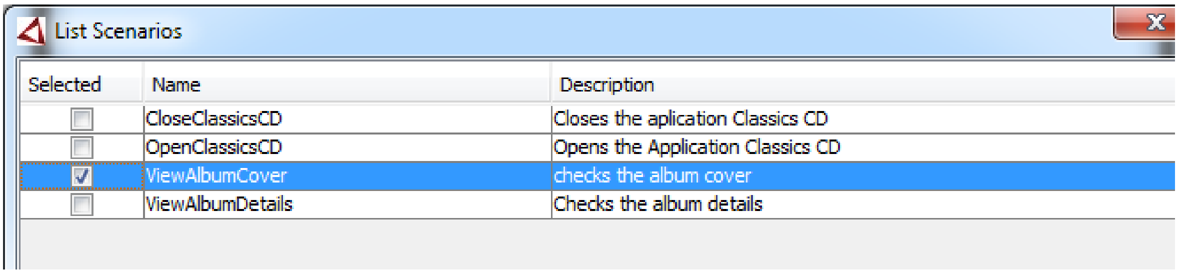
9.2.2 Select the scenario(s) to use



Highlight the Filter Name and either select the **Add a scenario** icon or Right click on the Filter and select **Add Scenarios**

A list of the scenarios in the Test Suite is displayed. Select only those that you wish to use in your filter. All scenarios selected MUST have a filter built for them. If a selected scenario does not have a filter defined the filter is considered incomplete and is not available when executing the Test Suite.

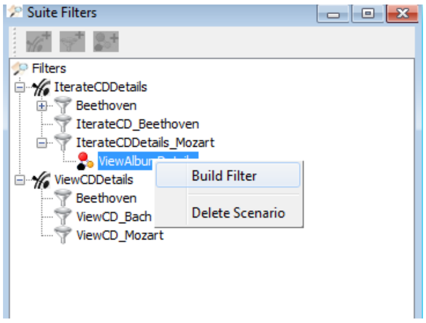
Once you have selected the scenarios required click the Add button or right click on the selected scenario(s) and click on the **Add** option. Once the required scenario is selected, close the List Scenarios window.



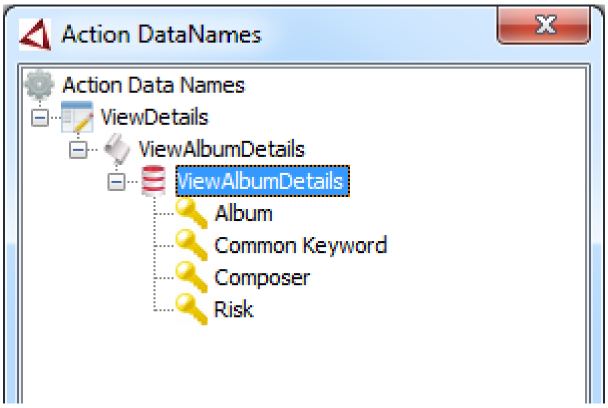
Selected	Name	Description
□	CloseClassicsCD	Closes the aplication Classics CD
□	OpenClassicsCD	Opens the Application Classics CD
☑	ViewAlbumCover	checks the album cover
□	ViewAlbumDetails	Checks the album details

9.2.3 Build the filter for each Scenario

The selected Scenario is added to the new Suite Filter:

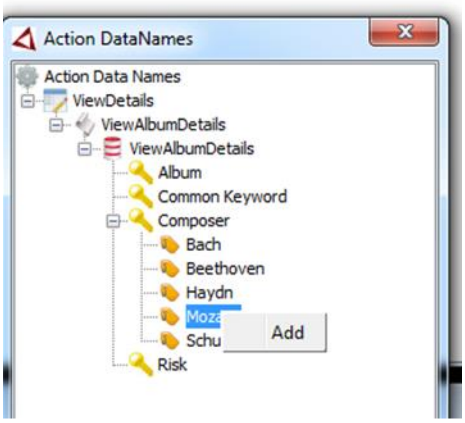


1. Right Click on the Scenario and select **Build Filter;**



2. The Action Data Names window is displayed. Expand the nodes until the field names for the Data Group are displayed.

3. Click on the field name(s) required to display the values available in that field.



4. Create the filter detail by selecting the values that you wish to include in your filter, then right click and Add them to the filter.

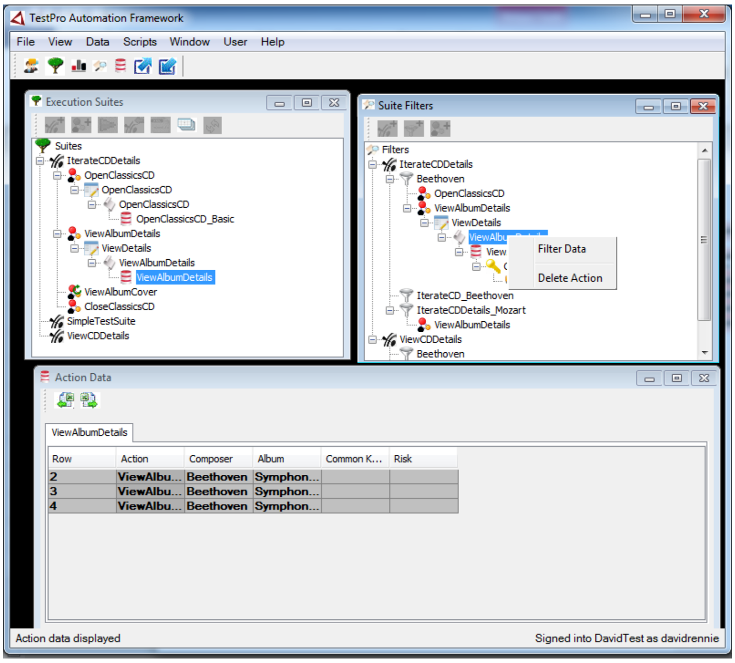
5. Multiple selection of values is available by using CTRL or Shift keys.

Multiple fields within the same Data Group can be selected. Negative filtering is not available.

Repeat this process for each Scenario, Action Data Group or Field that you wish to include in the filter.

9.2.4 View Filtered Data

Once the filter is created, the data can be checked in the Action Data window.



Row	Action	Composer	Album	Common K...	Risk
2	ViewAlbu...	Beethoven	Symphon...		
3	ViewAlbu...	Beethoven	Symphon...		
4	ViewAlbu...	Beethoven	Symphon...		

Action data displayed

Signed into DavidTest as davidrennie

Select the Action or Data Group in the Execution Suites window. Right click and select **Show Data**. The action data is displayed in the Action Data window.

In the Suite Filters window, right click on the Action and select Filter Data.

The Action Data display is reduced to those rows that will be selected when this action is used. (Only the filtered data is displayed)

9.3 Additional Functionality in the Suites Filters window

9.3.1 Rename Filter

The filter can be renamed by right-clicking on the Filter name and selecting **Rename Suite Filter**. When renaming, the user is required to enter a reason.

If a filter is used in a generated build file the build file will no longer work when the filter name is changed. Renaming a filter in the production environment should be avoided for this reason.

9.3.2 Modify Filter

There are two ways to modify a filter. Any node can be deleted to remove all selections below that level as described in delete node.

A filter can be added to, by repeating the Build Filter process described above. Select additional values in fields to adjust the filter. Existing values are retained unless deleted.

9.3.2.1 Delete Node

All nodes, except the root, in the Suites filter can be deleted by right-clicking on the node and selecting **Delete**. The node and all information below the node are removed from the filter. It is important to remember that every node must have a selected value in a field to be complete. If a node cannot be expanded to a selected value it will have to be deleted before the filter will function.

10 Executing Tests

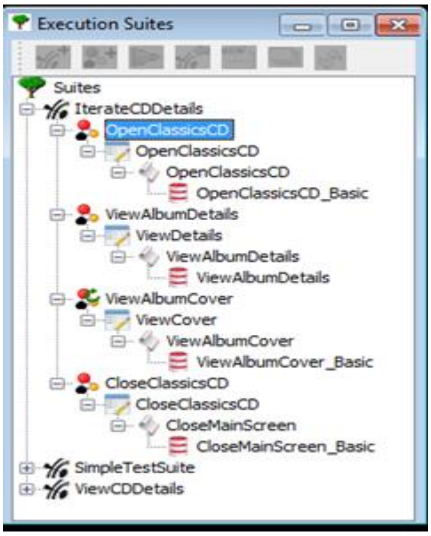
10.1 Overview

Test Suites developed in TAF Pro can be executed in a number of ways.

- The test suites can be run from the TAF application. This gives the user a great deal of flexibility when running the test.
- The test suites can be also run using generated Build Files. There are three methods of running Test Suites using generated Build Files:
  - TAF Pro **Command Line Interface** tool: This method uses the PC interface so only a single job can be run per machine. This method uses the generated build file; the tester is not required to log in to TAF to simply run the test;
  - Batch file calling Ant: The test suites can be run by executing a batch file that calls the build file. This is effectively the same as using the TAF Pro CLI interface tool and is primarily used to test the setup of a scheduled run;
  - Scheduled run using a scheduling tool: A tool such as Windows Scheduler can be used to initiate the batch file at an appropriate time. Depending on the scheduling tool used, it can be possible to run several jobs simultaneously using this method.

The methods using Build Files do not allow the tester to alter the run details and therefore they provide a more consistent set of results when the test are repeated over a period of time. The test designer can modify the Test Suite and Filters within TAF Pro without the need to regenerate the build file, provided the Test Suite and Filters are not renamed and not removed. When using any of these techniques the results are stored in, and available from the TAF Pro datastore.

10.2 Test Execution with the TAF Pro GUI Interface



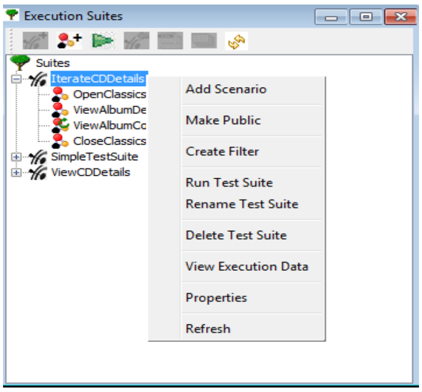
TAF Pro Test Suites can be run from the Execution Suites window.

TAF Pro acts as a driver for the underlying scripts in a testing tool such as IBM Rational Functional Tester using TAF Pro Actions.

Actions are created and developed using the testing tool and imported when scripts are registered. Once a script is registered, modifications can be automatically incorporated into the run if required by using the run options.

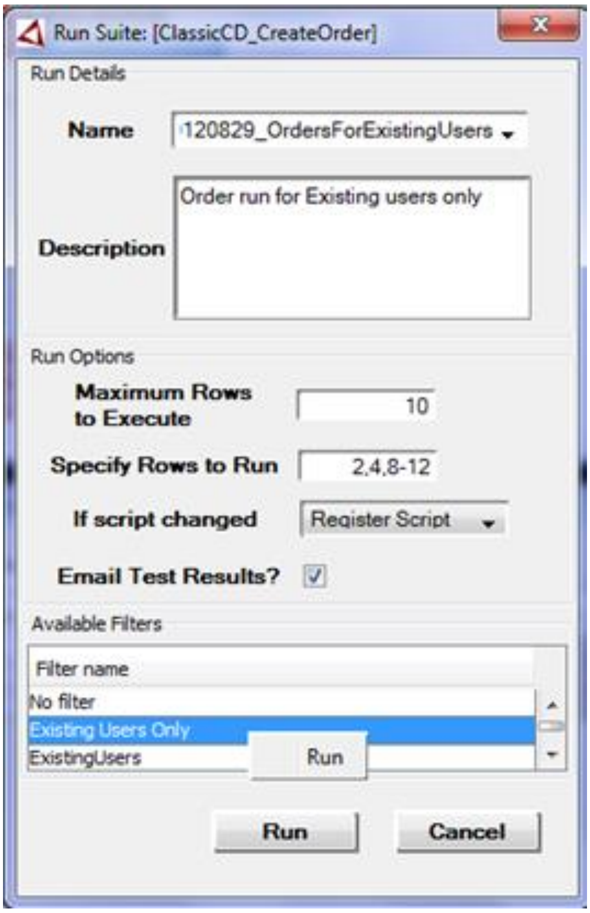
TAF Pro controls the Test Suite execution so that errors are captured in the Results and subsequent tests can continue to run until the entire suite is completed. Run options control how a failed test is handled.

TAF Pro ensures that each Test Suite run is executed to completion.



The steps to execute a test from the Execution Suite window are:

1. ***Click*** on the test suite to be run;
2. Select the ***Run Test Suite*** icon or ***Right Click*** and choose ***Run Test Suite*** as shown below.



Run Suite: [ClassicCD_CreateOrder]

Run Details

Name 120829_OrdersForExistingUsers

Description Order run for Existing users only

Run Options

Maximum Rows to Execute 10

Specify Rows to Run 2,4,8-12

If script changed Register Script

Email Test Results? ☑

Available Filters

Filter name
No filter
Existing Users Only
ExistingUsers

If the Test Suite is not completely configured appropriate error messages are displayed. The errors must be corrected before the run can commence. Any scenarios that are modified should be regenerated. The Test Suite should be refreshed before attempting the run a second time.

Enter the Run Details in the Run Suite window:

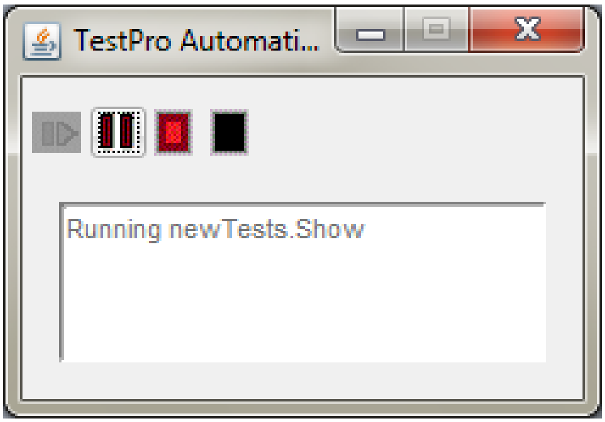
1. Run Name– The run name should differentiate this run based on the test suite and parameters used. It can only contain 45 characters and will often use abbreviations. The run name can be selected from previous runs of this Test Suite.
2. Run Description – This should include a more complete explanation of the parameters and filters used in this run.
3. Maximum Rows to Execute – Enter the number of rows to be executed. The default value for all runs is set in User Preferences. Set this to 0 to execute all rows in the data group(s).

4. Specify Rows to Run – If specific rows need to be tested enter the row numbers here. As with specifying pages, values are separated by commas, e.g. 1-5, 7, 9, 15-18 etc. If the number of rows specified is greater than the value for Maximum Rows to Execute, the test will stop when the value of Maximum Rows is reached. The values entered in this field will NOT override the parameter entered in the Maximum Rows to Execute field. Default to blank for all rows
5. If Script Changed–The user can nominate what action to take at run time if TAF Pro detects that an IBM Rational Functional Tester Script has changed since Registration of the Script. The user can nominate to:
    - Register Script – Re-registers the script and run the latest version;
    - Run Script – Run the script as is;
    - Cancel Run – Terminate the test.
6. Email Test Results:
    - ***Click*** on check box if results are to be emailed to the user;
    - The user has a choice to configure email settings specifically for this **Execution Suite** using this option. If email settings are not provided, TAF Pro will use, email settings set at **User Environment Level**. If User Environment Level settings are not configured, then **Environment Level** email settings will be applied or finally the **SMTP** settings if none other are supplied;
    - Refer to section this section number has to be changed once the numbering of the user guide happens for details on configuring Email Settings at **Execution Suite Level.**
7. Select Filters:
    - No Filter – Run the Execution Suite for all attached data without filters;
    - With Filter – Select the filter name to use according to the test required; highlight the specific filter and it will turn to blue.
8. ***Select Run*** to start the test.

10.2.1 Execution Controls within the GUI

10.2.1.1 TAF Pro Toolbar

When TAF Pro is executing the TAF Pro Automation Toolbar will be available.



Running newTests.Show

The TAF Pro toolbar controls the TAF processing and automatically initiates actions which are implemented using the Test Scripting application.

The text window displays the script currently being executed and will also indicate if TAF processing has been paused, about to stop or aborted.

The tool bar should be minimized when not required to avoid affecting the test.

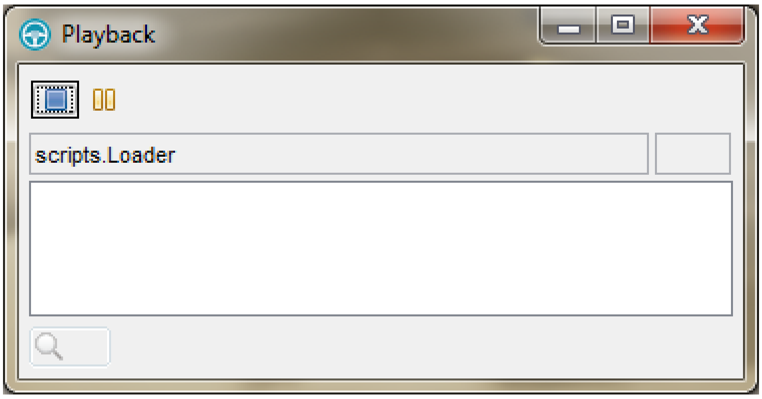
The buttons are:

- ***Resume*** – this button is activated when the Pause button is clicked.
    - Paused execution will be resumed.
- ***Pause*** – this button is activated on display and when the Resume button is clicked.
    - TAF Pro processing is paused.
- ***Stop*** – this button is activated on display.
    - When this button is clicked, the currently executing row will be completed before execution stops.
- ***Abort*** – this button is activated on display.
    - When this button is clicked, the currently executing script will be completed before execution stops.

If a toolbar button is clicked whilst a script is executing, the button operation will not be activated until the currently executing script completes.

This dialog automatically closes when test execution completes.

10.2.1.2 Rational Functional Tester toolbar



The RFT toolbar is available when RFT scripts are running. The Rational Functional Tester window controls RFT script execution.

Individual steps within the script are displayed within the window.

When a script is executing, clicking the Stop / Pause / Resume buttons will affect the currently executing script. Clicking the TAF Pro toolbar will have no effect until the RFT script completes!

If a TAF Pro toolbar button is clicked when an RFT script is NOT executing, the RFT toolbar functions will be inactivated until the TAF Pro toolbar Resume button has been clicked.

10.2.1.3 Stopping Test Suite Execution

Using the Stop button on the Script Execution Tool (e.g. RFT) toolbar will halt the script immediately. The TAF line will not complete.

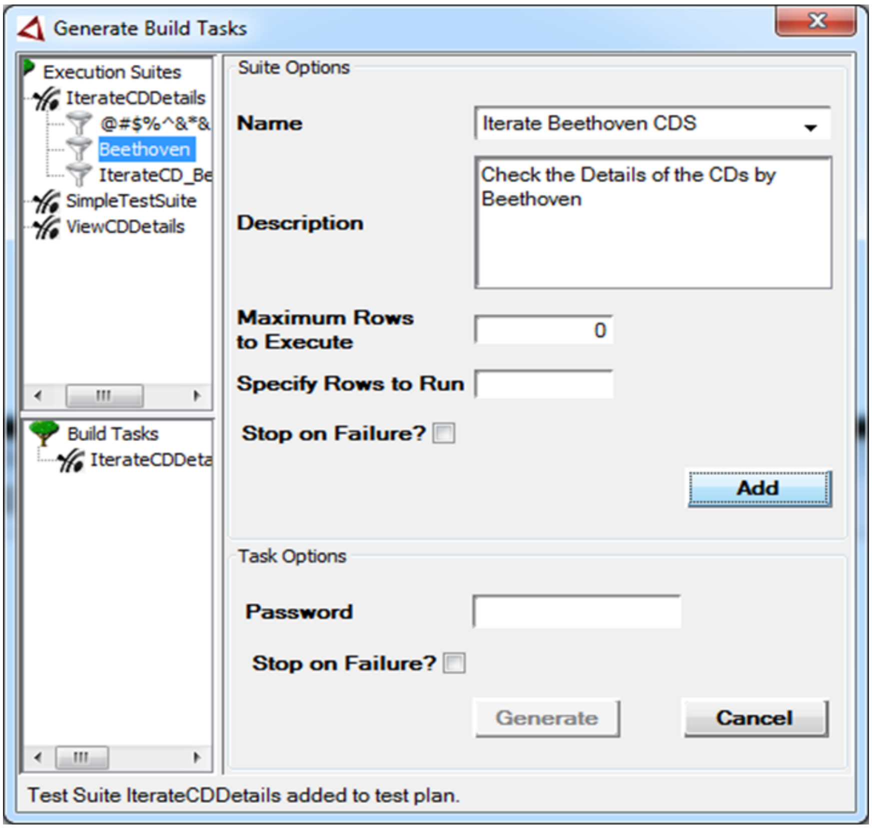
Using the Stop button on the TAF Pro toolbar will allow the TAF line to complete and then the Test Suite will stop executing.

10.3 Generating Test Plan and Build Files within the TAF GUI interface

The Command Line Interface (CLI) provides testers with the ability to run tests from the command line using a scheduling program or by manually executing a batch file.

The command line Build Files allow the inclusion of several test suites and multiple filters, so that a significant test exercise can be created to run independently.

The Build Files link to the TAF datastore so that test results, emails and test logs are captured in the same manner as when using the GUI Interface. When preparing the CLI files, all the run options available within the GUI are available.



Name: Iterate Beethoven CDS

Description: Check the Details of the CDs by Beethoven

Maximum Rows to Execute: 0

Specify Rows to Run:

Stop on Failure? □

Password:

Stop on Failure? □

Test Suite IterateCDDetails added to test plan.

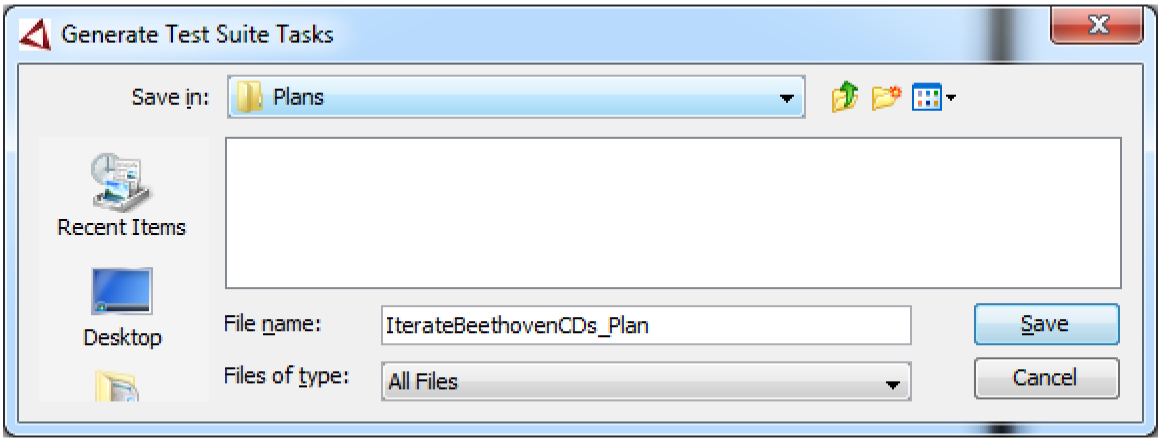
Open the Generate Build Tasks window, in the Execution Suites window.

Click on the Execution Suites node and select the Build Task icon in the toolbar or right-click and select Generate Build Tasks from the menu.

The Generate Build Tasks window will be displayed.

1. Click on the Execution Suite Node to display the available Execution Suites if needed.
2. Click on a single Execution Suite to display the available Filters
3. If no filter is required, click on the Execution Suite Name to Select
4. If the Execution Suite is to be run with the Filter, Click on the Filter name to be applied
5. Enter the Run Name to be used (Mandatory)
6. Enter the Run Description to be used (Mandatory)
7. Enter Maximum Rows to Run or Enter 0 for All Rows
8. Enter any Row numbers where specific Rows are Required in Specify Rows to Run
9. Check on Stop on Failure if the Execution of this Suite is required to stop if any errors are encountered or tests Fail.
10. Click Add to save settings. The Test Suite is added to the Build Tasks.

11. Repeat Steps 1-10 to add additional Execution Suites to the Test Plan
12. When all required Execution Tests are added to the Test Plan enter **Task Options**
    - The TAF Pro password for the logged in User
    - Stop on Failure if the whole test should **Stop** if any errors are encountered during execution.
13. Click Generate to create the Test Plan.
14. The Generate Test Suite Tasks screen is displayed and defaults to the top level directory of the datastore.

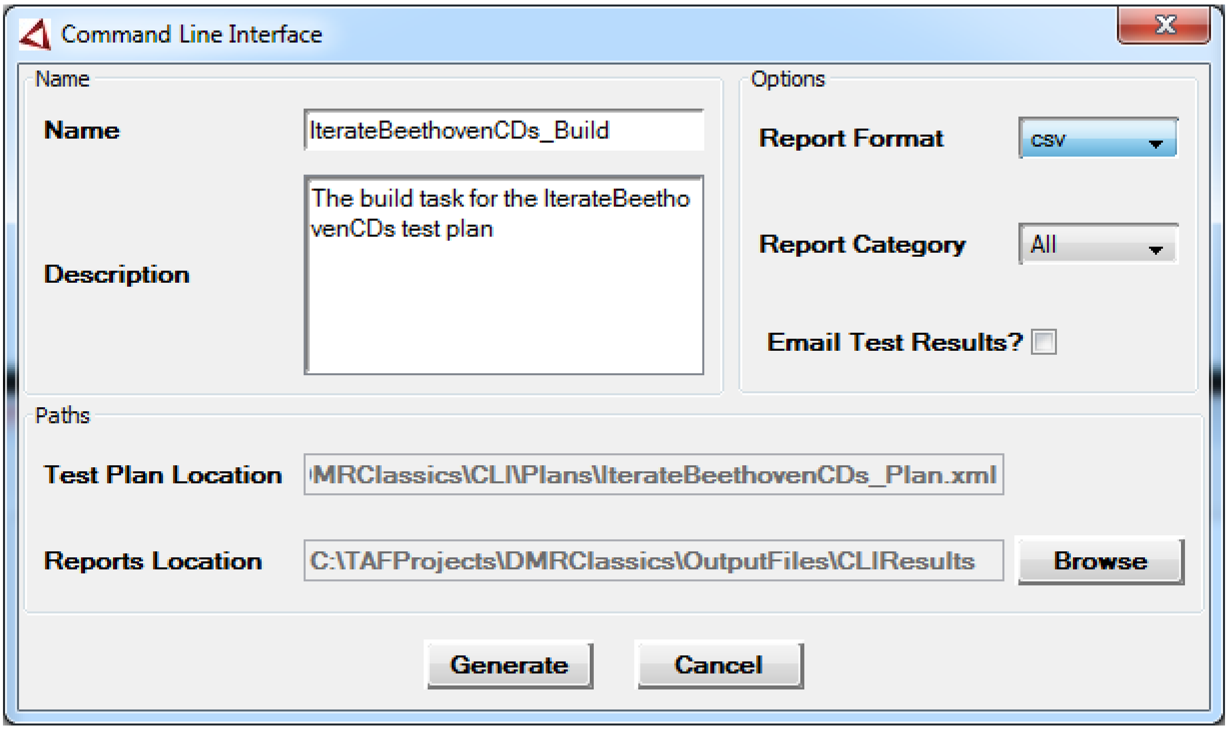


Save in: Plans

File name: IterateBeethovenCDs_Plan

Files of type: All Files

15. If you have not already done so create a folder for storing CLI Plans. Navigate to the sub folder. Save the plan. The Command Line Interface window is displayed



Name: IterateBeethovenCDs_Build

Description: The build task for the IterateBeethovenCDs test plan

Report Format: csv

Report Category: All

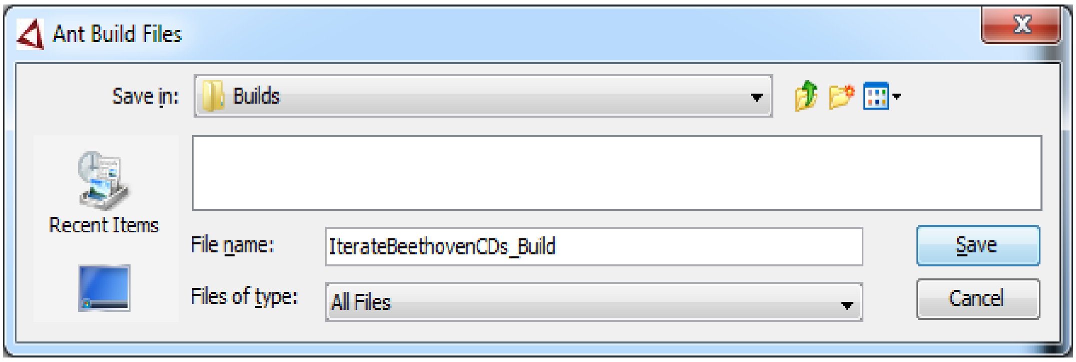
Email Test Results? □

Test Plan Location: MRClassics\CLI\Plans\IterateBeethovenCDs_Plan.xml

Reports Location: C:\TAFProjects\DMRClassics\OutputFiles\CLIResults

16. On the Command Line Interface window:
    - Enter the name and description for the build

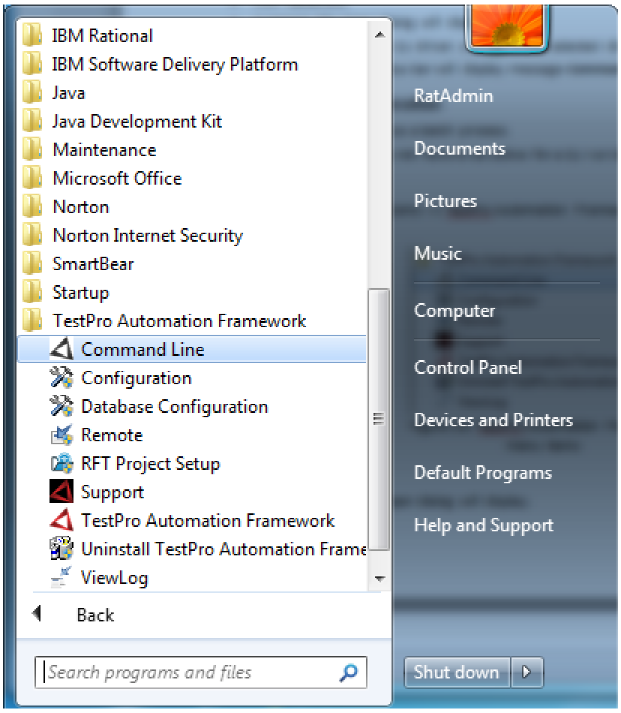
- The only report format currently available is csv
- Select a report category.
- Select Email Results if required. You will be given the opportunity to alter the email information entered in your preferences.
- Select an alternative Report location if required
- Click the Generate button to generate and save the build XML file to the file system where the CLI folder is created.



File name: IterateBeethovenCDs_Build

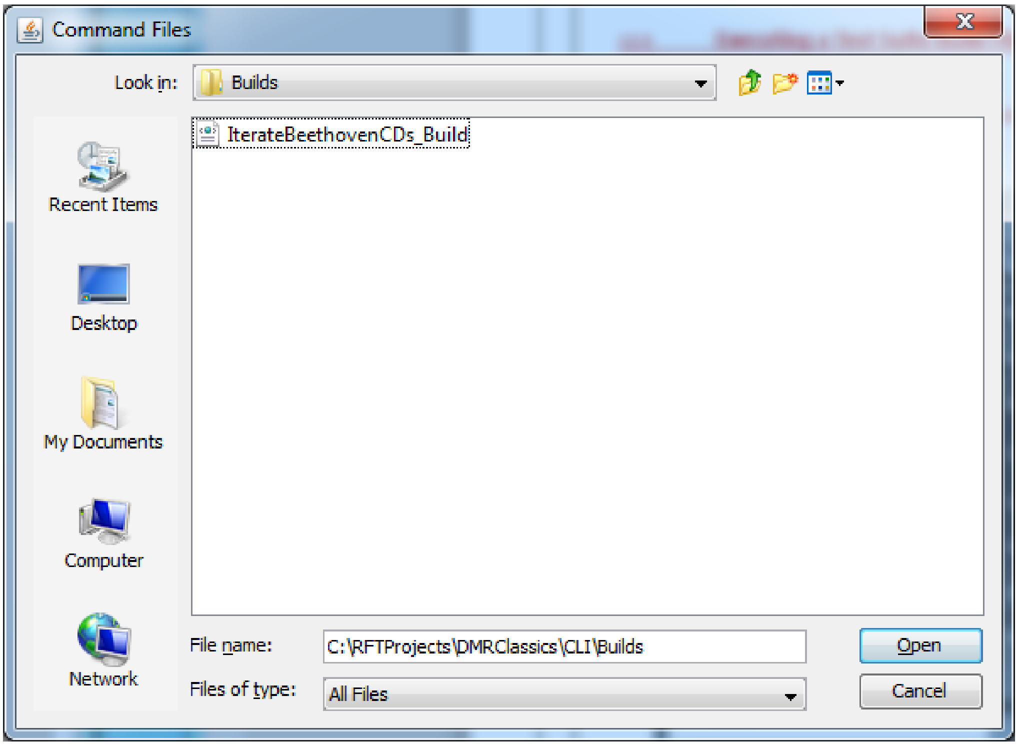
Files of type: All Files

10.4 Executing a Test Suite Build File using the TAF Pro Command Line Utility



TAF Pro provides a Command Line Utility for running the generated execution files. It is available from the Windows Start menu. Running a test this way requires access to the screen on the PC used. Only a single build file can be used at any time.

1. When the Command Files window opens navigate to the folder containing your build.



Look in: Builds

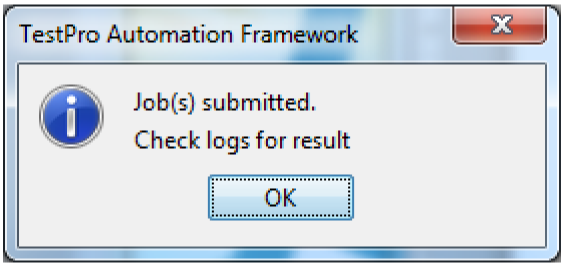
IterateBeethovenCDs_Build

File name: C:\RFTProjects\DMRClassics\CLI\Builds

Files of type: All Files

2. Navigate to the folder containing your Build Files and Select the build required. Click Open.

3. The following screen will be displayed and the command line run will commence after a few seconds. Click on the "Ok" button. A test may take some time to complete. Refresh the Results panel to check the progress of the run.



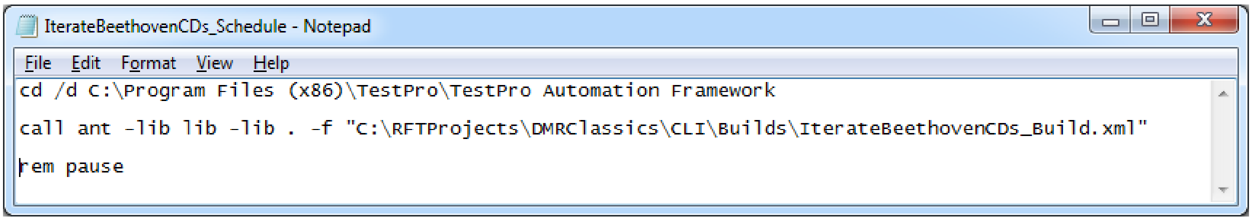
Job(s) submitted.
Check logs for result

4. When the job is completed the results will appear in the Execution Results window.

10.5 Executing Tests using a Batch File

The primary purpose for starting a batch file to run the build is to confirm that the files can be run using a scheduler.

A sample batch file is displayed below. Modify the file to point to the required build file, save it in your preferred folder, and the batch file is ready to go.


IterateBeethovenCDs_Schedule - Notepad
File Edit Format View Help

```
cd /d C:\Program Files (x86)\TestPro\TestPro Automation Framework
call ant -lib lib -lib . -f "C:\RFTProjects\DMRClassics\CLI\Builds\IterateBeethovenCDs_Build.xml"
rem pause
```

To execute the batch file:

1. Navigate to the folder containing the batch file;
2. Run the batch file either by double clicking or using the Run as Administrator menu option.

10.1 Executing Tests using a Scheduler

Scheduling tests allows for regular out-of-hours regression testing. Typically when using a scheduler access to an email service is required to allow the process to forward the results without the need to log into the testing machine.

There are various scheduling programs available and this is not a manual on how to use them.

Create a task in the scheduling tool utilising the batch file created in the previous section, which calls the Ant program and set the time for the run.

10.2 Execution Output Files

10.2.1 TAF Pro and Ant Logs

Log file from running Execution Suites will be located at: C:\Users\Public\TestPro\TestPro Automation Framework\Logs (This is the default path set during installation for TAF Pro, it is not the location of the scripting tool logs for the users' environment.

The files produced are:

- TAF Pro.log – output by TAF Pro, contains detail before and after test execution;
- TAF ProTool.log – output by TAF Pro, contains detail during test execution;
- CLIyyyyMMddhhmmss.log – output by Ant, contains BUILD SUCCESSFUL or BUILD FAILED result.

10.2.2 Scripting Tool Logs

Located at: C:\{*TAF ProjectsPath*}\{*TAF ProjectName*}\LogFiles

- LOGyyyyMMddhhmmss – output from Rational Functional Tester, contains its logging data.

10.2.3 Results File

The results file is located in the Reports directory set in user Preferences.

- <suite>_DyyyyMMddhhmmss.csv – contains the result details that can be seen in the results window after execution.

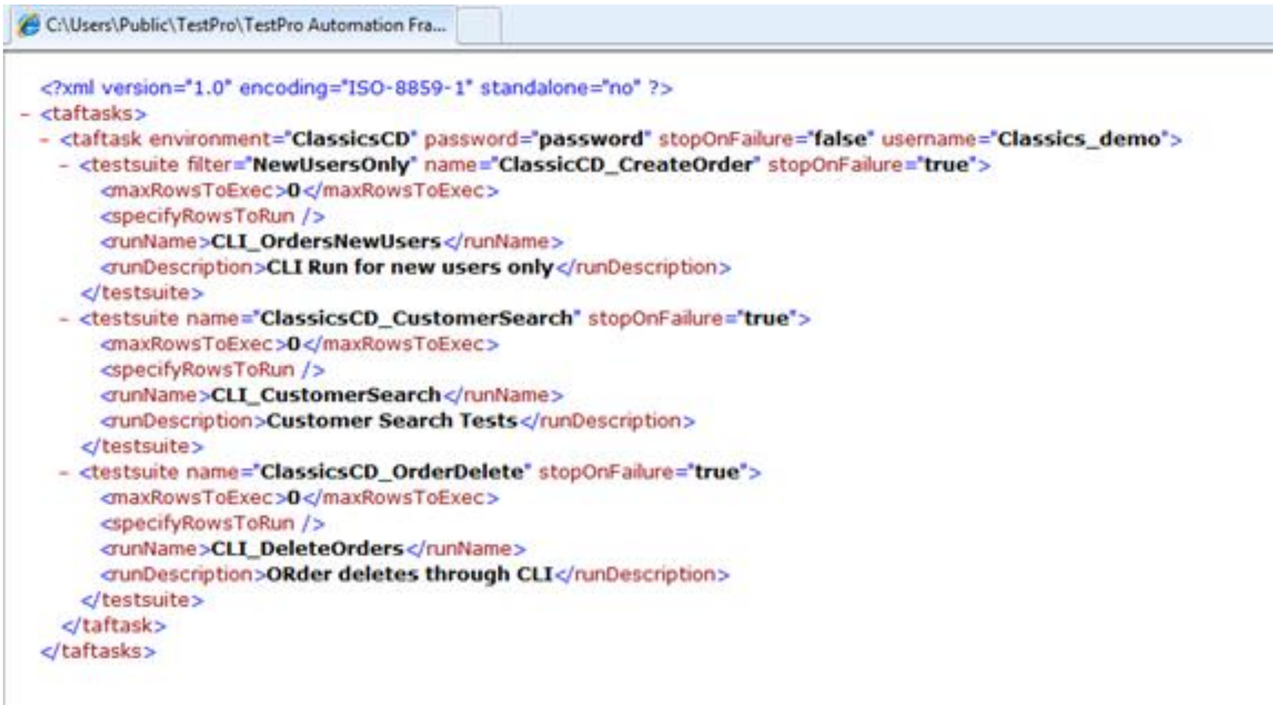
This file may also be emailed to recipients during test execution.

10.3 Understanding the Generated Files

10.3.1 Generated Test Plan XML File

The TAF Pro Test Plan file contains the information that enables the build file to communicate with TAF. The Plan file identifies the Test Suites, Filters and run parameters to be used during execution.

The extract shown below is an example of the Test Plan XML file created by TAF Pro:



```
  <?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
- <taftasks>
  - <taftask environment="ClassicsCD" password="password" stopOnFailure="false" username="Classics_demo">
    - <testsuite filter="NewUsersOnly" name="ClassicCD_CreateOrder" stopOnFailure="true">
        <maxRowsToExec>0</maxRowsToExec>
        <specifyRowsToRun />
        <runName>CLI_OrdersNewUsers</runName>
        <runDescription>CLI Run for new users only</runDescription>
      </testsuite>
    - <testsuite name="ClassicsCD_CustomerSearch" stopOnFailure="true">
        <maxRowsToExec>0</maxRowsToExec>
        <specifyRowsToRun />
        <runName>CLI_CustomerSearch</runName>
        <runDescription>Customer Search Tests</runDescription>
      </testsuite>
    - <testsuite name="ClassicsCD_OrderDelete" stopOnFailure="true">
        <maxRowsToExec>0</maxRowsToExec>
        <specifyRowsToRun />
        <runName>CLI_DeleteOrders</runName>
        <runDescription>ORder deletes through CLI</runDescription>
      </testsuite>
    </taftask>
  </taftasks>
```

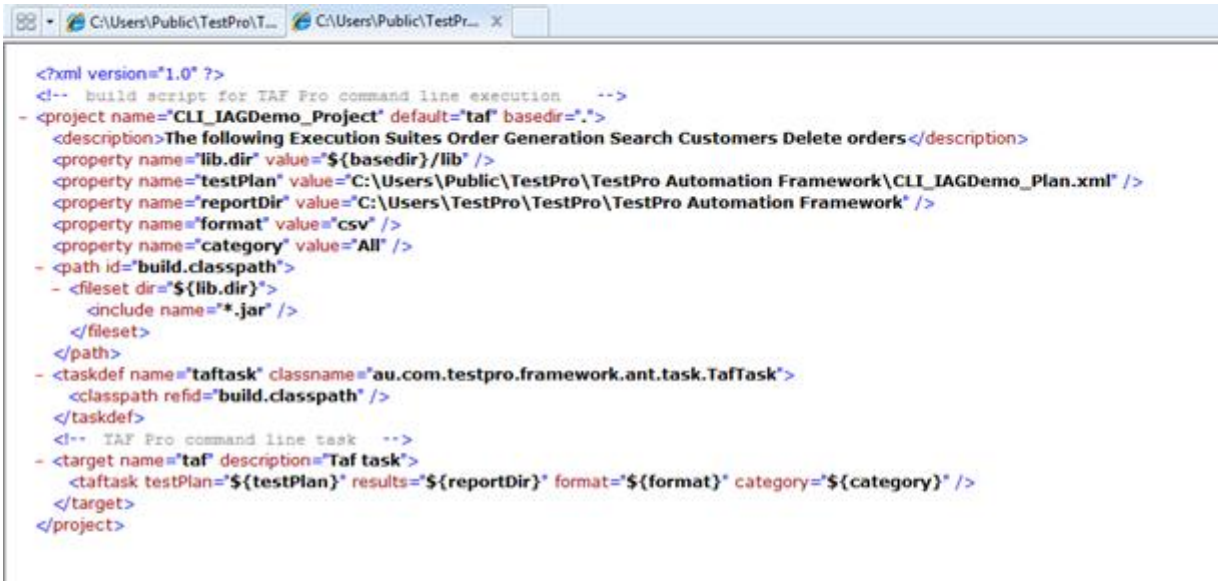
The table below contains an explanation of the instruction set and suggested updates as needed**.**

Tag	Description
<taftasks>	Initiating statement of the Task Set
<taftask>	Sets the TAF Pro Environment, username and password to be used for test execution. stopOnFailure – set to true will not execute the test if the previous test suite execution returned true to Environment level stopOnFailure
<testsuite>	Sets the Execution suite to be executed, any filter to be applied and internal control to stop if any errors are encountered.
<maxRowsToExec>	Determines the number of rows to be executed in the test run. Defaulted from User > Preferences > General
<specifyRowsToRun>	Determines specific rows to run. Defaults to the rows to run for all.
<runName>	Defaulted from User > Preferences > Command Line > Name and is the name under which results will be stored

Tag	Description
<runDescription>	Description given to the run name and defaulted Line > Name.
</testsuite>	Denotes the close for the parameters for the TestSuite Section
</taftask>	Denotes the close for the parameters for the taftask Section
</taftasks>	Denotes the close for the Test Set

10.3.2 Generated Ant Build Driver

An illustration of the generated Ant build file is shown below.



```
  <?xml version="1.0" ?>
  <!--  build script for TAF Pro command line execution    -->
- <project name="CLI_IAGDemo_Project" default="taf" basedir=".">
    <description>The following Execution Suites Order Generation Search Customers Delete orders</description>
    <property name="lib.dir" value="${basedir}/lib" />
    <property name="testPlan" value="C:\Users\Public\TestPro\TestPro Automation Framework\CLI_IAGDemo_Plan.xml" />
    <property name="reportDir" value="C:\Users\TestPro\TestPro\TestPro Automation Framework" />
    <property name="format" value="csv" />
    <property name="category" value="All" />
  - <path id="build.classpath">
    - <fileset dir="${lib.dir}">
        <include name="*.jar" />
      </fileset>
    </path>
  - <taskdef name="taftask" classname="au.com.testpro.framework.ant.task.TafTask">
      <classpath refid="build.classpath" />
    </taskdef>
    <!--  TAF Pro command line task   -->
  - <target name="taf" description="Taf task">
      <taftask testPlan="${testPlan}" results="${reportDir}" format="${format}" category="${category}" />
    </target>
  </project>
```

Symbolic	Description
testPlan	The file name saved after the **Build Tasks Generate** buttons was clicked.
reportDir	The value from **Preferences → Command Line → Paths**.
format	the value from **Preferences → Command Line → Options**
category	the value from **Preferences → Command Line → Options**

Symbolic	Description
email	Set to **true** if the email checkbox on the **Command Line Interface** dialog was checked, otherwise **false**.

11 Examining Test Results

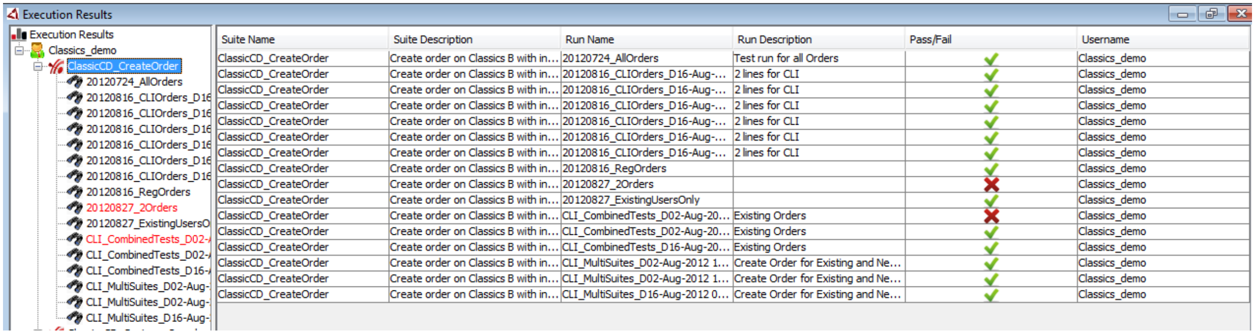
11.1 Overview

Results from each test run are displayed within the Execution Results window. The user is able to view test results at different levels within the Results window to provide a summary view of the tests. The different view levels from lowest to highest level include:

- Datagroup Level – A single data set within a test run;
- Action Level – A Single action data set within the test run. May have several data group details;
- Test Case Level – All Actions results within a single iteration of the Test Case which may include several actions in the single test case;
- Scenario Level – All Test Case results within a single iteration of the Scenario, which may include several test cases in a single scenario;
- Row Level – All scenarios within a single test iteration. This level may include several scenarios;
- Run Level – All test iterations for the test run. This level will include all the tests completed within a test execution. If the same test run name is used for more than one Test Execution, this level will include the test results for all test executions run with the same name. Only if all rows run in the test pass will this level show as a pass;
- Suite Level –Summarizes all test results into a single test run line. This level reports at summary level for all Test runs completed for the Test Suite. Only if all rows run in the test pass will this level show as a pass;
- User Level - Summarizes all test results into a single test run line. This level reports at summary level for all Test runs completed for all test runs for all execute suites owned by the User.

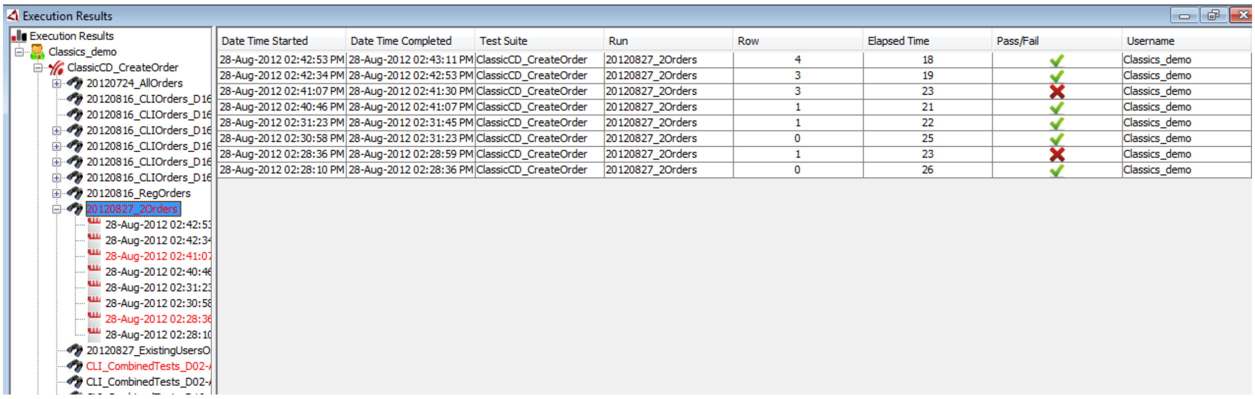
11.2 View Test Results

1. Click in the Results Window
2. Expand the Results Tree to the Level of view required.
3. Click on the required level
4. Results are displayed in the results panel.



Suite Name	Suite Description	Run Name	Run Description	Pass/Fail	Username
ClassicCD_CreateOrder	Create order on Classics B with in...	20120724_AllOrders	Test run for all Orders	✓	Classics_demo
ClassicCD_CreateOrder	Create order on Classics B with in...	20120816_CLIOrders_D16-Aug-...	2 lines for CLI	✓	Classics_demo
ClassicCD_CreateOrder	Create order on Classics B with in...	20120816_CLIOrders_D16-Aug-...	2 lines for CLI	✓	Classics_demo
ClassicCD_CreateOrder	Create order on Classics B with in...	20120816_CLIOrders_D16-Aug-...	2 lines for CLI	✓	Classics_demo
ClassicCD_CreateOrder	Create order on Classics B with in...	20120816_CLIOrders_D16-Aug-...	2 lines for CLI	✓	Classics_demo
ClassicCD_CreateOrder	Create order on Classics B with in...	20120816_CLIOrders_D16-Aug-...	2 lines for CLI	✓	Classics_demo
ClassicCD_CreateOrder	Create order on Classics B with in...	20120816_CLIOrders_D16-Aug-...	2 lines for CLI	✓	Classics_demo
ClassicCD_CreateOrder	Create order on Classics B with in...	20120816_RegOrders		✓	Classics_demo
ClassicCD_CreateOrder	Create order on Classics B with in...	20120827_2Orders		✗	Classics_demo
ClassicCD_CreateOrder	Create order on Classics B with in...	20120827_ExistingUsersOnly		✓	Classics_demo
ClassicCD_CreateOrder	Create order on Classics B with in...	CLI_CombinedTests_D02-Aug-20...	Existing Orders	✗	Classics_demo
ClassicCD_CreateOrder	Create order on Classics B with in...	CLI_CombinedTests_D02-Aug-20...	Existing Orders	✓	Classics_demo
ClassicCD_CreateOrder	Create order on Classics B with in...	CLI_CombinedTests_D16-Aug-20...	Existing Orders	✓	Classics_demo
ClassicCD_CreateOrder	Create order on Classics B with in...	CLI_MultiSuites_D02-Aug-2012 1...	Create Order for Existing and Ne...	✓	Classics_demo
ClassicCD_CreateOrder	Create order on Classics B with in...	CLI_MultiSuites_D02-Aug-2012 1...	Create Order for Existing and Ne...	✓	Classics_demo
ClassicCD_CreateOrder	Create order on Classics B with in...	CLI_MultiSuites_D16-Aug-2012 0...	Create Order for Existing and Ne...	✓	Classics_demo

All results are linked to the User and Execution Suite. This is then further defined by the Run Name provided at execution time. Within the Run Name all iterations of the test are displayed by date and time in descending order as displayed below.



Date Time Started	Date Time Completed	Test Suite	Run	Row	Elapsed Time	Pass/Fail	Username
28-Aug-2012 02:42:53 PM	28-Aug-2012 02:43:11 PM	ClassicCD_CreateOrder	20120827_2Orders	4	18	✓	Classics_demo
28-Aug-2012 02:42:34 PM	28-Aug-2012 02:42:53 PM	ClassicCD_CreateOrder	20120827_2Orders	3	19	✓	Classics_demo
28-Aug-2012 02:41:07 PM	28-Aug-2012 02:41:30 PM	ClassicCD_CreateOrder	20120827_2Orders	3	23	✗	Classics_demo
28-Aug-2012 02:40:46 PM	28-Aug-2012 02:41:07 PM	ClassicCD_CreateOrder	20120827_2Orders	1	21	✓	Classics_demo
28-Aug-2012 02:31:23 PM	28-Aug-2012 02:31:45 PM	ClassicCD_CreateOrder	20120827_2Orders	1	22	✓	Classics_demo
28-Aug-2012 02:30:58 PM	28-Aug-2012 02:31:23 PM	ClassicCD_CreateOrder	20120827_2Orders	0	25	✓	Classics_demo
28-Aug-2012 02:28:36 PM	28-Aug-2012 02:28:59 PM	ClassicCD_CreateOrder	20120827_2Orders	1	23	✗	Classics_demo
28-Aug-2012 02:28:10 PM	28-Aug-2012 02:28:36 PM	ClassicCD_CreateOrder	20120827_2Orders	0	26	✓	Classics_demo

Each test iteration result can be further refined to display the results for each scenario within the test and further to a single action and dataset used in a single test line.

11.3 Set Execution Results View Criteria

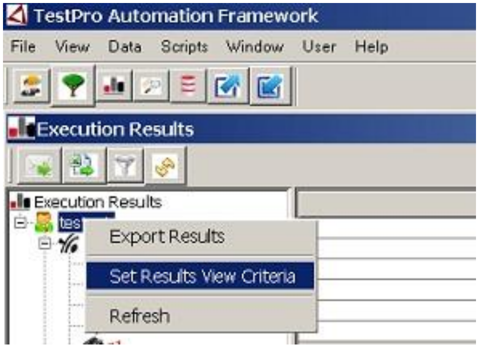
This functionality helps to filter Execution Results for a particular time period by excluding other results from view. These results can be filtered at a ***User level*** and ***Execution Suite level*** in the **Execution Results window**

11.3.1 Environment Level search criteria

When the search criteria are set at Environment level it displays all the execution results which were run within this time frame by this user across all environments.

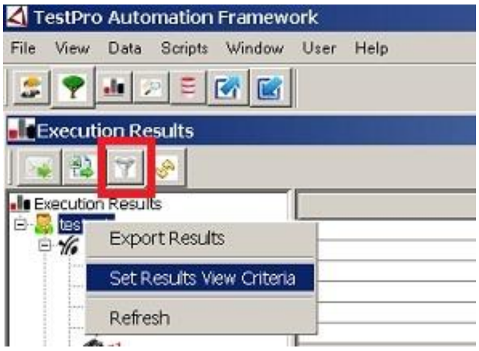
How to set Search Results Criteria at Environment Level:

1. Right Click on User node > Set Results View Criteria

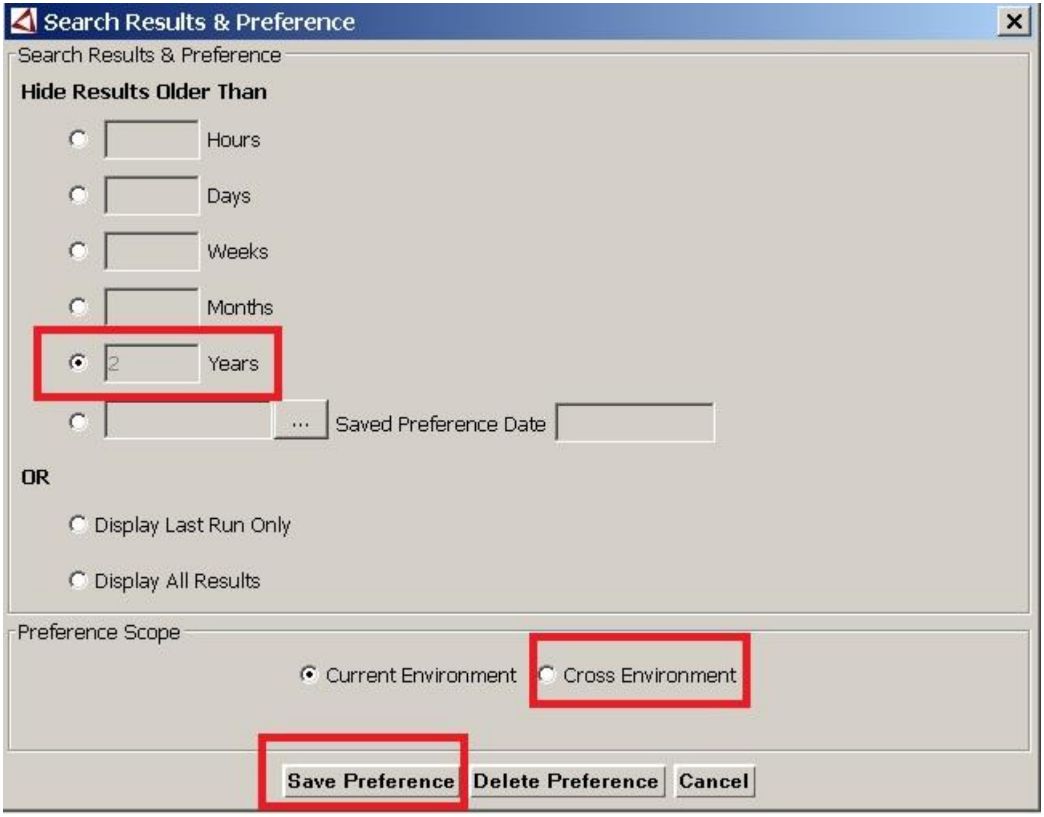

TestPro Automation Framework
File View Data Scripts Window User Help
Execution Results
Execution Results
tes
Export Results
Set Results View Criteria
Refresh

OR

Select on User node > click on Filter Results button in the Tool Bar


TestPro Automation Framework
File View Data Scripts Window User Help
Execution Results
Execution Results
tes
Export Results
Set Results View Criteria
Refresh

2. Fill in criteria by selecting the appropriate time period radio button in ***Search Results & Preference*** pane as shown below, click ***Cross Environment*** radio button in ***Preference Scope*** pane.
3. Click the ***Save Preference*** button.
4. To waive a preference setting, click ***Delete Preference*** button.



Search Results & Preference

Search Results & Preference

Hide Results Older Than

○ Hours

○ Days

○ Weeks

○ Months

● 2 Years

○ ... Saved Preference Date

OR

○ Display Last Run Only

○ Display All Results

Preference Scope

● Current Environment ○ Cross Environment

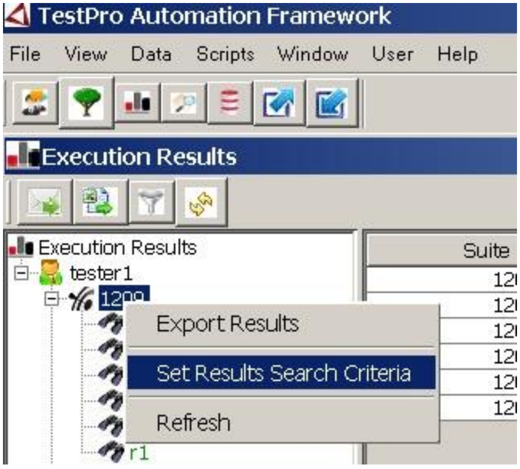
11.3.2 User Level search criteria

When the search criteria are set at User level it displays all the execution results which were run within this time frame by this user in the current environment. This setting overrides the Environment Level setting.

To set User Level search criteria, follow the same steps illustrated in above section (Environment search criteria), except that click on ***current Environment*** radio button in the ***Preference Scope*** pane.

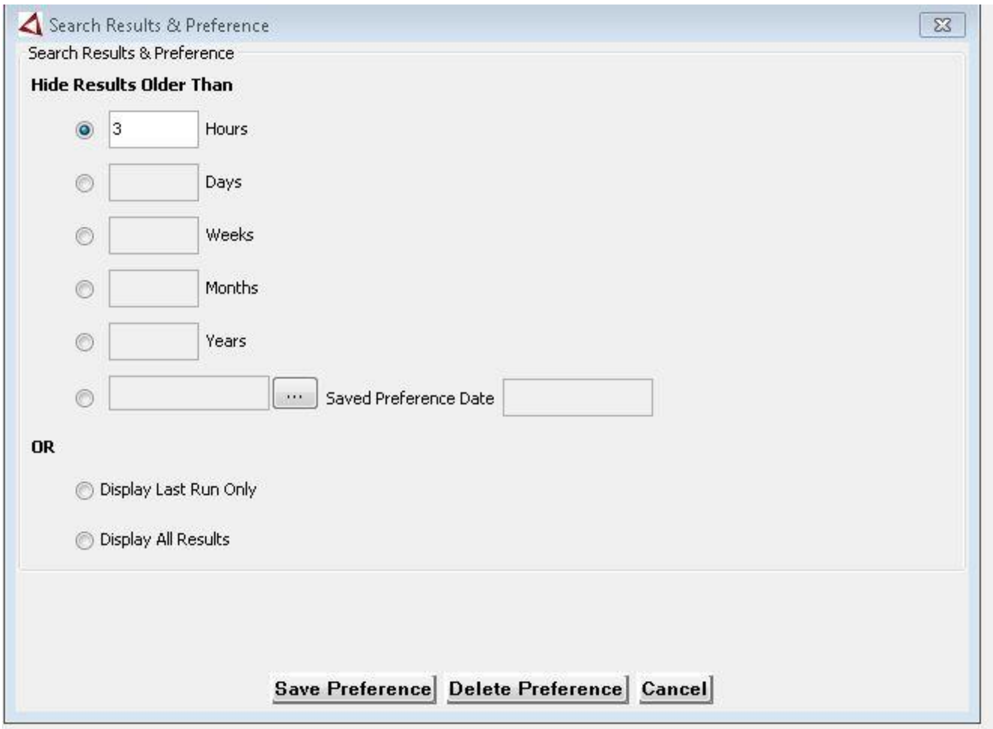
11.3.3 Execution Suite level search criteria

When the search criteria are set at Execution Suite level it only filters the results in that particular suite. This setting overrides the User Level setting.


TestPro Automation Framework
File View Data Scripts Window User Help
Execution Results
Execution Results
tester1
1209
r1
Suite
12
Export Results
Set Results Search Criteria
Refresh

How to set Search Results Criteria at Execution Suite level:

1. Right click on the ***Execution level*** and select **Set Results Search Criteria**.
   OR
   Click on the execution node, then click on ***Filter Results*** button in the Tool Bar.
2. The **Search Results & Preferences** window will open


Search Results & Preference

Search Results & Preference

Hide Results Older Than

● 3 Hours

○ Days

○ Weeks

○ Months

○ Years

○ ... Saved Preference Date

OR

○ Display Last Run Only

○ Display All Results

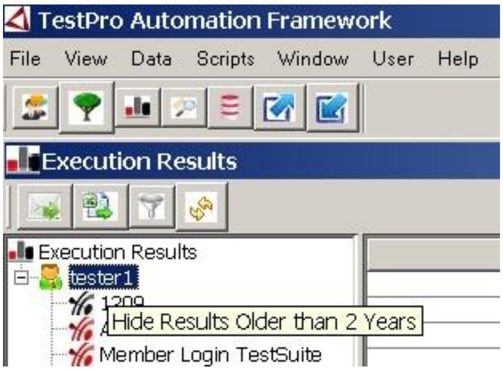
3. The user can select from any of the criteria listed, either hours/days/weeks/months or years, by selecting the radio button which will then enable the associated field.
4. Click on ***Save Preference*** button to save changes and close the window.

Prior to TAF Pro version 4.2.0, execution results were grouped by run name collectively as a single record regardless of different execution periods. Therefore the search criteria will not be able to filter all data within the test run. If any record within a test run meets the search criteria, all records within that run will be displayed.

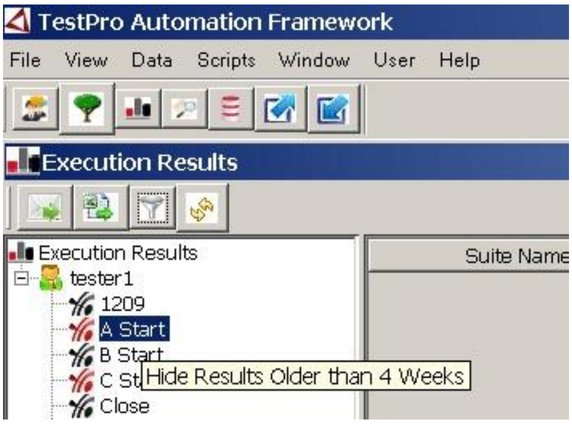
After TAF Pro 4.2.0, each test execution is saved as a separate record even if it has the same test run name as another test execution. Therefore, search criteria applied to results executed in the new version will be able to filter results to provide only those results that meet the relevant criteria.

11.4 View Search Criteria Settings

Hover the mouse over a Suite or User name to display the search criteria currently applied for filtering results.


TestPro Automation Framework
File View Data Scripts Window User Help
Execution Results
Execution Results
tester1
1209
Hide Results Older than 2 Years
Member Login TestSuite

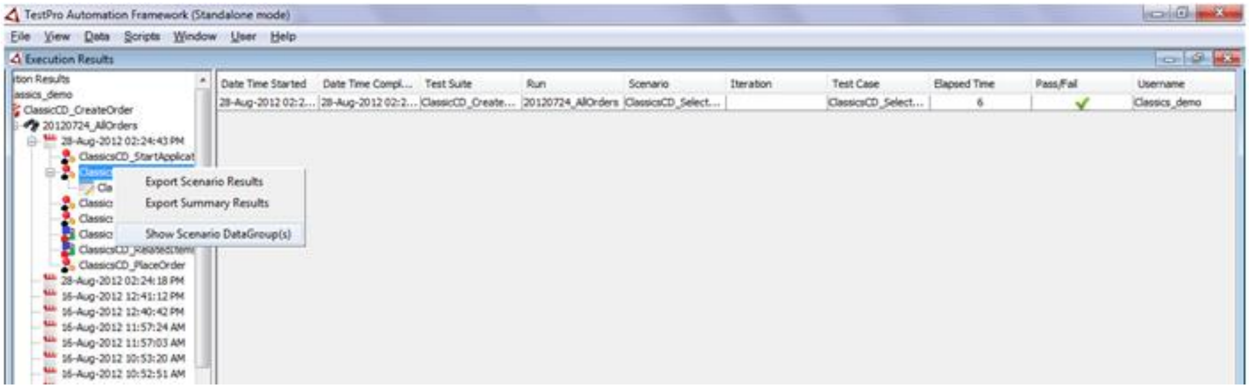
View Search Criteria setting for the current User


TestPro Automation Framework
File View Data Scripts Window User Help
Execution Results
Execution Results
tester1
1209
A Start
B Start
C St
Close
Hide Results Older than 4 Weeks
Suite Name

View Search Criteria setting for an Execution Suite

11.5 Show Data Groups

When viewing test results the user can either continue to drill down through the layers of reporting to see the lowest level of detail for an action at the data group detail or at any of the scenario, test case or action level the user can right click and immediately view the DataGroups applicable to that level. An example is shown below.



Date Time Started	Date Time Compl...	Test Suite	Run	Scenario	Iteration	Test Case	Elapsed Time	Pass/Fail	Username
28-Aug-2012 02:2...	28-Aug-2012 02:2...	ClassicCD_Create...	20120724_AllOrders	ClassicsCD_Select...		ClassicsCD_Select...	6	✓	Classics_demo

When the user clicks on the scenario level on the execution tree, they are presented with summary data for the scenario. By selecting Show Data Groups at the scenario level the use is able to view the detailed data groups applicable to the scenario. This is also available at Test Case and Action level views.

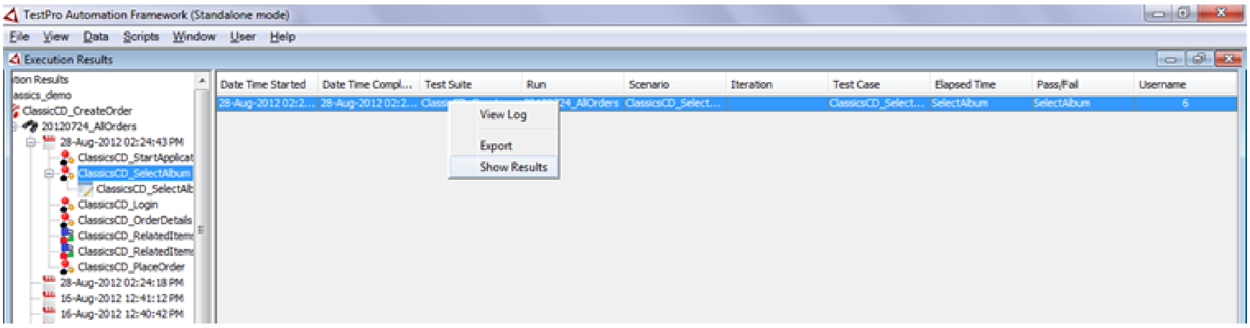
11.6 Show Test Results

The data of each individual test can be displayed to help with identifying issues that may explain test failure or confirm the inputs and outputs to the test to ensure it is functioning correctly.

When Results are displayed the user is able to drill further into a specific test to look at the actual inputs and outputs from the single action in the test. Likewise the user is able to view the detailed data and information on any failed tests to determine where the issues lie.

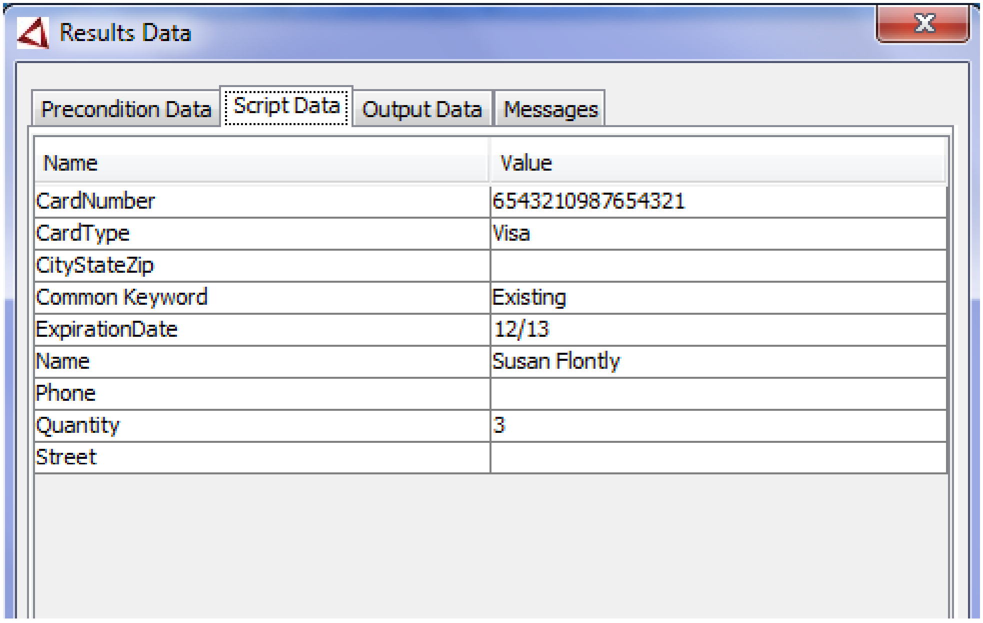
To view test data:

1. ***Click*** on the Results Tree ***to expand*** the test results to the datagroup level lowest data level is displayed in the results window;
2. ***Select*** the ***data line;***
3. ***Right click*** and select ***Show Results*** as shown below.



Date Time Started	Date Time Compl...	Test Suite	Run	Scenario	Iteration	Test Case	Elapsed Time	Pass/Fail	Username
28-Aug-2012 02:2...	28-Aug-2012 02:2...	Classi	24_AllOrders	ClassicsCD_Select...		ClassicsCD_Select...	SelectAlbum	SelectAlbum	6

4. The Results Data window is displayed:



Name	Value
CardNumber	6543210987654321
CardType	Visa
CityStateZip	
Common Keyword	Existing
ExpirationDate	12/13
Name	Susan Flontly
Phone	
Quantity	3
Street	

5. The Results Data window consists of 4 tabs:

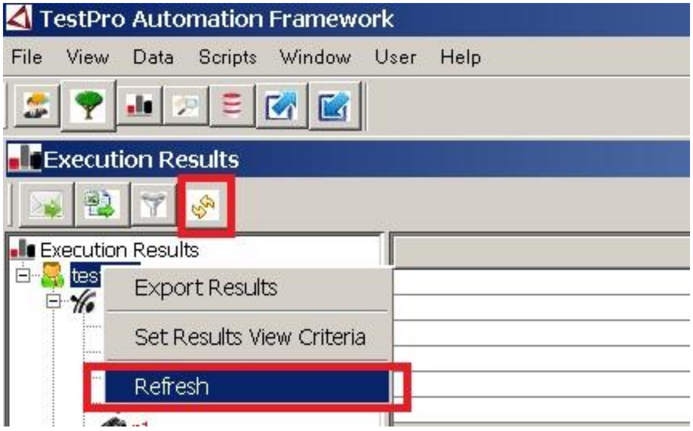
   a. **Precondition Data** – Data values that were passed to the action and used during the processing;
   b. **Script Data** – The individual row of data values used for the selected iteration;
   c. **Output Data**–Data values generated as a result of the test action that are provided to the test suite to be used as input in the next test in the scenario;
   d. **Messages** – Shows the pass or fail message for the test execution. Failures will details the error message received from IBM Rational Functional Tester whereas Pass messages are generally incorporated into the script when it is written.

11.7 Refreshing the Results Window

At times user needs to refresh the result to see the latest run result. For instance, if a recently completed test result isn’t showing up under an already expanded execution node. In this case, user needs to refresh the result’s view.

In order to check ongoing progress of a test execution:

1. click on the root node of the Execution Results Tree or on an Execution Suite node;
2. ***Right Click*** and select ***Refresh*** (see example below);
3. **OR**
4. Click on the ***Refresh*** button in the Tool Bar;
5. The screen will update with all the completed test suite results.


TestPro Automation Framework
File View Data Scripts Window User Help
Execution Results
Execution Results
tes
Export Results
Set Results View Criteria
Refresh

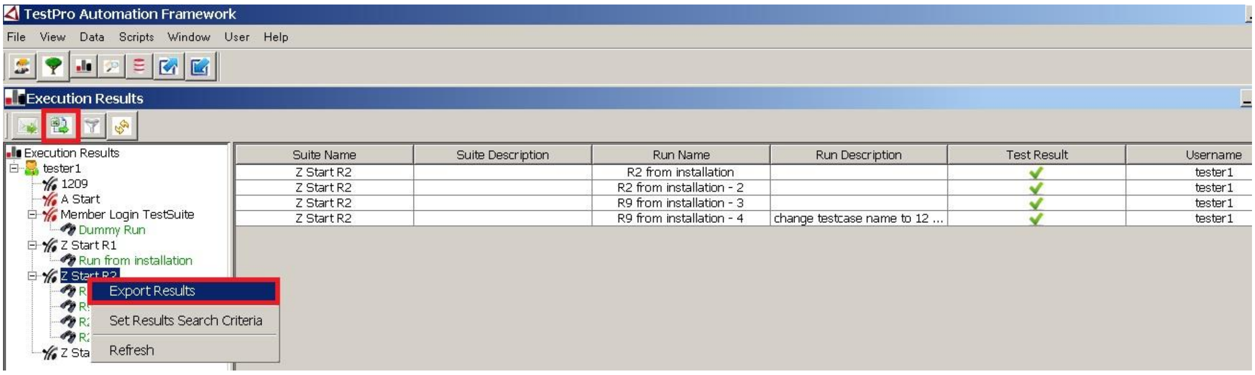
11.8 Exporting Test Results

TAF Pro records all results for tests executed. The view is user-specific and can only be seen by the user who executed the test. In order to communicate test progress and results, TAF Pro provides the ability to export test results in CSV format to make the test data available to alternative reporting tools, or to Excel to format as required.

The user is able to view test results at different levels within the results window to provide a summary view of the tests. The export will export the data to the level selected in the Test Results Tree.

To export test results:

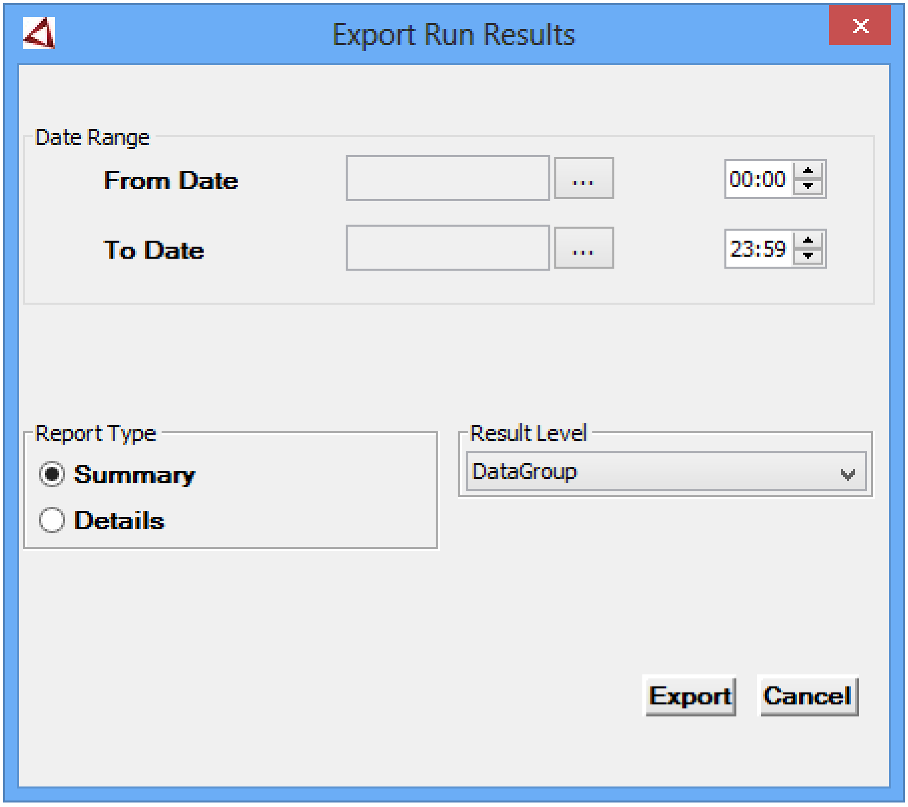
1. Expand the test results tree to the level view required
2. Double click on the relevant level to display the test results as shown below:



Suite Name	Suite Description	Run Name	Run Description	Test Result	Username
Z Start R2		R2 from installation		✓	tester1
Z Start R2		R2 from installation - 2		✓	tester1
Z Start R2		R9 from installation - 3		✓	tester1
Z Start R2		R9 from installation - 4	change testcase name to 12 ...	✓	tester1

3. Right click and select Export Results;

   OR
4. Click on Export Results button in Tool Bar;
5. The Export Data Window is displayed defaulted to your Output Files destination directory which was set at login.



Export Run Results

Date Range

From Date		...	00:00
To Date		...	23:59

Report Type

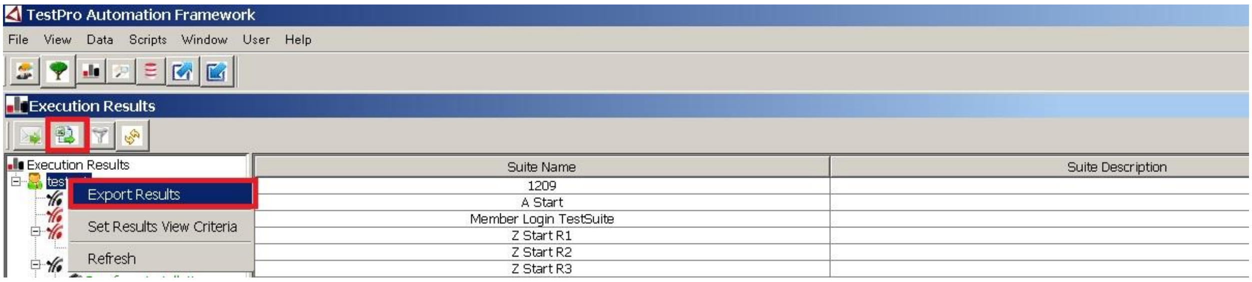
● Summary

○ Details

Result Level: DataGroup

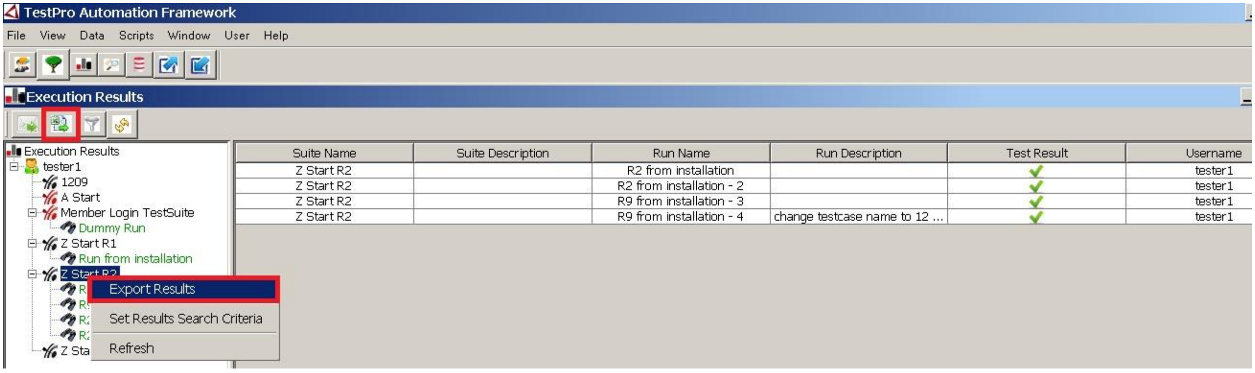
6. Enter **date** and **time** of test results, select **Export** and save the file;
7. Detailed and Summary Results can be exported at any level within the Execution Results Window. The images below show the different levels at which Detailed and Summary results can be exported.

User Level Results will provide an export at detailed level of all results for Tests executed by the user.



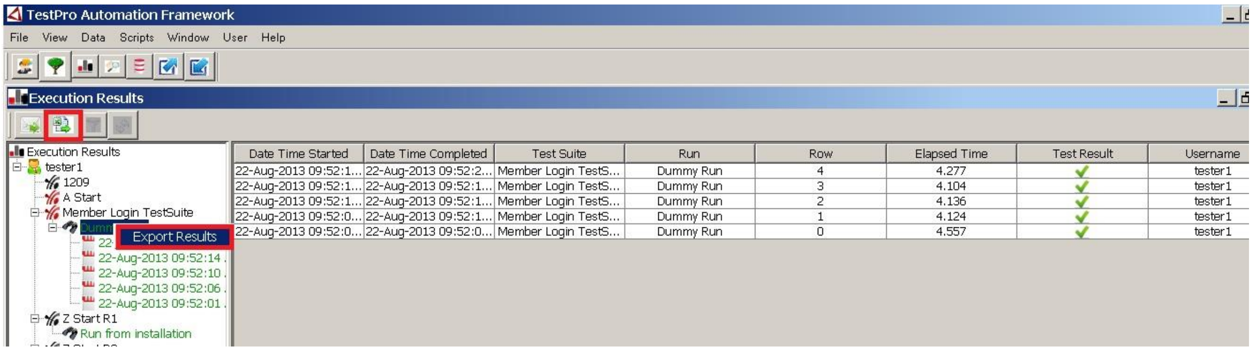
Suite Name	Suite Description
1209	
A Start	
Member Login TestSuite	
Z Start R1	
Z Start R2	
Z Start R3	

Suite Level Results provides an export at either Summary or Detailed level of all results for tests executed on the selected Execution Suite for the User.



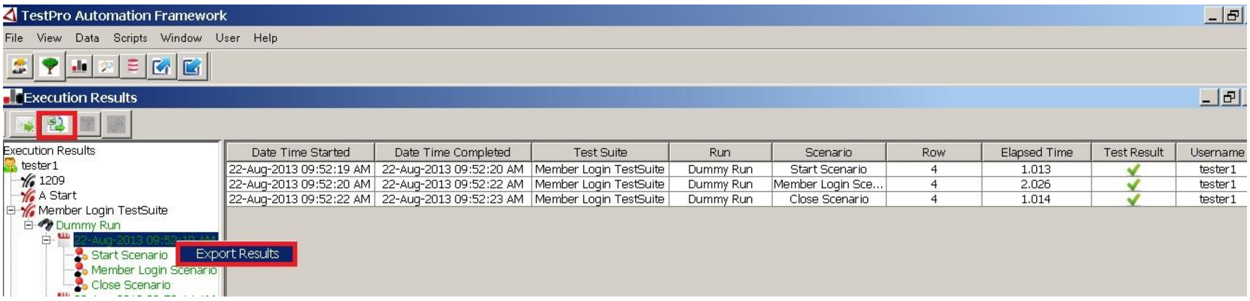
Suite Name	Suite Description	Run Name	Run Description	Test Result	Username
Z Start R2		R2 from installation		✓	tester1
Z Start R2		R2 from installation - 2		✓	tester1
Z Start R2		R9 from installation - 3		✓	tester1
Z Start R2		R9 from installation - 4	change testcase name to 12 ...	✓	tester1

Run Level Results provides an export at either Summary or Detailed level of all results recorded for the selected Test Run for an Execution Suite.



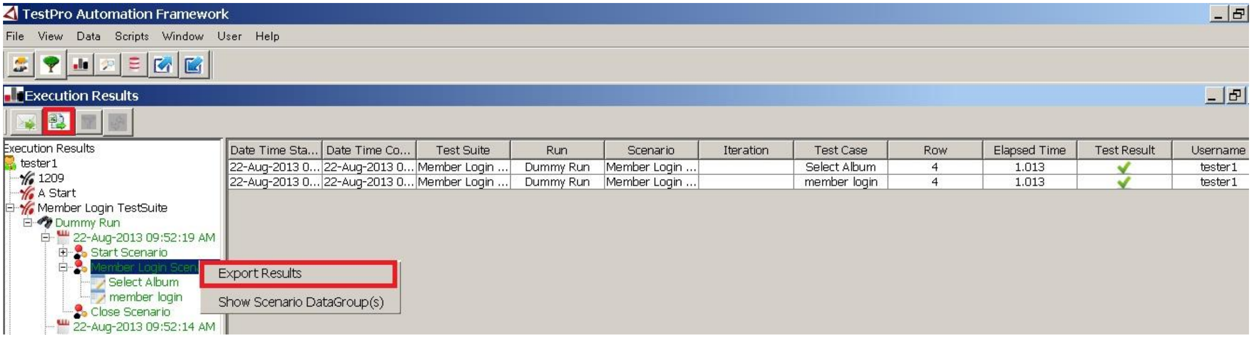
Date Time Started	Date Time Completed	Test Suite	Run	Row	Elapsed Time	Test Result	Username
22-Aug-2013 09:52:1...	22-Aug-2013 09:52:2...	Member Login TestS...	Dummy Run	4	4.277	✓	tester1
22-Aug-2013 09:52:1...	22-Aug-2013 09:52:1...	Member Login TestS...	Dummy Run	3	4.104	✓	tester1
22-Aug-2013 09:52:1...	22-Aug-2013 09:52:1...	Member Login TestS...	Dummy Run	2	4.136	✓	tester1
22-Aug-2013 09:52:0...	22-Aug-2013 09:52:1...	Member Login TestS...	Dummy Run	1	4.124	✓	tester1
22-Aug-2013 09:52:0...	22-Aug-2013 09:52:0...	Member Login TestS...	Dummy Run	0	4.557	✓	tester1

Row Level Results provides and export at either Summary or Detailed level for the nominated row selected within a Run for and Execution Suite.



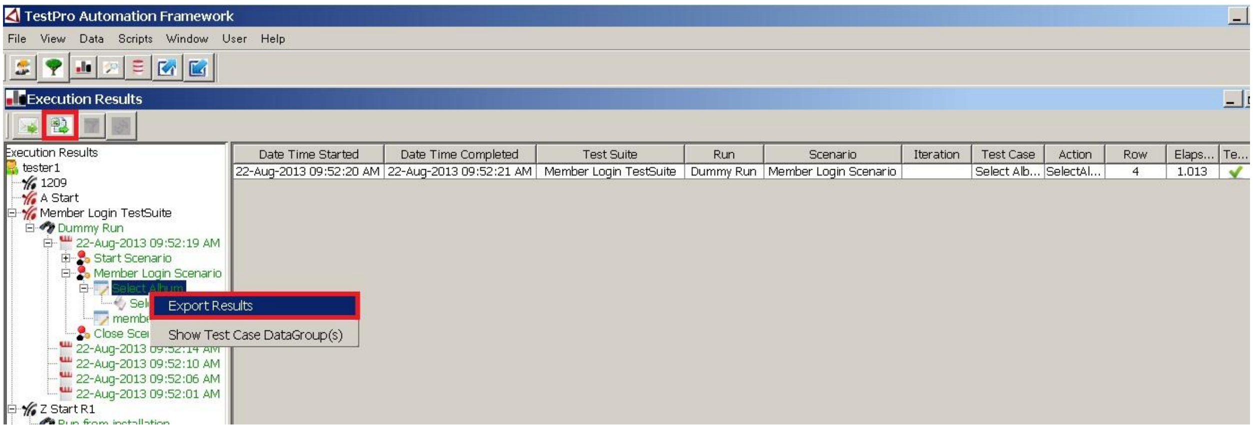
Date Time Started	Date Time Completed	Test Suite	Run	Scenario	Row	Elapsed Time	Test Result	Username
22-Aug-2013 09:52:19 AM	22-Aug-2013 09:52:20 AM	Member Login TestSuite	Dummy Run	Start Scenario	4	1.013	✓	tester1
22-Aug-2013 09:52:20 AM	22-Aug-2013 09:52:22 AM	Member Login TestSuite	Dummy Run	Member Login Sce...	4	2.026	✓	tester1
22-Aug-2013 09:52:22 AM	22-Aug-2013 09:52:23 AM	Member Login TestSuite	Dummy Run	Close Scenario	4	1.014	✓	tester1

Scenario Level Results will export Summary or Detailed results for the selected Scenario within a Test Run.



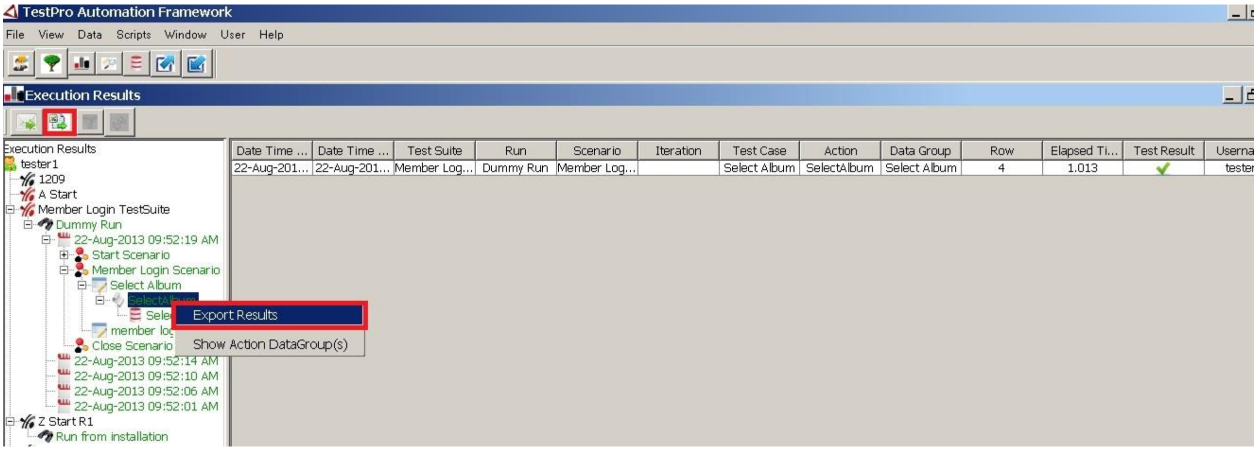
Date Time Sta...	Date Time Co...	Test Suite	Run	Scenario	Iteration	Test Case	Row	Elapsed Time	Test Result	Username
22-Aug-2013 0...	22-Aug-2013 0...	Member Login ...	Dummy Run	Member Login ...		Select Album	4	1.013	✓	tester1
22-Aug-2013 0...	22-Aug-2013 0...	Member Login ...	Dummy Run	Member Login ...		member login	4	1.013	✓	tester1

Test Case Level Results will export Summary or Detailed results for the selected Test case.



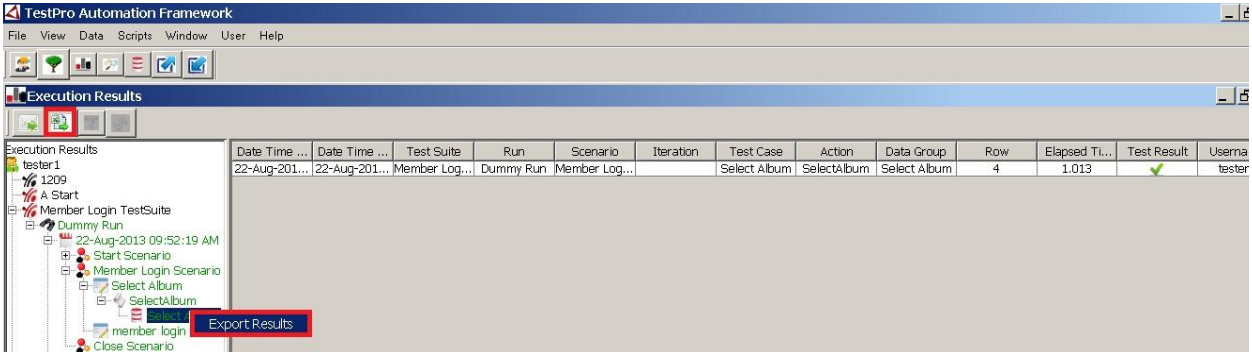
Date Time Started	Date Time Completed	Test Suite	Run	Scenario	Iteration	Test Case	Action	Row	Elaps...	Te...
22-Aug-2013 09:52:20 AM	22-Aug-2013 09:52:21 AM	Member Login TestSuite	Dummy Run	Member Login Scenario		Select Alb...	SelectAl...	4	1.013	✓

Action Level Results will export Summary or Detailed results for the selected Action.



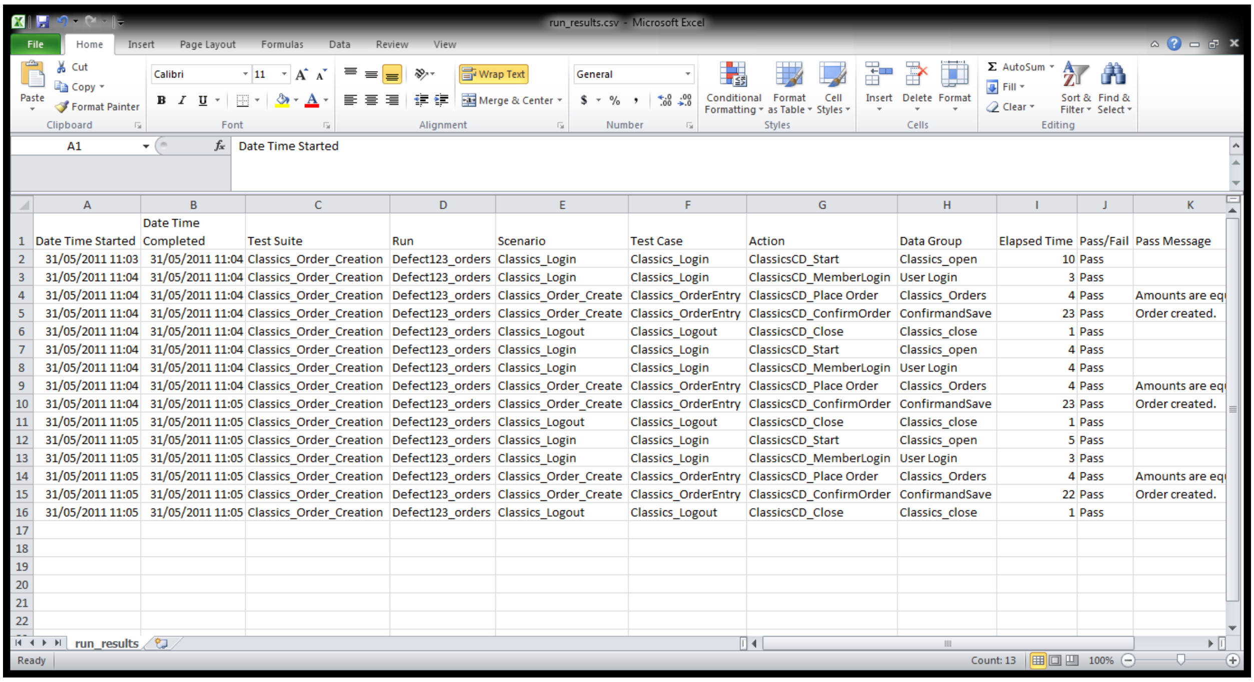
Date Time ...	Date Time ...	Test Suite	Run	Scenario	Iteration	Test Case	Action	Data Group	Row	Elapsed Ti...	Test Result	Userna
22-Aug-201...	22-Aug-201...	Member Log...	Dummy Run	Member Log...		Select Album	SelectAlbum	Select Album	4	1.013	✓	tester

Data Group Level Results will export detailed results for the selected Datagroup.



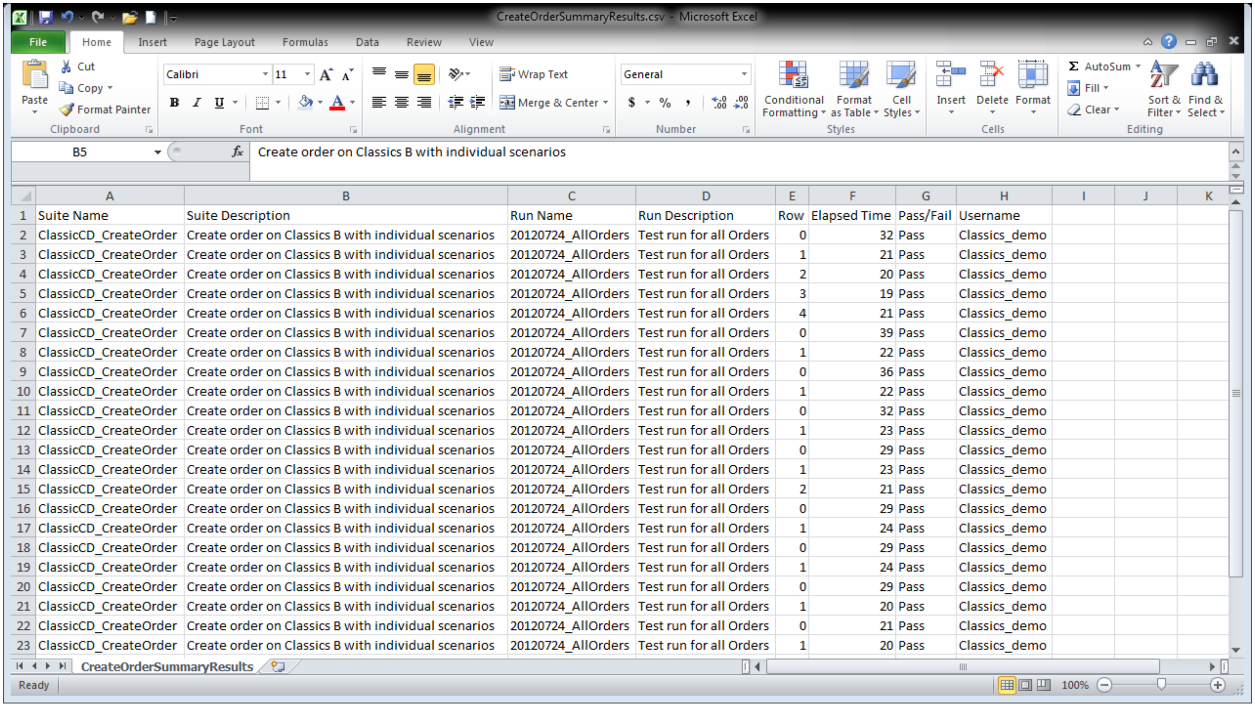
Date Time ...	Date Time ...	Test Suite	Run	Scenario	Iteration	Test Case	Action	Data Group	Row	Elapsed Ti...	Test Result	Userna
22-Aug-201...	22-Aug-201...	Member Log...	Dummy Run	Member Log...		Select Album	SelectAlbum	Select Album	4	1.013	✓	tester

The exported test results can then be viewed and manipulated in Excel as shown below. Results are exported in detail only to allow maximum manipulation to be carried out in the selected reporting tool.



	A	B	C	D	E	F	G	H	I	J	K
1	Date Time Started	Date Time Completed	Test Suite	Run	Scenario	Test Case	Action	Data Group	Elapsed Time	Pass/Fail	Pass Message
2	31/05/2011 11:03	31/05/2011 11:04	Classics_Order_Creation	Defect123_orders	Classics_Login	Classics_Login	ClassicsCD_Start	Classics_open	10	Pass	
3	31/05/2011 11:04	31/05/2011 11:04	Classics_Order_Creation	Defect123_orders	Classics_Login	Classics_Login	ClassicsCD_MemberLogin	User Login	3	Pass	
4	31/05/2011 11:04	31/05/2011 11:04	Classics_Order_Creation	Defect123_orders	Classics_Order_Create	Classics_OrderEntry	ClassicsCD_Place Order	Classics_Orders	4	Pass	Amounts are eq
5	31/05/2011 11:04	31/05/2011 11:04	Classics_Order_Creation	Defect123_orders	Classics_Order_Create	Classics_OrderEntry	ClassicsCD_ConfirmOrder	ConfirmandSave	23	Pass	Order created.
6	31/05/2011 11:04	31/05/2011 11:04	Classics_Order_Creation	Defect123_orders	Classics_Logout	Classics_Logout	ClassicsCD_Close	Classics_close	1	Pass	
7	31/05/2011 11:04	31/05/2011 11:04	Classics_Order_Creation	Defect123_orders	Classics_Login	Classics_Login	ClassicsCD_Start	Classics_open	4	Pass	
8	31/05/2011 11:04	31/05/2011 11:04	Classics_Order_Creation	Defect123_orders	Classics_Login	Classics_Login	ClassicsCD_MemberLogin	User Login	4	Pass	
9	31/05/2011 11:04	31/05/2011 11:04	Classics_Order_Creation	Defect123_orders	Classics_Order_Create	Classics_OrderEntry	ClassicsCD_Place Order	Classics_Orders	4	Pass	Amounts are eq
10	31/05/2011 11:04	31/05/2011 11:05	Classics_Order_Creation	Defect123_orders	Classics_Order_Create	Classics_OrderEntry	ClassicsCD_ConfirmOrder	ConfirmandSave	23	Pass	Order created.
11	31/05/2011 11:05	31/05/2011 11:05	Classics_Order_Creation	Defect123_orders	Classics_Logout	Classics_Logout	ClassicsCD_Close	Classics_close	1	Pass	
12	31/05/2011 11:05	31/05/2011 11:05	Classics_Order_Creation	Defect123_orders	Classics_Login	Classics_Login	ClassicsCD_Start	Classics_open	5	Pass	
13	31/05/2011 11:05	31/05/2011 11:05	Classics_Order_Creation	Defect123_orders	Classics_Login	Classics_Login	ClassicsCD_MemberLogin	User Login	3	Pass	
14	31/05/2011 11:05	31/05/2011 11:05	Classics_Order_Creation	Defect123_orders	Classics_Order_Create	Classics_OrderEntry	ClassicsCD_Place Order	Classics_Orders	4	Pass	Amounts are eq
15	31/05/2011 11:05	31/05/2011 11:05	Classics_Order_Creation	Defect123_orders	Classics_Order_Create	Classics_OrderEntry	ClassicsCD_ConfirmOrder	ConfirmandSave	22	Pass	Order created.
16	31/05/2011 11:05	31/05/2011 11:05	Classics_Order_Creation	Defect123_orders	Classics_Logout	Classics_Logout	ClassicsCD_Close	Classics_close	1	Pass	
17											
18											
19											
20											
21											
22											

Detailed Level Results Export



CreateOrderSummaryResults.csv - Microsoft Excel

B5 — Create order on Classics B with individual scenarios

	A	B	C	D	E	F	G	H
1	Suite Name	Suite Description	Run Name	Run Description	Row	Elapsed Time	Pass/Fail	Username
2	ClassicCD_CreateOrder	Create order on Classics B with individual scenarios	20120724_AllOrders	Test run for all Orders	0	32	Pass	Classics_demo
3	ClassicCD_CreateOrder	Create order on Classics B with individual scenarios	20120724_AllOrders	Test run for all Orders	1	21	Pass	Classics_demo
4	ClassicCD_CreateOrder	Create order on Classics B with individual scenarios	20120724_AllOrders	Test run for all Orders	2	20	Pass	Classics_demo
5	ClassicCD_CreateOrder	Create order on Classics B with individual scenarios	20120724_AllOrders	Test run for all Orders	3	19	Pass	Classics_demo
6	ClassicCD_CreateOrder	Create order on Classics B with individual scenarios	20120724_AllOrders	Test run for all Orders	4	21	Pass	Classics_demo
7	ClassicCD_CreateOrder	Create order on Classics B with individual scenarios	20120724_AllOrders	Test run for all Orders	0	39	Pass	Classics_demo
8	ClassicCD_CreateOrder	Create order on Classics B with individual scenarios	20120724_AllOrders	Test run for all Orders	1	22	Pass	Classics_demo
9	ClassicCD_CreateOrder	Create order on Classics B with individual scenarios	20120724_AllOrders	Test run for all Orders	0	36	Pass	Classics_demo
10	ClassicCD_CreateOrder	Create order on Classics B with individual scenarios	20120724_AllOrders	Test run for all Orders	1	22	Pass	Classics_demo
11	ClassicCD_CreateOrder	Create order on Classics B with individual scenarios	20120724_AllOrders	Test run for all Orders	0	32	Pass	Classics_demo
12	ClassicCD_CreateOrder	Create order on Classics B with individual scenarios	20120724_AllOrders	Test run for all Orders	1	23	Pass	Classics_demo
13	ClassicCD_CreateOrder	Create order on Classics B with individual scenarios	20120724_AllOrders	Test run for all Orders	0	29	Pass	Classics_demo
14	ClassicCD_CreateOrder	Create order on Classics B with individual scenarios	20120724_AllOrders	Test run for all Orders	1	23	Pass	Classics_demo
15	ClassicCD_CreateOrder	Create order on Classics B with individual scenarios	20120724_AllOrders	Test run for all Orders	2	21	Pass	Classics_demo
16	ClassicCD_CreateOrder	Create order on Classics B with individual scenarios	20120724_AllOrders	Test run for all Orders	0	29	Pass	Classics_demo
17	ClassicCD_CreateOrder	Create order on Classics B with individual scenarios	20120724_AllOrders	Test run for all Orders	1	24	Pass	Classics_demo
18	ClassicCD_CreateOrder	Create order on Classics B with individual scenarios	20120724_AllOrders	Test run for all Orders	0	29	Pass	Classics_demo
19	ClassicCD_CreateOrder	Create order on Classics B with individual scenarios	20120724_AllOrders	Test run for all Orders	1	24	Pass	Classics_demo
20	ClassicCD_CreateOrder	Create order on Classics B with individual scenarios	20120724_AllOrders	Test run for all Orders	0	29	Pass	Classics_demo
21	ClassicCD_CreateOrder	Create order on Classics B with individual scenarios	20120724_AllOrders	Test run for all Orders	1	20	Pass	Classics_demo
22	ClassicCD_CreateOrder	Create order on Classics B with individual scenarios	20120724_AllOrders	Test run for all Orders	0	21	Pass	Classics_demo
23	ClassicCD_CreateOrder	Create order on Classics B with individual scenarios	20120724_AllOrders	Test run for all Orders	1	20	Pass	Classics_demo

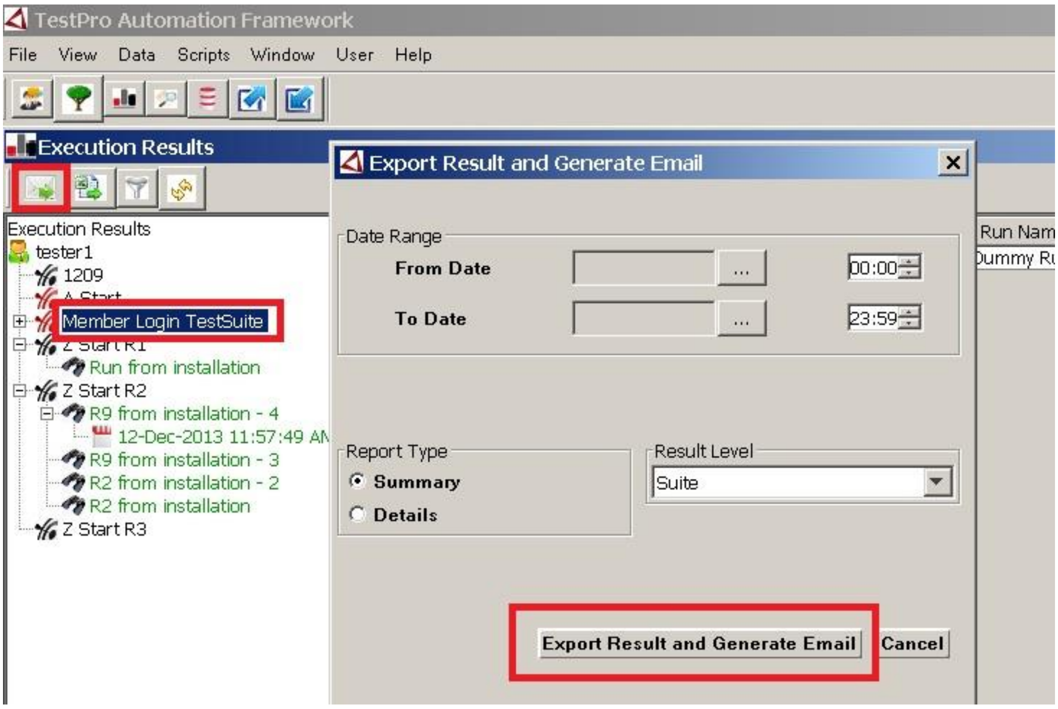
CreateOrderSummaryResults

Summary Level Results Export

11.9 Email Results

User can send a set of results via email in following steps:

1. User selects what results to be sent out. TAF Pro exports the results and save the result as .csv file specified by the user.



Execution Results
- tester1
  - 1209
  - A Start
  - Member Login TestSuite
  - Z Start R1
    - Run from installation
  - Z Start R2
    - R9 from installation - 4
      - 12-Dec-2013 11:57:49 AM
    - R9 from installation - 3
    - R2 from installation - 2
    - R2 from installation
  - Z Start R3

Export Result and Generate Email

Date Range

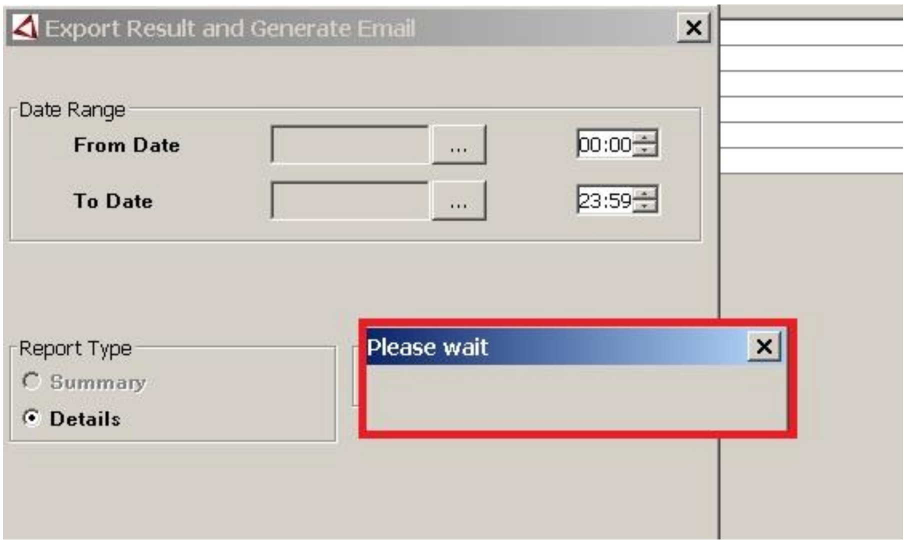
From Date: [ ] ... 00:00

To Date: [ ] ... 23:59

Report Type: ● Summary ○ Details

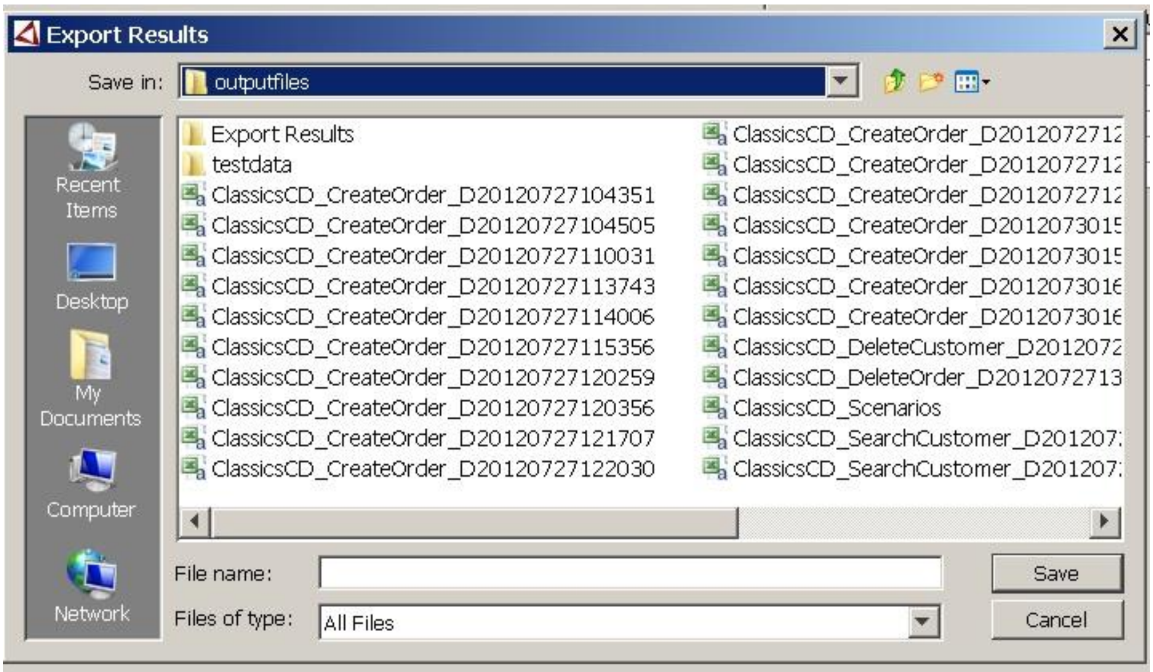
Result Level: Suite

Users can email results from any level by clicking on the required node from the result tree. In this example the result of Test Suite “Member Login TestSuite” will be emailed out. The user then can specify the date/time range in which the result was executed and “Summary” or “Detail” result types. Then click on ***Export Result and Generate Email*** button.



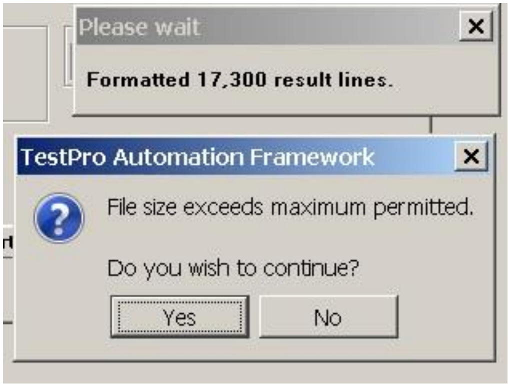
Report Type: ○ Summary ● Details

“Please wait” popup will display if the result-set contains a large volume of data.



Export Results
testdata
ClassicsCD_CreateOrder_D20120727104351
ClassicsCD_CreateOrder_D20120727104505
ClassicsCD_CreateOrder_D20120727110031
ClassicsCD_CreateOrder_D20120727113743
ClassicsCD_CreateOrder_D20120727114006
ClassicsCD_CreateOrder_D20120727115356
ClassicsCD_CreateOrder_D20120727120259
ClassicsCD_CreateOrder_D20120727120356
ClassicsCD_CreateOrder_D20120727121707
ClassicsCD_CreateOrder_D20120727122030
ClassicsCD_CreateOrder_D2012072712
ClassicsCD_CreateOrder_D2012072712
ClassicsCD_CreateOrder_D2012072712
ClassicsCD_CreateOrder_D2012073015
ClassicsCD_CreateOrder_D2012073015
ClassicsCD_CreateOrder_D2012073016
ClassicsCD_CreateOrder_D2012073016
ClassicsCD_DeleteCustomer_D2012072
ClassicsCD_DeleteOrder_D2012072713
ClassicsCD_Scenarios
ClassicsCD_SearchCustomer_D201207
ClassicsCD_SearchCustomer_D201207

2. Specify the path and file name for the exported result file to be saved.
3. TAF Pro checks the size of the exported result file against the threshold value set in User > Preferences > Email Attachments. User can choose to terminate the email process when the result file size exceeds the threshold value, or continue.



Please wait

Formatted 17,300 result lines.

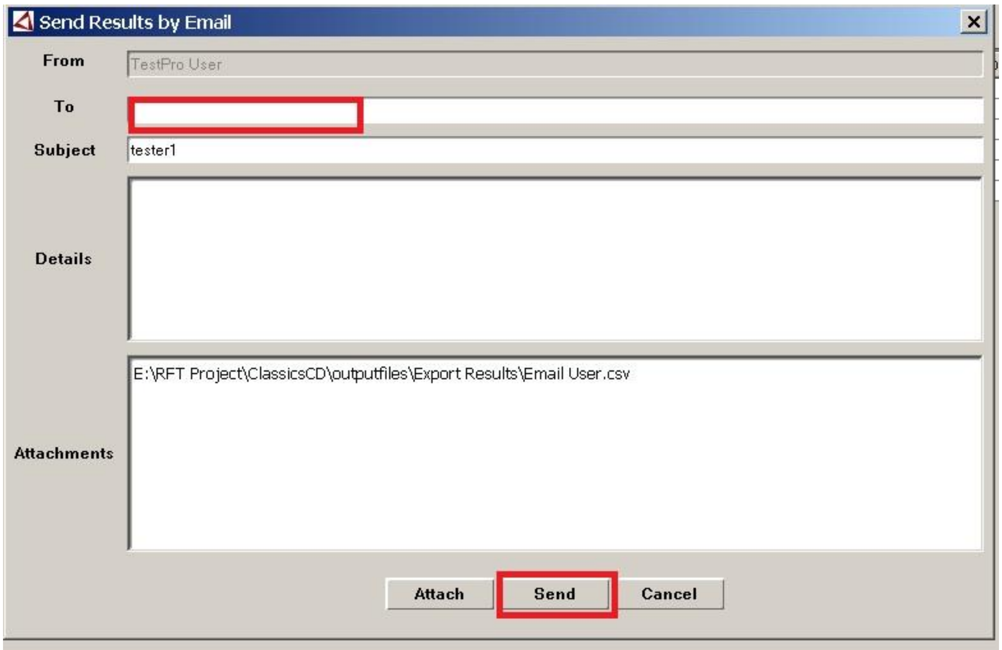
TestPro Automation Framework

File size exceeds maximum permitted.

Do you wish to continue?

This popup will be shown when the size of the exported result file exceeds the size threshold. The User may choose to continue with email process or terminate here.

4. TAF Pro renders a popup for user to fill in email message body with below pre-populated fields:
   - From field: this value is obtained from User > Preferences > Email Setup. Cannot be modified from this popup however can be changed from preferences.
   - To email address: this value is obtained from User > Preferences > Email Setup. User can change this value to fit situations.
   - Subject field: the identity of the result i.e. it's either the User name, Execution Suite name, Run name, Run Row name etc. depending on which result node the user wants. User can change this value as appropriate.
   - Attachments: the exported result file in csv format. User can attach other documents or remove the pre-populated attachment.



Send Results by Email

From: TestPro User

To:

Subject: tester1

Details:

Attachments: E:\RFT Project\ClassicsCD\outputfiles\Export Results\Email User.csv

In this screen shot example, no email address is setup in User Preferences. User can choose to setup the User Preferences for all Email Result operations or type in the email address at this point.

12 Reviewing Execution Logs

TAF Pro maintains two logs monitoring the operation of TAF Pro and a separate log for each environment used.

12.1 TAF Pro logs.

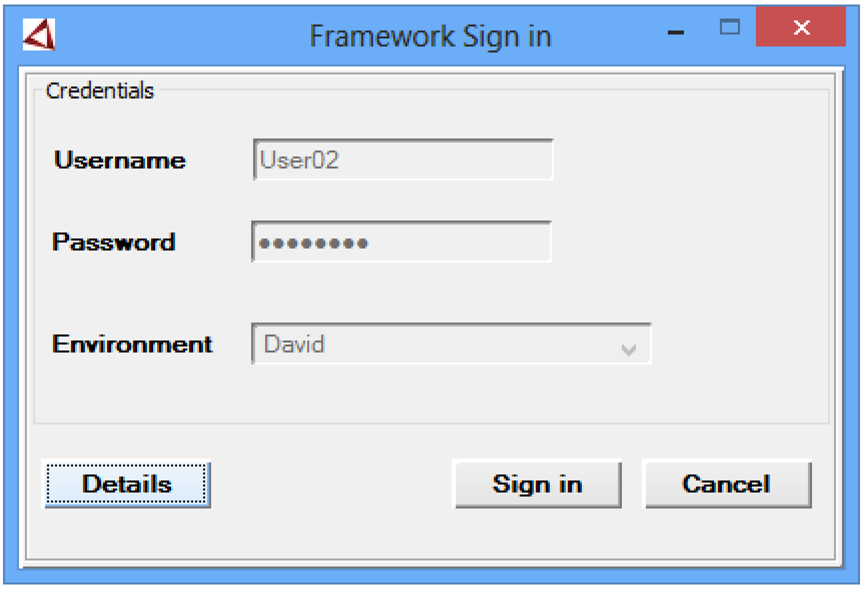
While TAF Pro is in use logs are being compiled in the background to provide the run status of all test executions within TAF Pro:

- The TAF Pro Log (TAF Pro.log) records the status of actions undertaken by TAF Pro and any errors encountered while using TAF Pro. This is the log that is helpful to TAF Pro support when errors are encountered or TAF Pro does not function as expected. The testpro.log file should be readily reviewed and any error occurs this log contains complete messaging whereas the TAF Pro Results often only contains partial messages due to length restrictions and that message can be very long;
- The TAF Pro Tool log (TAF ProTool.log).

The path for the TAF Pro logs is defined in the initial setup of TAF Pro, and can be located by selecting the TAF Pro, ViewLog function in the Windows Start menu.

12.2 Test Tool Logs

When the user logs into a TAF Pro environment the user has the option to define unique locations for the output and log files from the Test Tool.

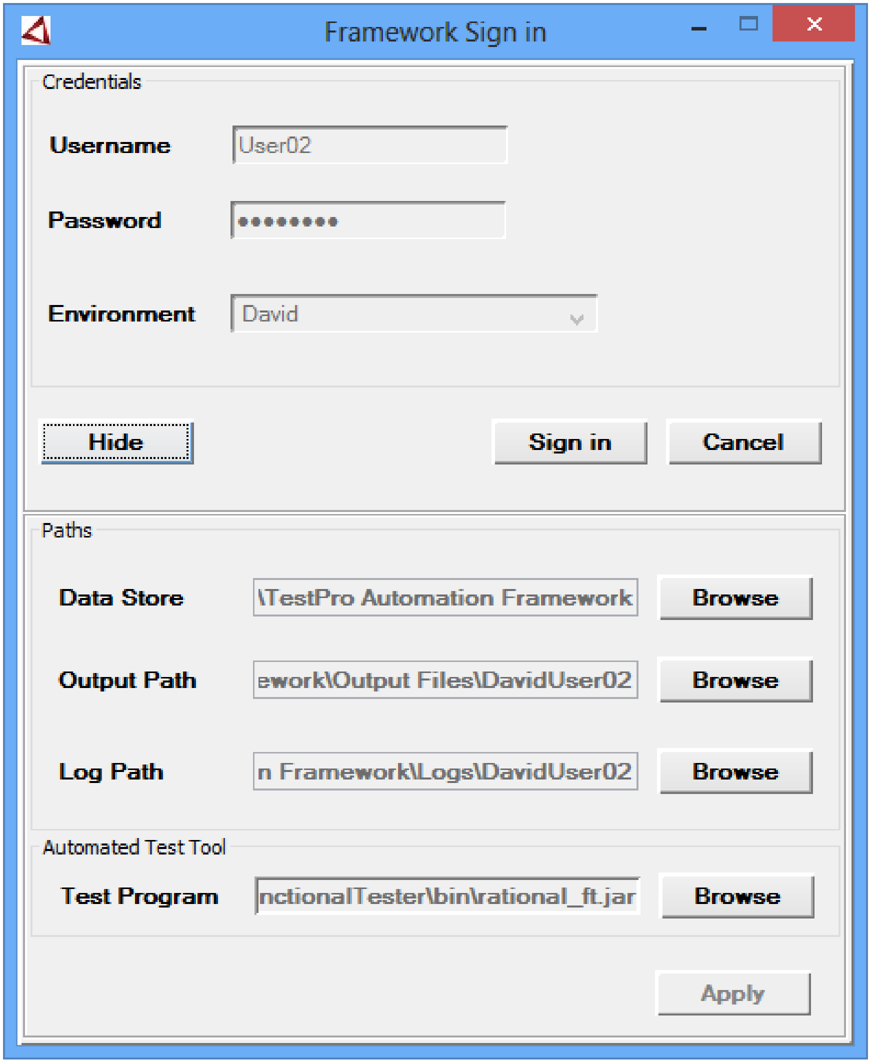


Username: User02

Password: •••••••••

Environment: David

Expanding the Details section allows the user to define the locations for the Data Store, Output Path, and Log Path for the testing tool used in the environment. It is recommended that each user creates separate folders for the outputs and logs within each environment.



Username User02

Password ●●●●●●●●●

Environment David

Data Store \TestPro Automation Framework

Output Path ework\Output Files\DavidUser02

Log Path n Framework\Logs\DavidUser02

Test Program nctionalTester\bin\rational_ft.jar

This log directory stored the output files from each test execution.

As each user's session log will be stored here, it is recommended that these directories be set for the user local drives, to avoid overwriting of logs.

The location of the Logs can be determined from within the TAF Environment by selecting **User → Preferences → Paths**

12.2.1 Viewing test tool log files

To view the Test Tool log files:

1. Navigate to the test tool log path directory;
2. Open the relevant log file folder;
3. Open the rational_ft_logframe file.

13 Effective Techniques when using TAF Pro

13.1 Data Import Shortcuts

The TAF Pro framework provides several enhancements which reduce the amount of work a User needs to enter to input and maintain datasheets.

Like Microsoft Excel, TAF Pro supports the following formula shortcuts:

Formulas are enclosed in an open/close braces **{}** and can be part of the **column value**, e.g. xxxxx{func} including the following values:

1. To alphabetically increment a value, enter {ALPHA}, or optionally, to set the length of the column value, enter **{ALPHA,n**} where n is a value between one **(1) and eight (8).** The default is eight. Every time ALPHA is accessed, the value is incremented, e.g. AAAA, AAAB, AAAC, etc.
2. To numerically increment a field, enter **{NUMERIC},** again with an optional length value, {NUMERIC,n}. The default length is again eight digits. E.g. 1000, 1001, 1002 etc.
3. **Date arithmetic** is handled via this formula, **{D_XS999},** where **D_** is constant (always starts with D_ which means todays date), **X can be D (days), M (months) or Y (years). S is a sign, + (plus) or - (minus).** Default is plus, hence only really need to put in a minus. Last one to three digits is the number of days, months or years by which today's date is to be adjusted.

   Examples: D_D0 will always return today's date.

   D_M3 will return today plus three months.

   D_D-30 will return today's date less 30 days.

   D_Y1 will return today's date plus one year.
4. To explicitly cater for when no data is to be entered into a cell the **{BLANK}** function can be used. This should not be confused with a data cell which really is blank which conveys a totally different meaning. If a blank cell is found on the second or any later data row (i.e. not a heading row), TAF Pro automatically inserts the last known value in that cell from the previous row.
5. To carry values forward from one row to the next, enter the value required in the first row and then leave additional rows blank. This will default the value from above.
6. The functions should not be conjoined in any single cell

Table 20-1he table below is an example showing how some of these features work in practice.

Column 1	Column 2	Column 3	Column 4	Column 5	Column 6	Column 7	Column 8	Column 9	Column 10	Column 11
{BLANK}	Aaa	Bbb		ccc	ddd		Eee	{BLANK}	222	111
		Xxx						222		
	Vvv		yyy	{BLANK}		yyy			{BLANK}	

Table 20-1 Sample data Sheet

Data Row 1reads as follows:

H1 = blank out the value in the field on screen as there may be a default value
H2 = aaa value entry
H3 = bbb value entry
H4 = accept the field as it is, no change to the field
H5 = ccc value entry
H6 = ddd value entry
H7 as for H4
H8 = eee value entry
H9 = as for H1
H10 = 222 value entry
H11 = 111 value entry

Data Row 2 reads as follows:

H1, H2 same value as line above
H3 = xxx value entry
H4, H5, H6, H7, H8 = same value as line above. It too is blank so value in the field will be accepted
H9 = 222 entry value
H10, H11 = as per line above (if it was set to blank that this carries down too)

Data Row 3 reads as follows:

H1 = Same value as line above which in this case inherited a blank from line 1
H2 = vvv value entry
H3 = as per line above
H4 = yyy value entry
H5 = remove value set in H5 and accept field empty
H6 = as per line above
H7 = yyy value entry

H8, H9 = value as per line above
H10 = remove default value and leave field empty
H11 = as per line above

13.2 Naming your TAF Pro Artifacts

As a matter of best practice when naming your test artefacts in TAF Pro, avoid using **special characters**. The following characters are accepted like ‘**-**‘ (dash), ‘**_**’ (underscore) and ‘**.**’ (full-stop) as separators in your names, but if you are going to use Command Line Execution, you cannot use ‘**&**’ (ampersand), as this is an XML Special Character and is used by **ANT** itself. Using ‘**<**’ (less than), ‘**>**’ (greater than) and ‘**&**’ (ampersand) will cause an error and your CLI execution will fail.

13.3 TAF Pro Refresh after Script Changes

The TAF Pro framework currently does not provide an automatic mechanism to update automate test scripts in the Execution Suite after they have been modified in the Automated Test Tool whilst TAF Pro is open. The recommended procedure is as follows:

If IBM Rational Functional Tester modifications have taken place whilst TAF Pro is open, close down TAF Pro and re-start it to pick up the changes. Follow the steps to Register Scripts, and then **Regenerate** the scenarios within the **Execution Suites** that use the script.

13.4 Stopping Test Execution

If a TAF Pro **Execution Suite** is running either via the **GUI** or **Command Line interface,** it is possible to stop the test if necessary.

To stop on-screen test execution via TAF Pro GUI:

1. Press **<F12>** to pause **IBM Rational Functional Tester** – (assuming that RFT is the Automated Test Tool you are using with TAF Pro);
2. Click into the **Execution Suites** window in **TAF Pro;**
3. Press the **[Esc] key** - to send a command to TAF Pro to terminate execution at the end of the currently executing record of the **Execution Suite;**
4. Click on the IBM Rational Functional Tester ‘**Playback**’ window
5. Click the ‘**Resume playback’** button on the toolbar to allow the current record to execute in the remainder of the **Planned Scenarios** in the current **Execution Suite**. If you have multiple Planned Scenarios in your Execution Suite, this may take some time while the current iteration of the test is completed.
6. The execution has finished when the Playback window displays **‘scripts.loader’** below the toolbar.

To stop Command Line Interface test execution:

1. Invoke Task Manager

2. Select the Java Task
3. Select End Task

13.5 Check Filters

If a datagroup is refreshed for any reason, any filters should be checked to **"filter data"** to ensure the filters are still valid.

Any filters that no longer return data will result in an error when the test suite is executed with the selected filter.

To avoid this issue, follow the steps in Section 9.2.4 to display filtered data.

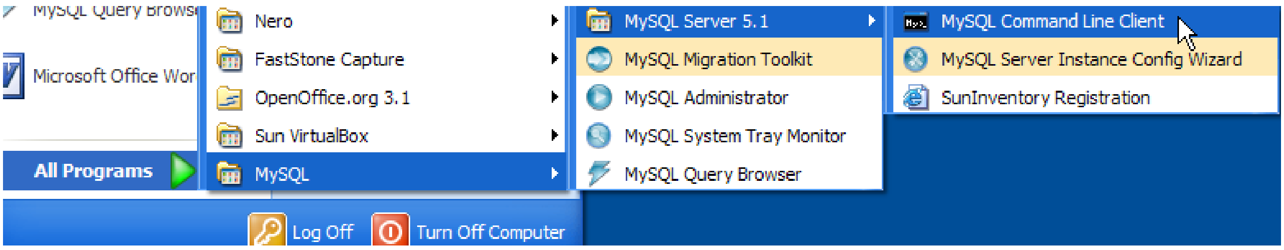
13.6 Trouble Shooting

13.6.1 MySQL Security Settings Error

Error:
Server Configuration Wizard issues **Error 1045** during 'MySQL Apply Security Settings step.

SOLUTION:
Start the MySQL command line client. Press ENTER when asked for a password.
Change the password by running the following line.

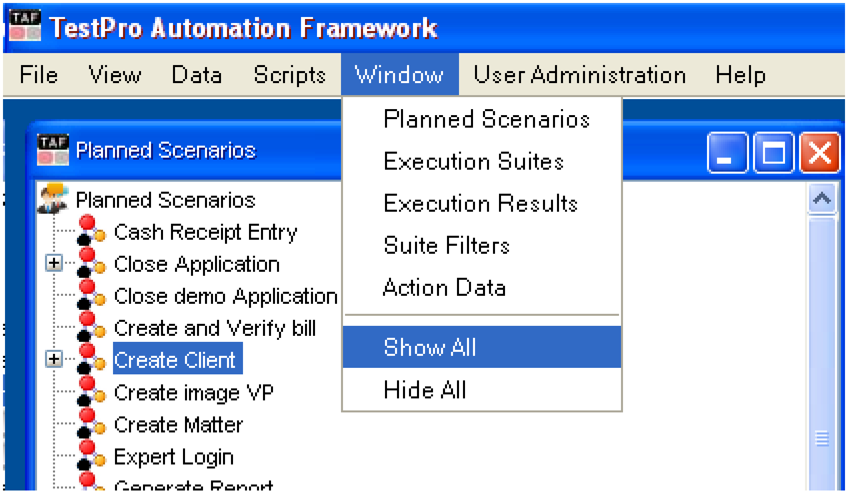
SET PASSWORD FOR 'root'@'localhost' = PASSWORD ('root');



13.6.2 TAF Pro Windows not visible

ERROR:
The five TAF Pros windows are not displayed.

SOLUTION:
At TestPro Automation Framework **Menu bar** choose ***Window>Show all.***



13.6.3 TAF Pro Crashes when F11 is pressed.

ERROR:
When I hit <F11>TAF Pro crashes.

SOLUTION:
This is a feature of Windows. It is not the only application where **<F11>** is effective. If there is **an IBM Rational Functional Tester** window open, and the user is in another unrelated application, (Excel for example) and they press **F11**, IBM Rational Functional Tester still closes. Hence the only way to avoid inadvertently closing down TAF Pro is to refrain from using **<F11>** key whilst TAF Pro is running.

13.6.4 Performance Issues on 64-bit Installations

ERROR:
TAF Pro runs slowly on 64 bit installations.

SOLUTION:

Some implementations using IBM Rational Functional Tester as the Automated Test Tool on 64-bit machines have exhibited unsatisfactory performance. This has been rectified by changing the Java instance utilised, changing to ORACLE Java rather than the RFT internal Java instance. The recommended installation should be followed and this setting changed only if IBM Rational Functional Tester appears to be running very slowing.

To update the JAVA Setting go to **User > Preferences > Java**. The JAVA setting can be changed from the IBM Java to ORACLE Java and back again. Each time this setting is changed the user should log out of TAF Pro and log back in to bring the change into effect.

14 Using Test Management Tools

14.1 Test Execution with Rational Quality Manager

To enable Test Execution with Rational Quality Manager:

1. Set up the Execution Suite in TAF Pro;
2. Set up the Execution Suite for Execution via Command Line Interface;
3. Rational Quality Manager Adapter has to be installed and enabled before executing TAF PRO.

14.1.1 Set up CLI for Rational Quality Manager Test Execution

1. Create a .bat file in either;
   - Where the directory path is your TAF Pro installation Path, or
   - Where the ANT command contains the path and filename where the XML to be run is located.
2. Add the following three lines to the .bat file:

```
cd C:\Program Files\TestPro\TestPro Automation Framework
ant -lib lib -lib . -f
"C:\Program Files\TestPro\TestPro Automation Framework\Build.xml"
```

3. **Save the .bat** with a name that is meaningful to the test to be executed.

14.1.2 Set up Rational Quality Manager for Test Execution

1. Go to **DOS prompt** and navigate to the path where your TAF Pro / Rational Quality Manager Adapter is installed, e.g. C:\Users\Public\TestPro\TafRQMadapterClient
2. Run **Start.bat**
3. Leave this DOS window open. Closing the window will terminate the connection.
4. Open Web Browser and **enter the URL** to connect to Rational Quality Manager. E.g. https://vhost1234.dc1.sg.ap.compote.ahost.com:9443/qm/web/console
5. **Login** to Rational Quality Manager user login (Professional type User)
6. **Create** a Test Script selecting TAF Pro as the Script Type
7. **Scroll down** to the TAF Pro Artefact Editor
8. **Select Location** for test Resources. Local or Shared
9. Click on **Select Adapter**
10. Click **NEXT**
11. **Set the project path** to where the.bat file you created in Set up for Rational Quality Manager Test Execution is located
12. **Select** the .bat file created in step above….
13. Select **SAVE**
14. **Associate** the Test script to a Test Case and Test Suite
15. **Execute** your Test Suite or Test Cases as needed.

Note: Check Adapter Status is Healthy before the Test Suite or Test Case is executed.

14.1.3 View Results after the Test is Completed

1. Click on Close and Show Results Button.
2. Scroll down to the results Detail Sections and Click on the Actual Result Column Link
3. The following Reports are accessible from the results:
    - Command Line Integration Log File;
    - Command Line Execution Result File - .csv results;
    - IBM Rational Functional Tester Log files.

14.1.4 Public URI Changes in Rational Quality Manager

From time to time it may be necessary to change the public URI of Quality Manager. Once the change is made in Rational Quality Manager the following steps must be completed on the TAF Pro Client to maintain connectivity with Rational Quality Manager.

Setting the Public URI of Rational Quality Manager is configurable via 2 methods:

- run argument with -repository option, or
- config.ini rqm.repository= entry.

After the public URI has been changed, the TAF Pro System Administrations will need to perform the following steps to re-establish the connection to Rational Quality Manager.

1. Stop the adapter
2. Updates either the value of run argument -repository or config.ini rqm.repository entry
3. Restart the adapter.

14.2 Test Execution with HP Quality Center

To enable Test Execution with HP Quality Center:

1. Set up the Execution Suite in TAF Pro.
2. Set up the Execution Suite for Execution via Command Line Integration

14.2.1 Set up CLI for HP Quality Center Test Execution

1. Create a .bat file in either
    - Where the directory path is your TAF Pro installation Path, or
    - Where the ANT command contains the path and filename where the XML to be run is located.
2. Add the following 2 lines to the .bat file

```
cd C:\Program Files\TestPro\TestPro Automation Framework
ant -lib lib -f "C:\Program Files\TestPro\TestPro Automation
framework\Start_Login_Close_Build.xml"
```

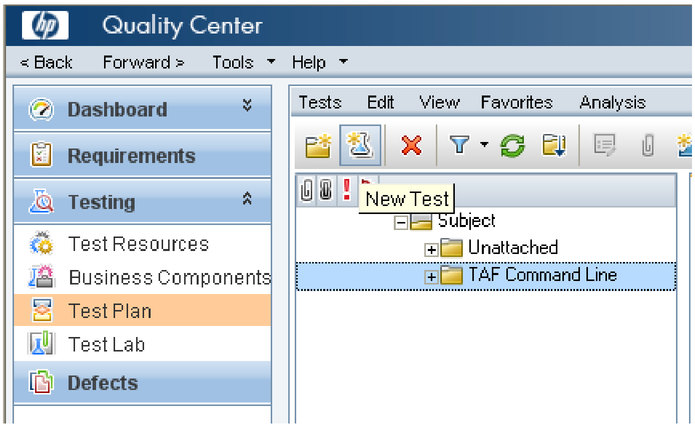
3. Save the .bat with a name that is meaningful to the test to be executed

14.2.2 Set up HP Quality Center for Test Execution

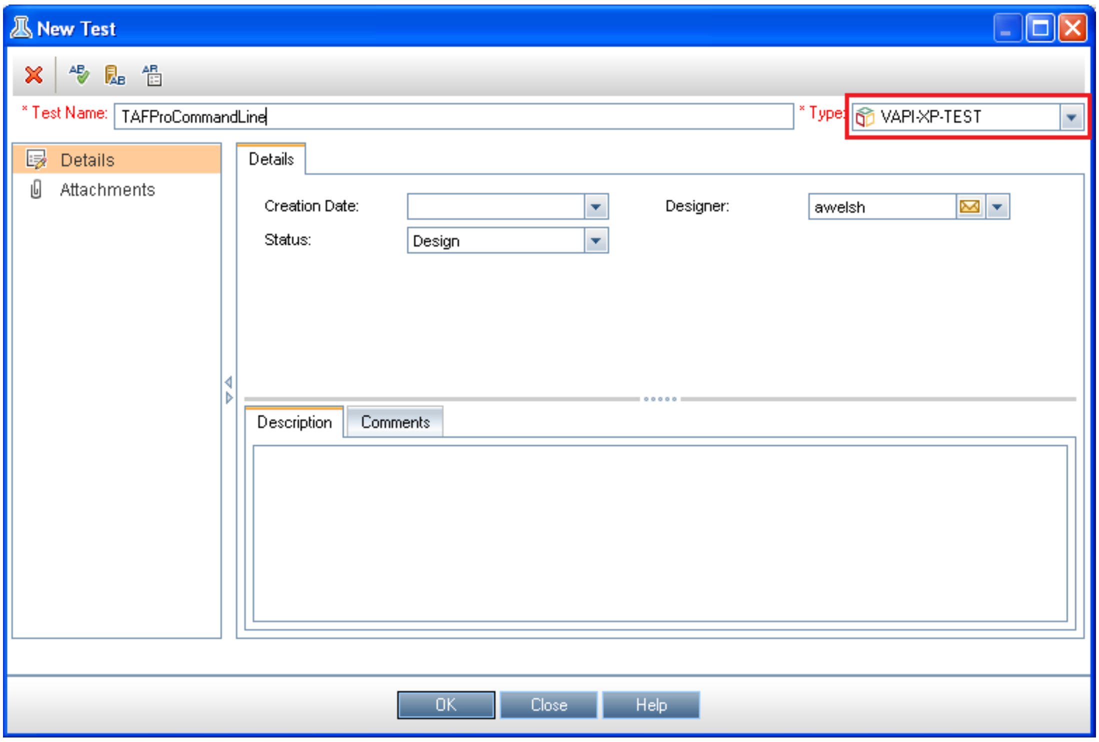
14.2.2.1 Create a VAPI-XP-TEST

To run a TAF Pro command line test in Quality Center a new VAPI-XP-TEST needs to be created:

1. Add a new "Test Plan" area as shown below;

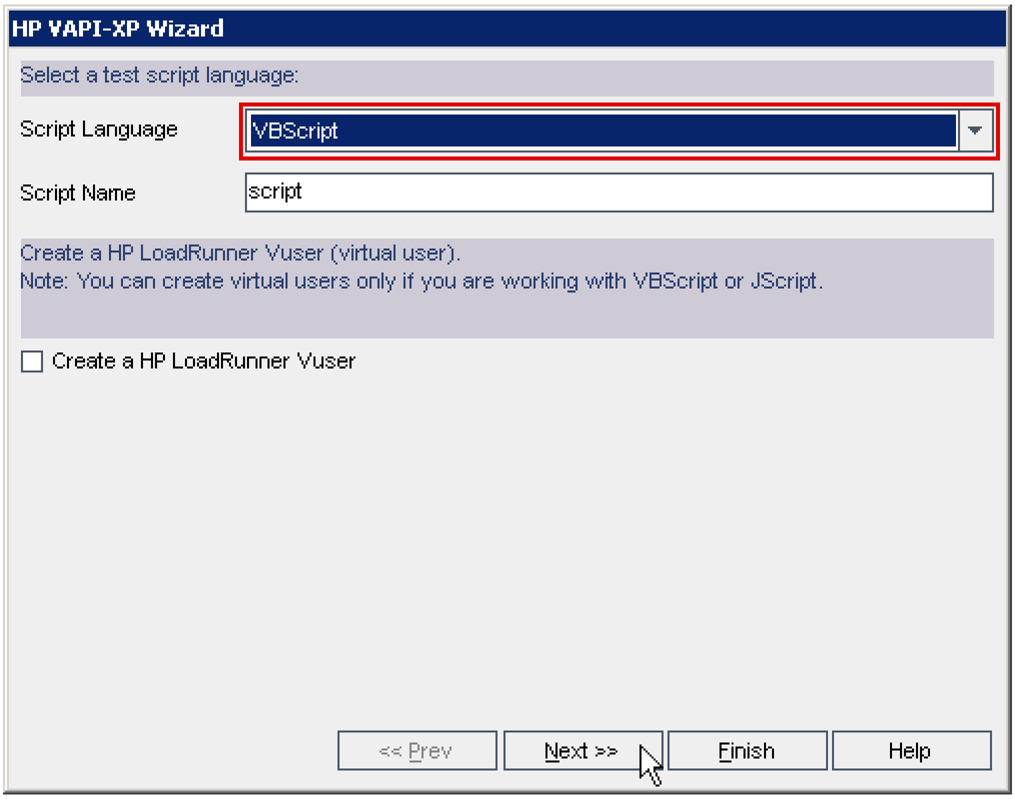


2. Set the test type to "VAPI-XP-TEST";



* Test Name: TAFProCommandLine
* Type: VAPI-XP-TEST
Creation Date:
Designer: awelsh
Status: Design

3. Set Script language to VBScript;



HP VAPI-XP Wizard

Select a test script language:

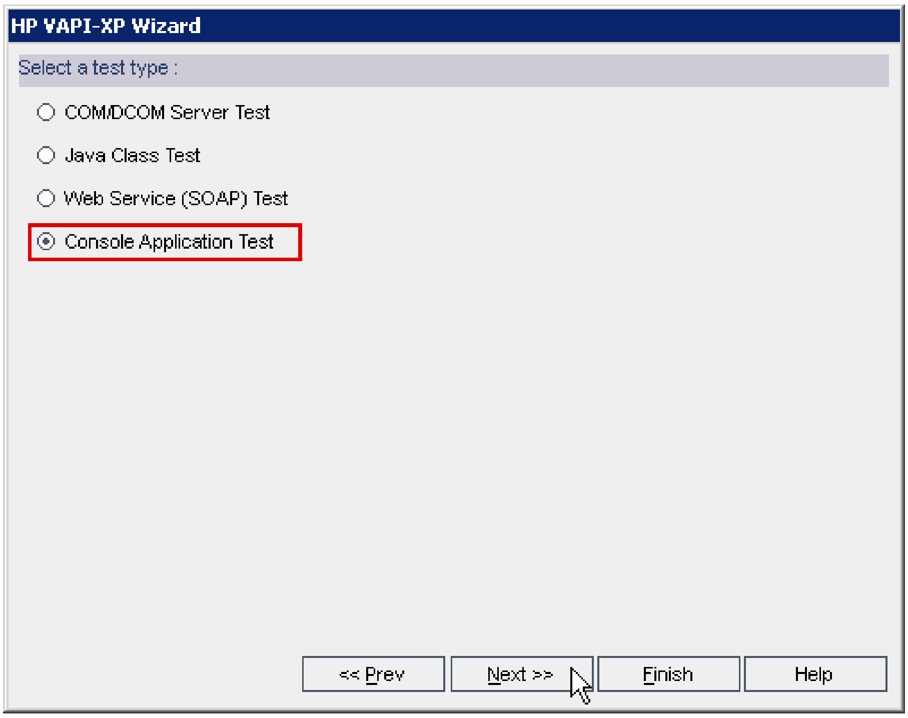
Script Language: VBScript

Script Name: script

Create a HP LoadRunner Vuser (virtual user).
Note: You can create virtual users only if you are working with VBScript or JScript.

☐ Create a HP LoadRunner Vuser

4. Set the test type as “Console Application Test”;



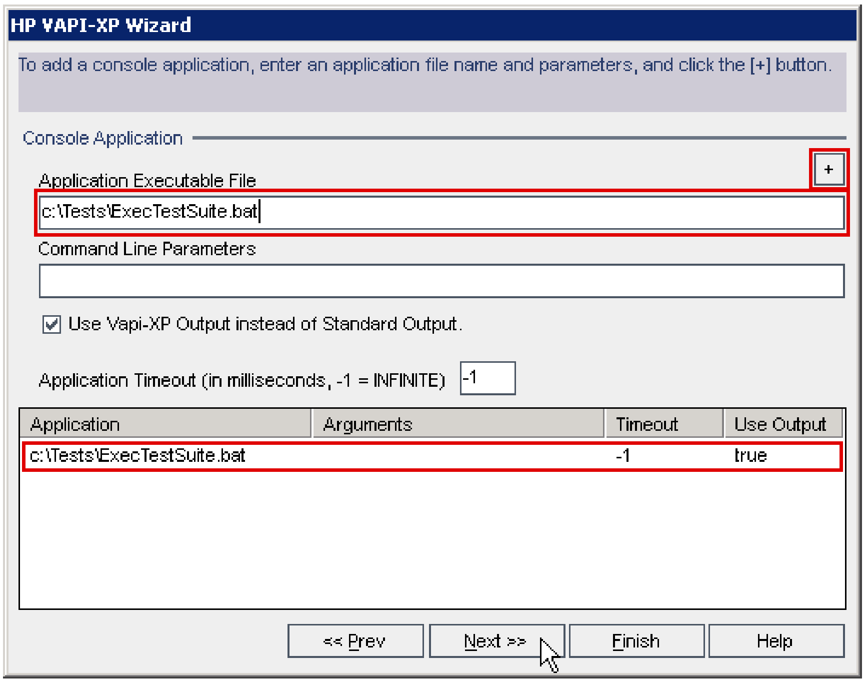
HP VAPI-XP Wizard

Select a test script type :

○ COM/DCOM Server Test
○ Java Class Test
○ Web Service (SOAP) Test
● Console Application Test

5. Enter the batch file location in “Application Executable File”;
6. Press the “+” button to add the application call to the script;

7. Select the Checkbox to “Use VAPI-XP Output instead of Standard Output”;



HP VAPI-XP Wizard

To add a console application, enter an application file name and parameters, and click the [+] button.

Console Application

Application Executable File: c:\Tests\ExecTestSuite.bat

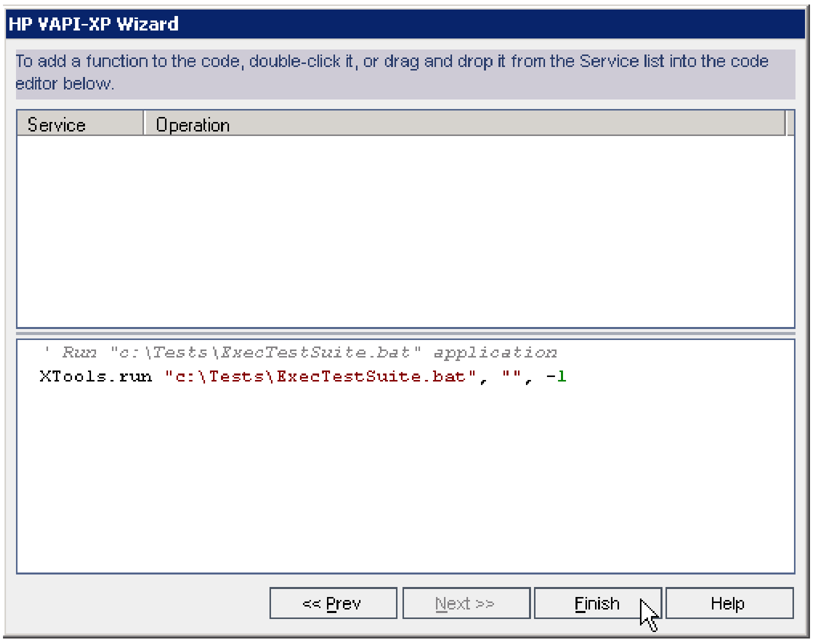
Command Line Parameters:

☑ Use Vapi-XP Output instead of Standard Output.

Application Timeout (in milliseconds, -1 = INFINITE) -1

Application	Arguments	Timeout	Use Output
c:\Tests\ExecTestSuite.bat		-1	true

8. Finish the HP VAPI-XP Wizard.



HP VAPI-XP Wizard

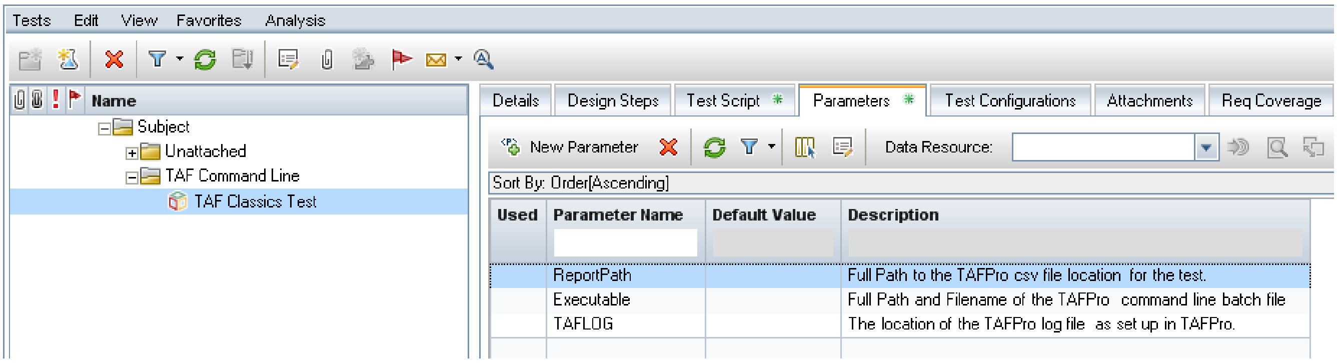
To add a function to the code, double-click it, or drag and drop it from the Service list into the code editor below.

Service	Operation

```
' Run "c:\Tests\ExecTestSuite.bat" application
XTools.run "c:\Tests\ExecTestSuite.bat", "", -1
```

14.2.3 Add Parameters to Specify the TAF Pro Automation

After the Test Plan has been set up, three parameters need to be added to the Test for Report Path, Executable and TAF Pro Log locations.

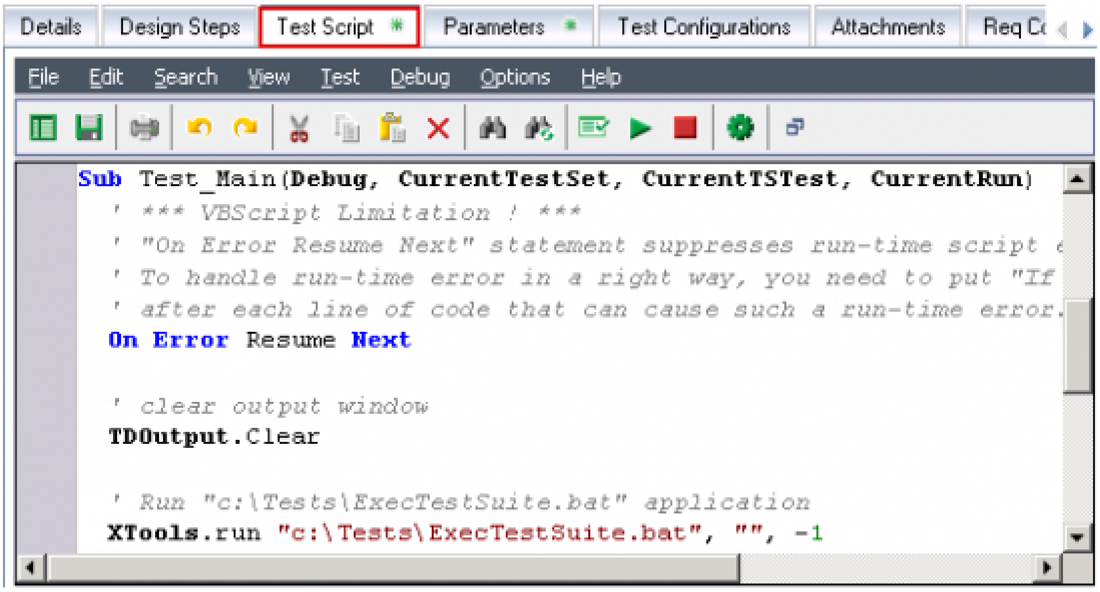


Used	Parameter Name	Default Value	Description
	ReportPath		Full Path to the TAFPro csv file location for the test.
	Executable		Full Path and Filename of the TAFPro command line batch file
	TAFLOG		The location of the TAFPro log file as set up in TAFPro.

1. ***Select*** the "Parameters" tab
2. ***Click*** on 'New Parameter"
3. ***Add "ReportPath"*** in the "Parameter Name" field.
4. ***Click OK*** to save the new parameter
5. ***Add "Executable***" in the "Parameter Name" field.
6. ***Click OK*** to save the new parameter
7. ***Add "TAFLOG"*** in the "Parameter Name" field.
8. ***Click OK*** to save the new parameter

14.2.4 Modify the VAPI-XP-TEST script

The default script generated by the VAPI-XP test wizard (shown below) is inadequate. It must be enhanced to support TAF Pro test reporting.



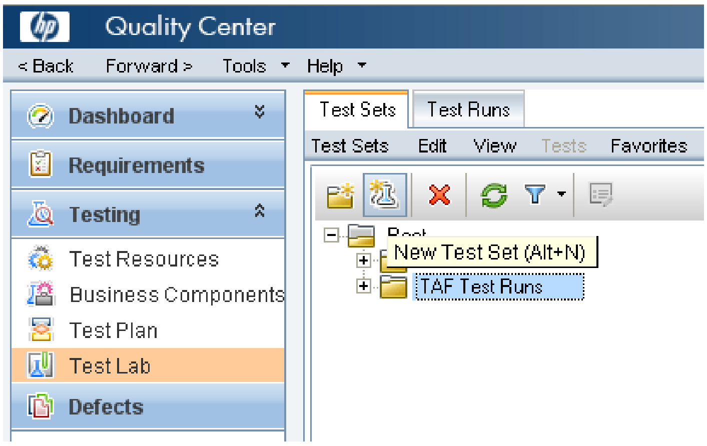
```
Sub Test_Main(Debug, CurrentTestSet, CurrentTSTest, CurrentRun)
  ' *** VBScript Limitation ! ***
  ' "On Error Resume Next" statement suppresses run-time script e
  ' To handle run-time error in a right way, you need to put "If
  ' after each line of code that can cause such a run-time error.
  On Error Resume Next

  ' clear output window
  TDOutput.Clear

  ' Run "c:\Tests\ExecTestSuite.bat" application
  XTools.run "c:\Tests\ExecTestSuite.bat", "", -1
```

The code listed in Appendix C of this User guide should be copied and pasted into the “Test Script” in Quality Center to replace the script that is generated.

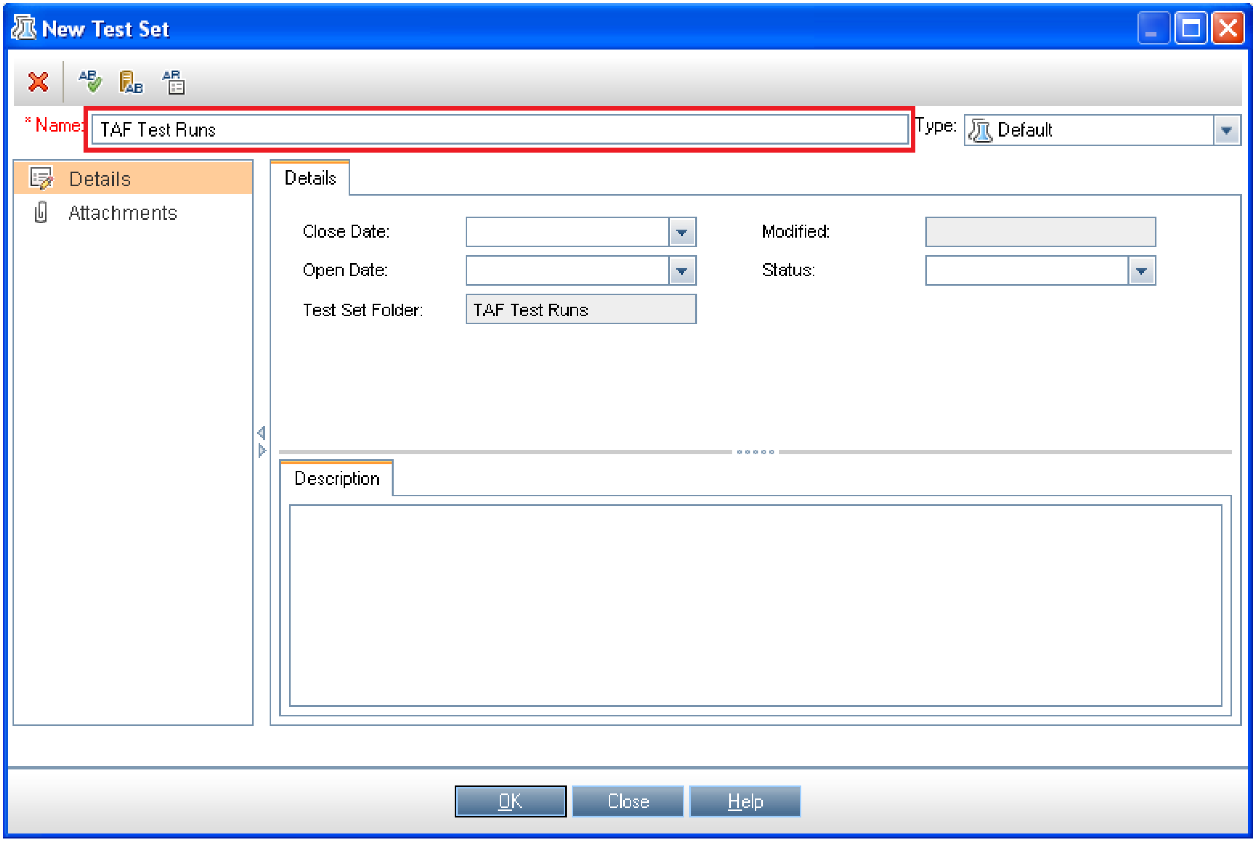
14.2.5 Create a Test Set for VAPI-XP-TEST Execution

After defining the Test in the “Test Plan” area, you can create a new test set in the “Test Lab” area to execute the VAPI-XP-TEST created earlier:

1. Switch from the “Test Plan” to the “Test Set” area;
2. Create a new test set;



3. The New Test Set Screen is displayed;
4. Enter the Test Set Name to suite the test being run;



New Test Set

* Name: TAF Test Runs

Type: Default

Close Date:

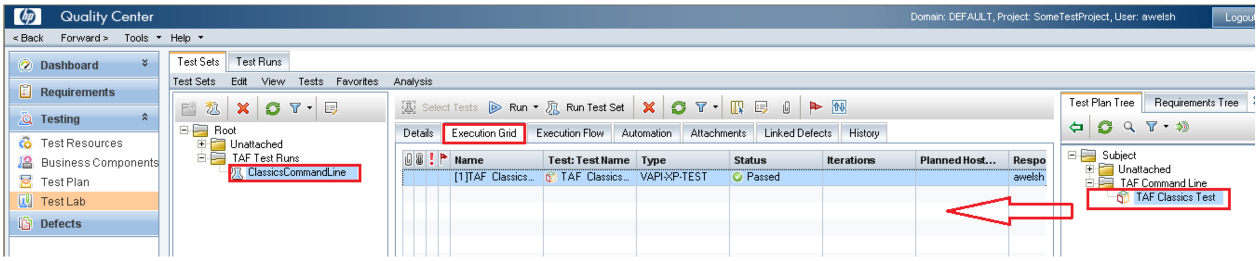
Open Date:

Test Set Folder: TAF Test Runs

Modified:

Status:

5. Add the previously created VAPI-XP-TEST to the test set by dragging and dropping to the execution grid;



Domain: DEFAULT, Project: SomeTestProject, User: awelsh

Root
- Unattached
- TAF Test Runs
  - ClassicsCommandLine

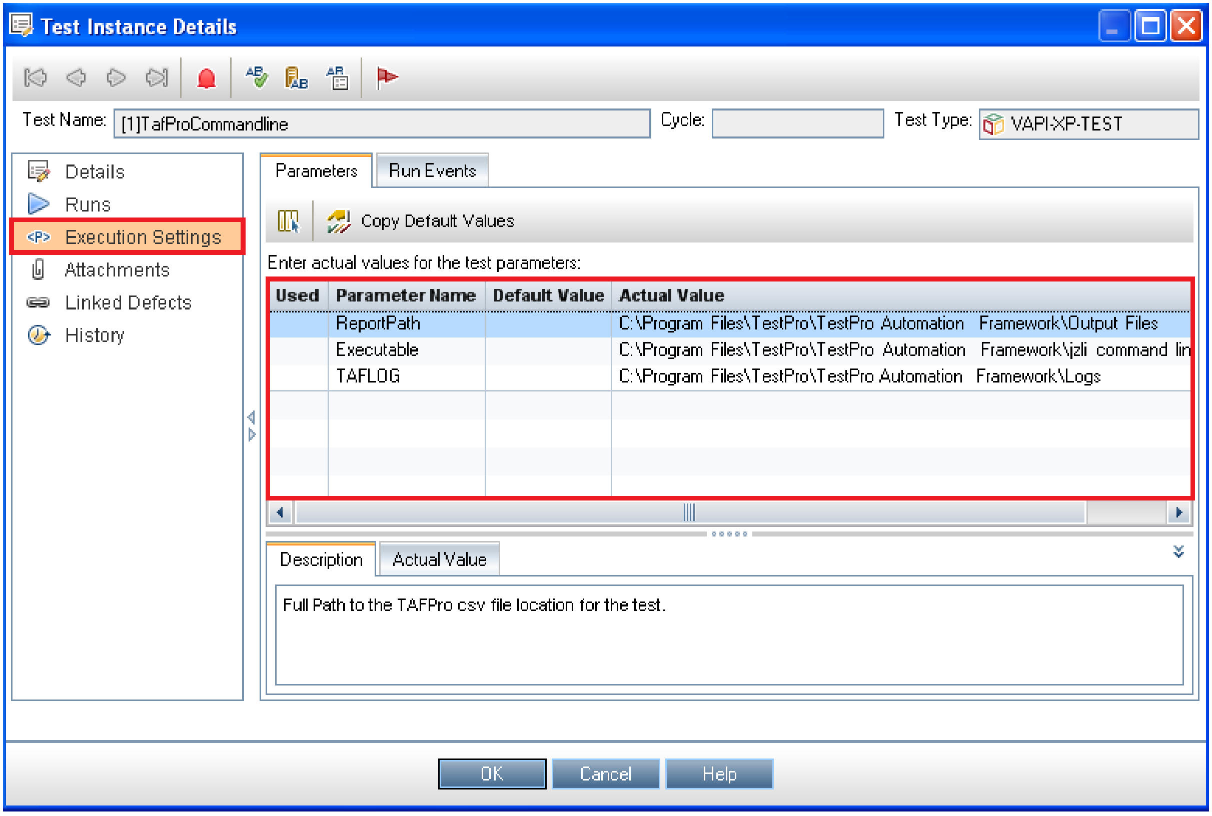
Name	Test: Test Name	Type	Status	Iterations	Planned Host...	Respo
[1]TAF Classics...	TAF Classics...	VAPI-XP-TEST	Passed			awelsh

Subject
- Unattached
- TAF Command Line
  - TAF Classics Test

14.2.6 Parameter Set Up

Parameter values need to be set up for the test to execute:

1. Open the test details by double clicking on the test in the execution grid;
2. The “Test Instance Details” window is displayed;



Test Name: [1]TafProCommandline Cycle: Test Type: VAPI-XP-TEST

Enter actual values for the test parameters:

Used	Parameter Name	Default Value	Actual Value
	ReportPath		C:\Program Files\TestPro\TestPro Automation Framework\Output Files
	Executable		C:\Program Files\TestPro\TestPro Automation Framework\jzli command lin
	TAFLOG		C:\Program Files\TestPro\TestPro Automation Framework\Logs

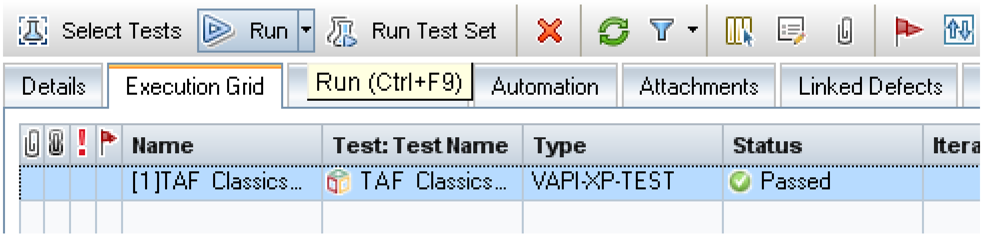
Full Path to the TAFPro csv file location for the test.

3. Select “Execution Settings” from the list, this displays the current parameters for the test. For the two parameters, click the “Actual Value” tab and enter in the details;
4. ReportPath: Full path to the TAFPro csv results for this test. This should be unique for each TAF Pro command line test you set up.
5. Executable : Full path and filename for the TAF Pro command line batch file
6. TAFLOG: Full path to the TAF log file. This should be the same folder as the one set up in TAF.

14.2.7 VAPI-XP-TEST Execution

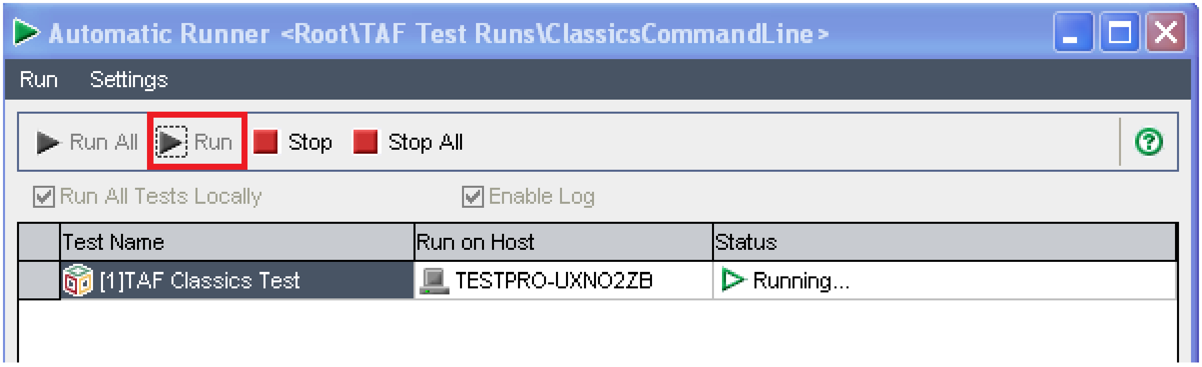
After set up is complete the test is ready to be executed:

1. From the Test Lab select the test to be executed;



Name	Test: Test Name	Type	Status	Itera
[1]TAF Classics...	TAF Classics...	VAPI-XP-TEST	Passed	

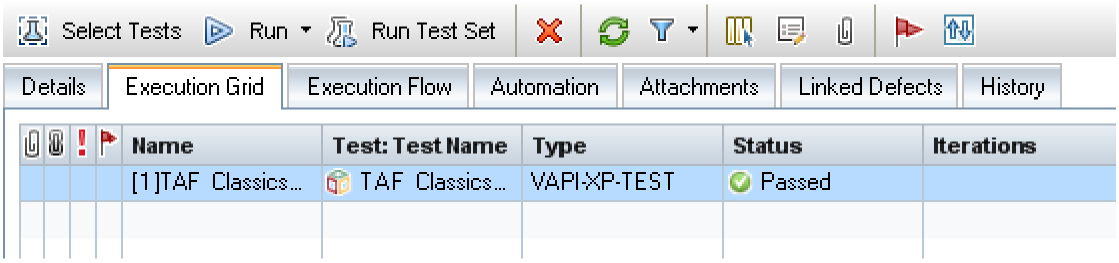
2. Click on the Run Button;
3. The Status of the Test will change to Running when the test execution has started;



☑ Run All Tests Locally ☑ Enable Log

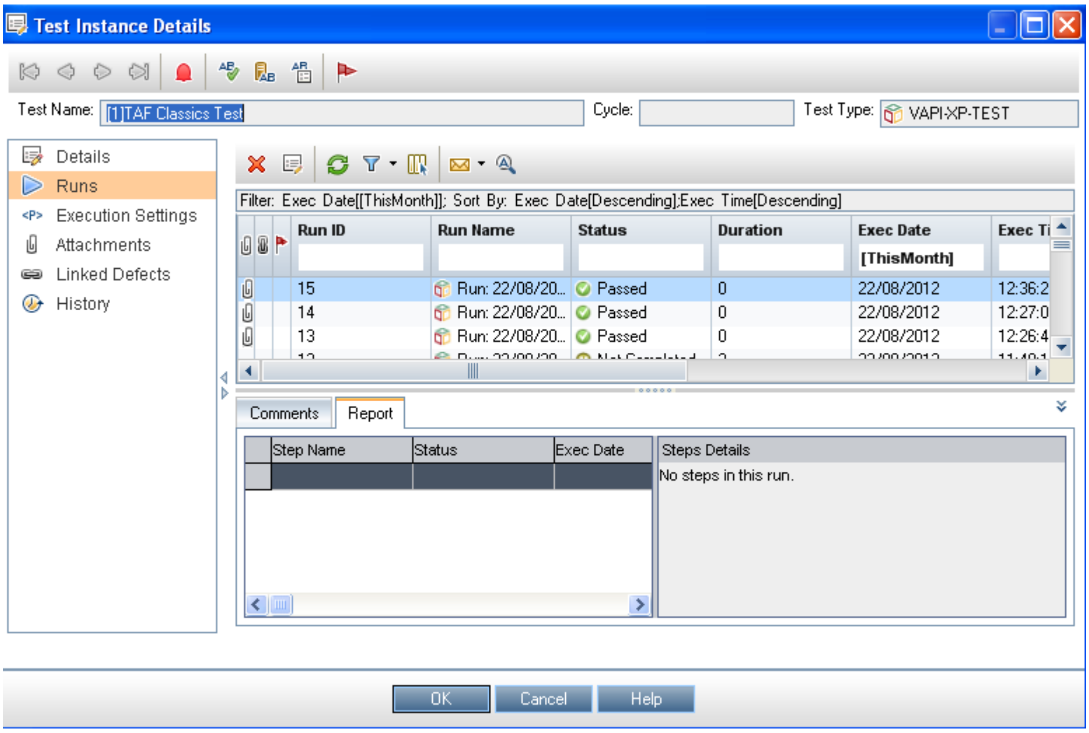
Test Name	Run on Host	Status
[1]TAF Classics Test	TESTPRO-UXNO2ZB	Running...

4. When the test is completed the status of the Test will change to Passed or Failed as shown below, depending on the test outcome;



Name	Test: Test Name	Type	Status	Iterations
[1]TAF Classics...	TAF Classics...	VAPI-XP-TEST	Passed	

5. Double Click on the Test name in "Test Instance Details" window to view results and access the TAF Pro csv results which are returned from the test execution;



Test Name: [1]TAF Classics Test Cycle: Test Type: VAPI-XP-TEST

Filter: Exec Date[[ThisMonth]]; Sort By: Exec Date[Descending];Exec Time[Descending]

Run ID	Run Name	Status	Duration	Exec Date	Exec Ti
				[ThisMonth]	
15	Run: 22/08/20...	Passed	0	22/08/2012	12:36:2
14	Run: 22/08/20...	Passed	0	22/08/2012	12:27:0
13	Run: 22/08/20...	Passed	0	22/08/2012	12:26:4

Step Name	Status	Exec Date

Steps Details
No steps in this run.

6. To view the test results returned from TAF Pro double-click the run in the run grid as in **Error! Reference source not found.** below.

14.2.8 Creating Additional TAF Pro Command Line Tests

To create a new TAF Pro test in HP Quality Center:

1. Copy the original test created earlier and rename;
2. When adding the new test to the Test Lab, update the parameters to point to the correct csv log and batch file for the new TAF Pro test.

15 Appendices

15.1 Appendix A – Glossary

Action	A labelled set of instructions contained within a Test Script. An action in a script represents an option selected within the script allowing for the reuse of common activity and reduction in the number of scripts that need to be created.
Common Keyword	The Common Keyword is used by filters in DataGroups to select which data items are eligible for each test execution. The Common Keyword normally represents the test being run e.g. UAT, SIT, Smoke.
CLI	Command Line Interface. A method of Executing suites of test cases outside of the TAF Pro GUI.
DataGroup	Data imported from a Spreadsheet into the framework is held in DataGroups, a collection of associated data required by TAF Pro during Execution Suite execution.
Environment	
Filters	Filters are used to select the data against which a particular Execution Suite will run.
Framework	See TestPro Automation Framework.
TAF Pro	See TestPro Automation Framework.
Test Case	A Test Case is a container for actions and may contain more than 1 action.
TestPro Automation Framework	This test automation framework.
Scenario	A Scenario is the basic execution component within the TAF Pro. It is a container for Test Cases and may contain 1 or more functionally related Test Cases.
Script	A Rational Functional Test Script. A series of instructions created in Rational to drive the application under test in a defined and known manner.
Sequence	Referred to in 7.2.1.1
Suite	A Suite is a grouped set of Scenarios, (a container for Scenarios) the scenarios may be logically sequenced with data flowing between them or logically grouped with no associated data between them.
Rational IBM Rational Functional Tester	This is a test automation tool from the IBM Rational test tool suite of products.

15.2 Appendix B – Detailed descriptions of Menu Options

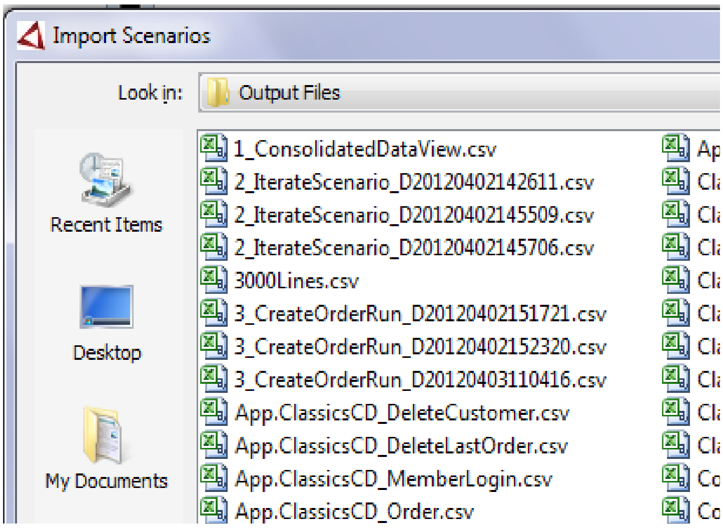
15.2.1 File Menu Functions in Detail

15.2.1.1 Import Functions

The ***File → Import*** function provides TAF Pro users with the capability to import Scenario and Test Case name and descriptions. This is an alternative to importing via the Execution Suite import function which is the simpler method. These files can then be used to import Scenario and Test Case names from another TAF Pro environment. This facility provides the ability to maintain naming conventions across environments and avoid re-keying large amounts of data when a new environment is created.

15.2.1.2 Importing Scenarios

1. Choose ***File → Import → Scenarios***
2. The **Import Scenarios File List** dialog is displayed.
3. ***Select*** the scenario file to load.



Import Scenarios

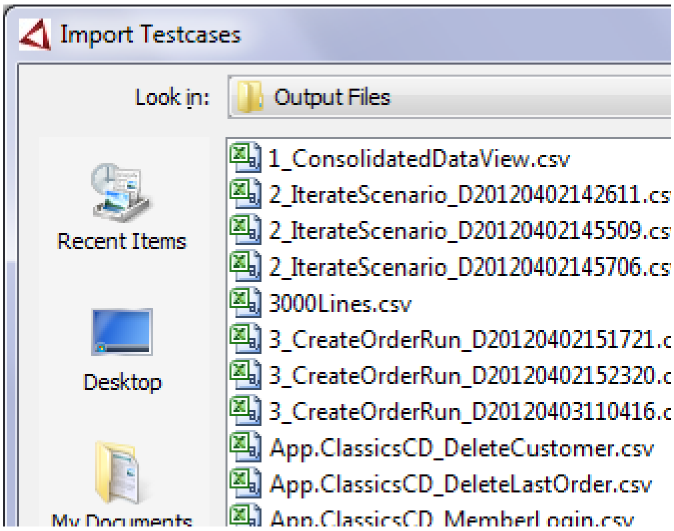
Look in: Output Files

Recent Items
Desktop
My Documents
1_ConsolidatedDataView.csv
2_IterateScenario_D20120402142611.csv
2_IterateScenario_D20120402145509.csv
2_IterateScenario_D20120402145706.csv
3000Lines.csv
3_CreateOrderRun_D20120402151721.csv
3_CreateOrderRun_D20120402152320.csv
3_CreateOrderRun_D20120403110416.csv
App.ClassicsCD_DeleteCustomer.csv
App.ClassicsCD_DeleteLastOrder.csv
App.ClassicsCD_MemberLogin.csv
App.ClassicsCD_Order.csv
Ap
Cl
Cl
Cl
Cl
Cl
Cl
Cl
Cl
Cl
Co
Co

4. Click ***Open***
5. Refer to the Status Bar for successful completion of the import

15.2.1.3 Importing Test Cases

1. Choose ***File → Import → Test Cases***
2. The **Import Test Cases File List** dialog is displayed
3. ***Select*** the test case names file to load



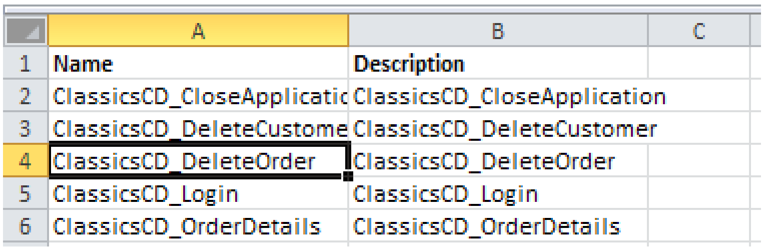
1_ConsolidatedDataView.csv
2_IterateScenario_D20120402142611.cs
2_IterateScenario_D20120402145509.cs
2_IterateScenario_D20120402145706.cs
3000Lines.csv
3_CreateOrderRun_D20120402151721.c
3_CreateOrderRun_D20120402152320.c
3_CreateOrderRun_D20120403110416.c
App.ClassicsCD_DeleteCustomer.csv
App.ClassicsCD_DeleteLastOrder.csv
App.ClassicsCD_MemberLogin.csv

4. Click ***Open***
5. Refer to the Status Bar for successful completion of the import

15.2.1.4 Export Functions

The **Export** functions allow Scenario and Test Case data to be output to .csv file for sharing across users and environments. The preferred method for transferring data between users and environments is using the Execution Suite import function.

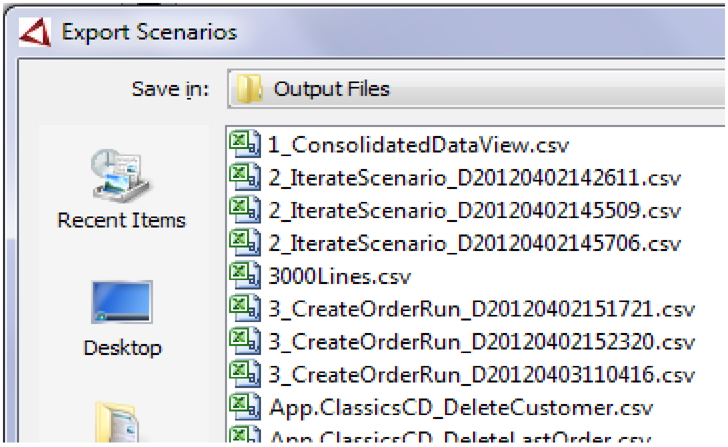
The exported **.csv** file contains Name and Description columns as shown below via Excel. This is the same format for both Scenario and Test Case imports.



	A	B	C
1	**Name**	**Description**	
2	ClassicsCD_CloseApplicatic	ClassicsCD_CloseApplication	
3	ClassicsCD_DeleteCustome	ClassicsCD_DeleteCustomer	
4	ClassicsCD_DeleteOrder	ClassicsCD_DeleteOrder	
5	ClassicsCD_Login	ClassicsCD_Login	
6	ClassicsCD_OrderDetails	ClassicsCD_OrderDetails	

15.2.1.5 Exporting Scenarios

1. Choose ***File → Export → Scenarios***
2. The **Export Scenarios File List** is displayed (as for Import)
3. ***Enter*** a filename or S***elect*** a file to overwrite



Export Scenarios

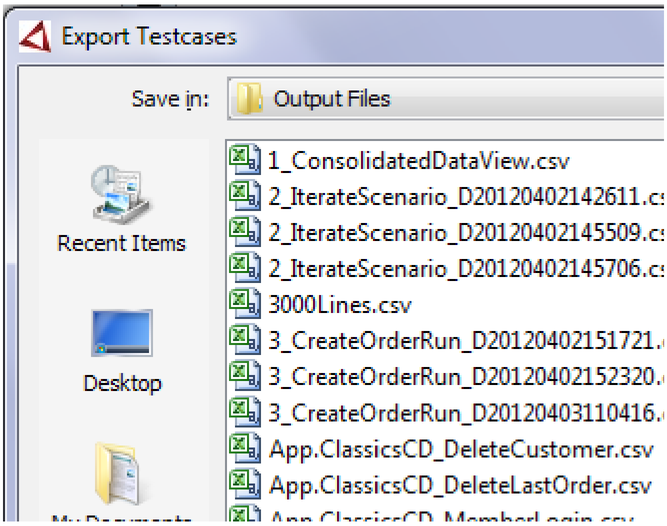
Save in: Output Files

1_ConsolidatedDataView.csv
2_IterateScenario_D20120402142611.csv
2_IterateScenario_D20120402145509.csv
2_IterateScenario_D20120402145706.csv
3000Lines.csv
3_CreateOrderRun_D20120402151721.csv
3_CreateOrderRun_D20120402152320.csv
3_CreateOrderRun_D20120403110416.csv
App.ClassicsCD_DeleteCustomer.csv
App.ClassicsCD_DeleteLastOrder.csv

4. Click ***Save***
5. Refer to the Status Bar for successful completion of the import

15.2.1.6 Exporting Test Cases

1. Choose **File > Export >Test Cases**
2. The Export Test Cases File List is displayed (as for scenarios).
3. Enter a filename or select the a file to overwrite
4. Click ***Save***



Export Testcases

Save in: Output Files

1_ConsolidatedDataView.csv
2_IterateScenario_D20120402142611.c
2_IterateScenario_D20120402145509.c
2_IterateScenario_D20120402145706.c
3000Lines.csv
3_CreateOrderRun_D20120402151721.
3_CreateOrderRun_D20120402152320.
3_CreateOrderRun_D20120403110416.
App.ClassicsCD_DeleteCustomer.csv
App.ClassicsCD_DeleteLastOrder.csv
App.ClassicsCD_MemberLogin.csv

5. Refer to the Status Bar for successful completion of the import

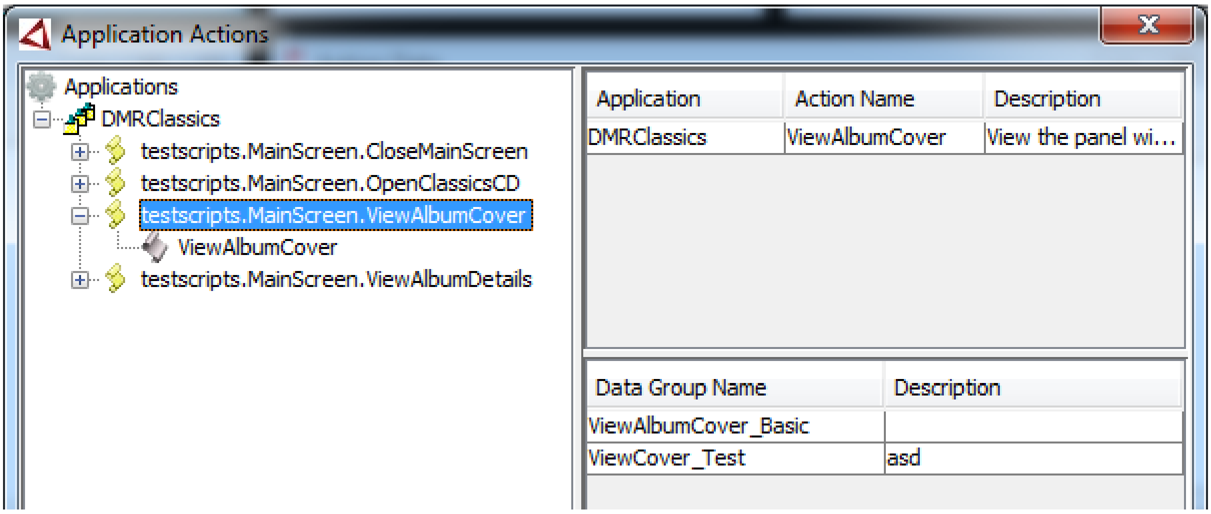
6. Click ***Save***
7. Refer to the Status Bar for successful completion of the import.

15.2.2 View Menu Functions in Detail

15.2.2.1 View Scripts

The View ➔Scripts function allows the TAF Pro user to view the available scripts and related action and Data Group details for each script within the environment.

1. Choose View ➔ Scripts
2. The Applications Actions Window shown below is displayed.
3. Click on the Applications node to expand it.


Application Actions
Applications
DMRClassics
testscripts.MainScreen.CloseMainScreen
testscripts.MainScreen.OpenClassicsCD
testscripts.MainScreen.ViewAlbumCover
ViewAlbumCover
testscripts.MainScreen.ViewAlbumDetails

Application	Action Name	Description
DMRClassics	ViewAlbumCover	View the panel wi...

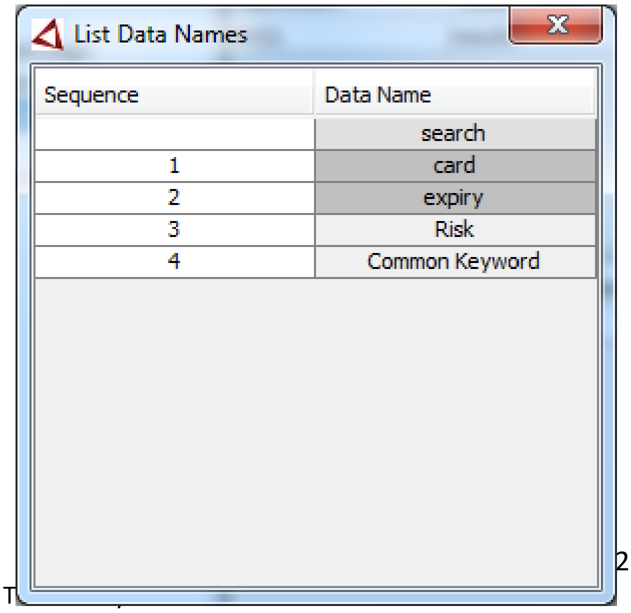
Data Group Name	Description
ViewAlbumCover_Basic	
ViewCover_Test	asd

4. Expand the tree to display test tool script paths and action names
5. Additional actions available:
   - Click on the script path to display the Action Data Groups.
   - Right click on the script path to show the Data names in the Data Group or create a Data Template
   - Right Click on the Data Group Name to see the Action Data or create a Data Template

6. View Script Data Names

At the Script level in the **Application Actions Window, *right click*** on a script, and choose ***Show Script Data Names.***

The following popup panel will display. Note the grey shading in the **Data Name** column.



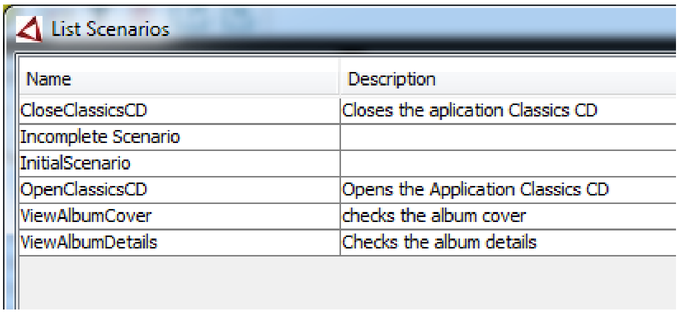
Sequence	Data Name
	search
1	card
2	expiry
3	Risk
4	Common Keyword

- Dark grey, e.g. card and expiry, indicates a datapool variable name used in a script
- Medium grey, e.g. search, indicates a precondition variable.
  - The values of these variables are passed between scripts during scenario execution.
  - Note also that precondition variables do not have sequences.
- Light grey, e.g. Risk and Common Keyword, indicates an additional variable. These fields are usually used for filtering

15.2.2.2 View Scenarios

View Scenarios displays all Scenarios existing in this users' current environment.

1. Select ***View* ➔ *Scenarios***
2. The **List *Scenarios* Window** will be displayed.
3. The Names and Descriptions of the scenarios are listed.

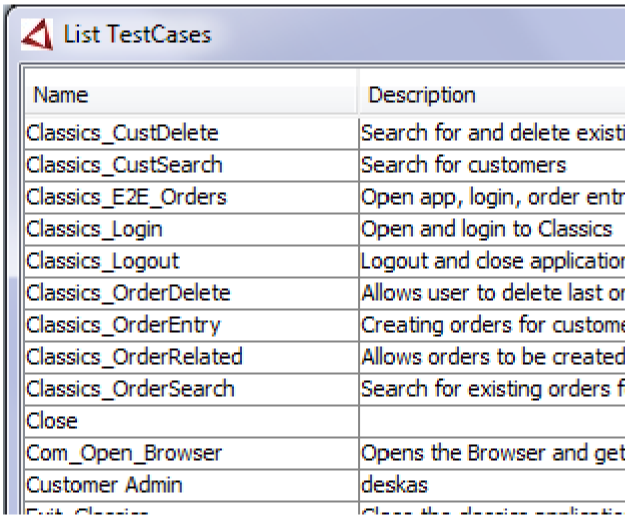


Name	Description
CloseClassicsCD	Closes the aplication Classics CD
Incomplete Scenario	
InitialScenario	
OpenClassicsCD	Opens the Application Classics CD
ViewAlbumCover	checks the album cover
ViewAlbumDetails	Checks the album details

15.2.2.3 View Test Cases

View Test Cases displays all Test Cases existing in the current user environment.

1. Select ***View* ➔ *Test Cases***
2. The ***List Test Cases Window*** will be displayed.

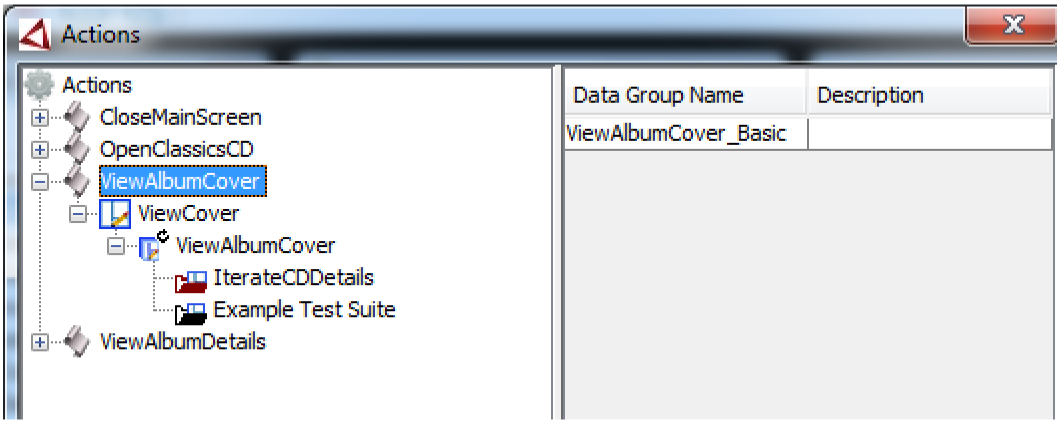


List TestCases

Name	Description
Classics_CustDelete	Search for and delete existi
Classics_CustSearch	Search for customers
Classics_E2E_Orders	Open app, login, order entr
Classics_Login	Open and login to Classics
Classics_Logout	Logout and close applicatior
Classics_OrderDelete	Allows user to delete last or
Classics_OrderEntry	Creating orders for custome
Classics_OrderRelated	Allows orders to be created
Classics_OrderSearch	Search for existing orders f
Close	
Com_Open_Browser	Opens the Browser and get
Customer Admin	deskas
Exit_Classics	Close the classics applicatio

15.2.2.4 View Actions

Selecting **Actions** will display the Actions panel.



Actions

- Actions
  - CloseMainScreen
  - OpenClassicsCD
  - ViewAlbumCover
    - ViewCover
      - ViewAlbumCover
        - IterateCDDetails
        - Example Test Suite
  - ViewAlbumDetails

Data Group Name	Description
ViewAlbumCover_Basic	

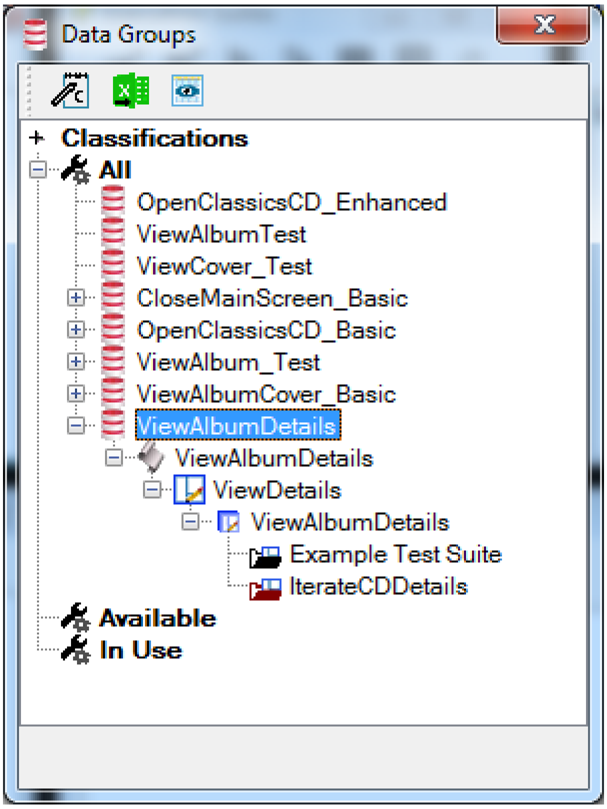
This display allows a TAF user to see which TestCases, Scenarios and Execution Suites use a selected action within an environment. Note that only actions that have been added to an execution suite will be shown.

When the Action is selected Data Groups used by the action are shown in the accompanying table over the page.

15.2.2.5 View Data Groups

Select **View** ➔ **Data Groups** to display the Data Groups panel. Data Groups are classified into groups **All**, **Available** and **In Use**. The display allows the user to identify which Actions, TestCases, Scenarios and Execution Suites use the data group.

Right clicking on a Data Group provides options to **Show Data** in the Action Data window or **Export** the usage structure to a file.



15.2.3 Data Menu Functions in Detail

TAF Pro stores data in two basic formats **Action Data** and **Execution Data**.

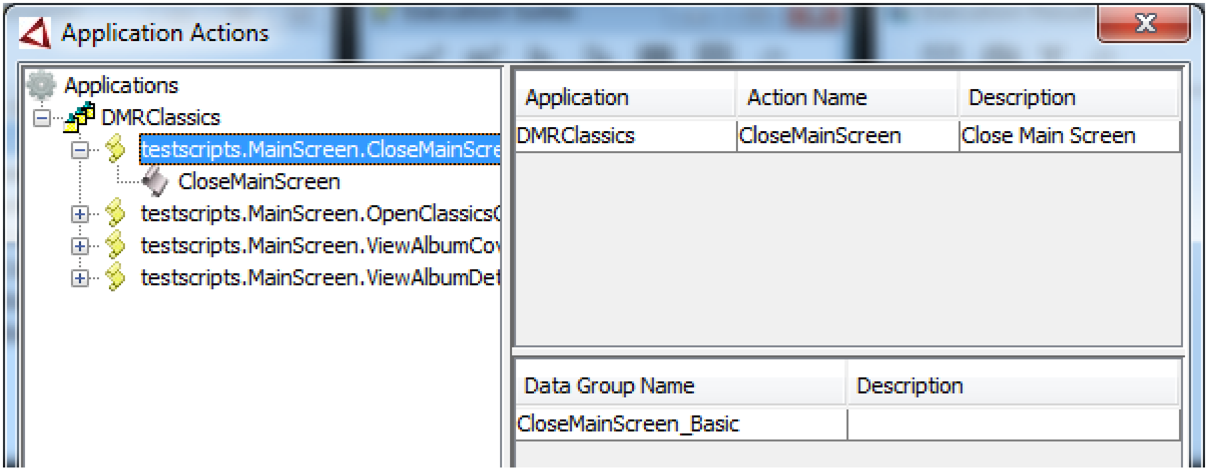
- An **Action** Data Group contains data relating to a single Action and is therefore typically named in relation to that Action with a suffix.
- An Execution Data Group contains data relating to all the Actions in an Execution Suite. It is there fore named in relation to the Execution Suite with a suffix.

TAF Pro provides the facility to create a Data Template for each data group that is then populated with data to be run in tests. Typically there will be several data groups relating to a particular Data Template.

- Data Templates should be stored in a separate subdirectory in the OutputFiles directory eg OutputFiles\DataTemplates
- Data Groups should be stored in a separate subdirectory in the OutputFiles directory eg OutputFiles\DataGroups.

15.2.3.1 Create Action Data Template

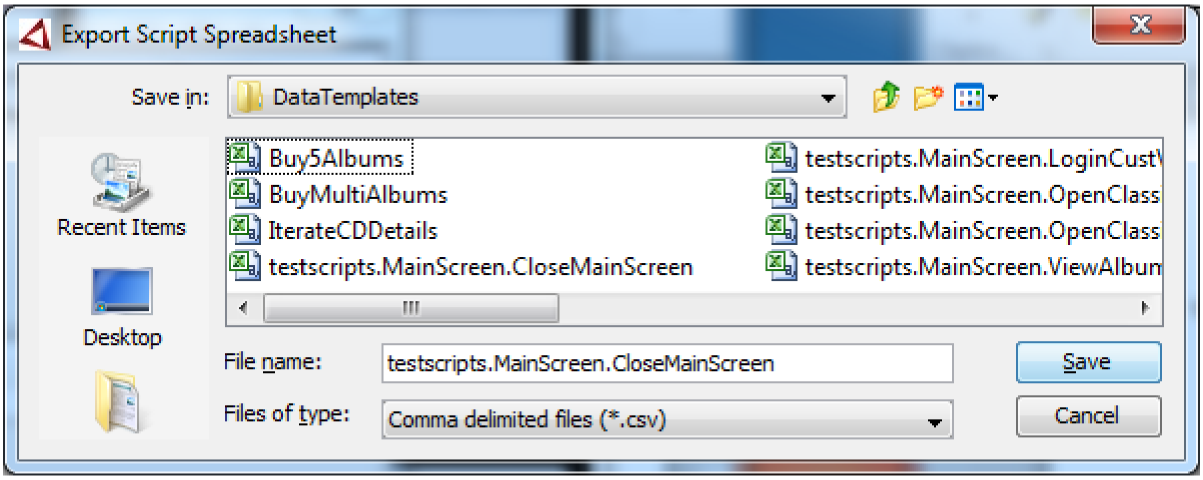
1. Select Data ➔ Create Data Template ➔ Create Action Data Template
2. The Application Actions window opens.


Application Actions
Applications
DMRClassics
testscripts.MainScreen.CloseMainScre
CloseMainScreen
testscripts.MainScreen.OpenClassics(
testscripts.MainScreen.ViewAlbumCov
testscripts.MainScreen.ViewAlbumDet

Application	Action Name	Description
DMRClassics	CloseMainScreen	Close Main Screen

Data Group Name	Description
CloseMainScreen_Basic	

3. Expand the Applications tree and select the script path containing the Action.
4. Right Click and select **Create Data Template.** The Export Script Spreadsheet opens.
5. Navigate to the folder you are using to store your Data Templates .


Export Script Spreadsheet
Recent Items
Desktop
Buy5Albums
BuyMultiAlbums
IterateCDDetails
testscripts.MainScreen.CloseMainScreen
testscripts.MainScreen.LoginCust\
testscripts.MainScreen.OpenClass
testscripts.MainScreen.OpenClass
testscripts.MainScreen.ViewAlbum
Save
Cancel

Save in: DataTemplates

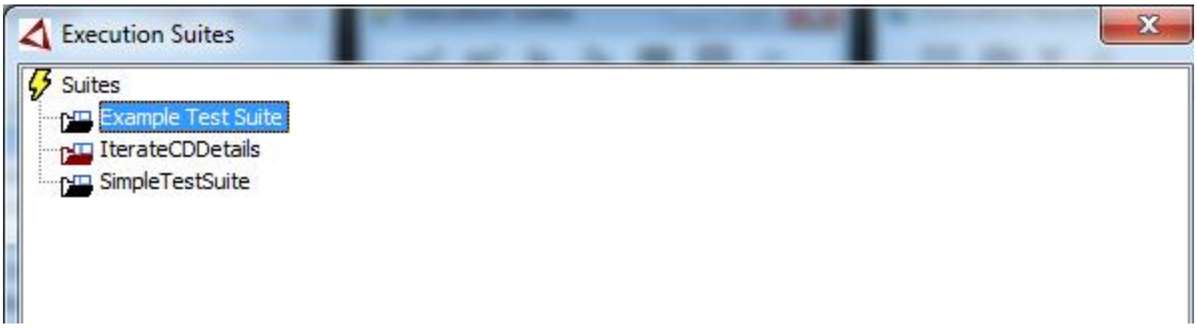
File name: testscripts.MainScreen.CloseMainScreen

Files of type: Comma delimited files (*.csv)

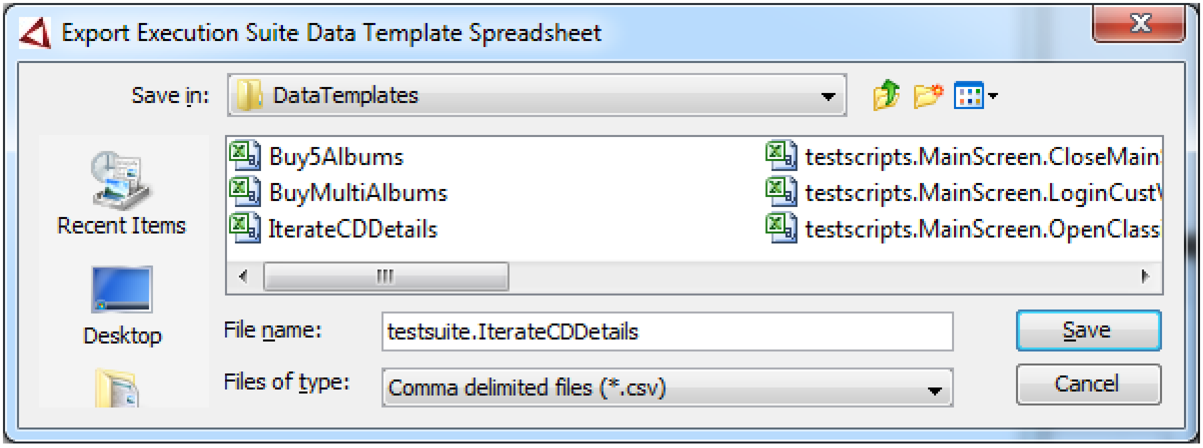
6. The recommended name for the Action template is the default name provided.
7. Click Save to create the template.

15.2.3.2 Create Execution Data Template

1. Select Data ➔ Create Data Template ➔ Create Execution Data Template
2. The Execution Suites window opens.



3. Select the Execution Suite you wish to create a template for.
4. Right click and select Create Execution Data Template



Export Execution Suite Data Template Spreadsheet

Save in: DataTemplates

Buy5Albums
BuyMultiAlbums
IterateCDDetails
testscripts.MainScreen.CloseMain
testscripts.MainScreen.LoginCust\
testscripts.MainScreen.OpenClass

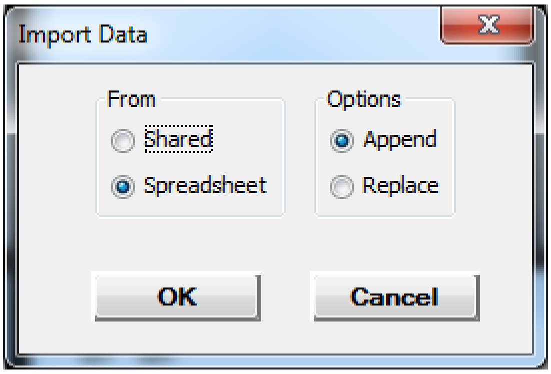
File name: testsuite.IterateCDDetails

Files of type: Comma delimited files (*.csv)

5. Navigate to the folder you are using to store your Data Templates .
6. You may wish to add a prefix to the default Template name to identify it as an Execution Data Template
7. Click Save

15.2.3.3 Import Action Data

1. Select Data ➔ Import ➔ Import Action Data
2. The Import Data panel is displayed.



Import Data

From
○ Shared
● Spreadsheet

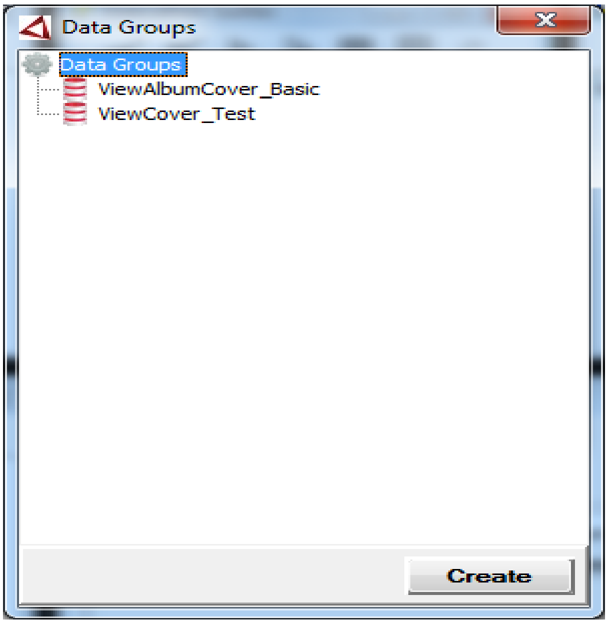
Options
● Append
○ Replace

Data can be imported from the TAF Pro shared environment or from a spreadsheet or .csv file.

On import, data can be Appended to a new or existing Data Group or the Replace option can be used to remove any existing data.

Default values can be altered in User ➔ Preferences ➔ Data

3. Select the Data File to be imported from the appropriate folder and click **Open**.
4. The Data Groups window opens.
5. Select the Data Group to be used or select **Create** to create a new Data Group.
6. If you select Create you will be required to provide a name and description of the new Data Group.
7. The Starting Import window will be displayed



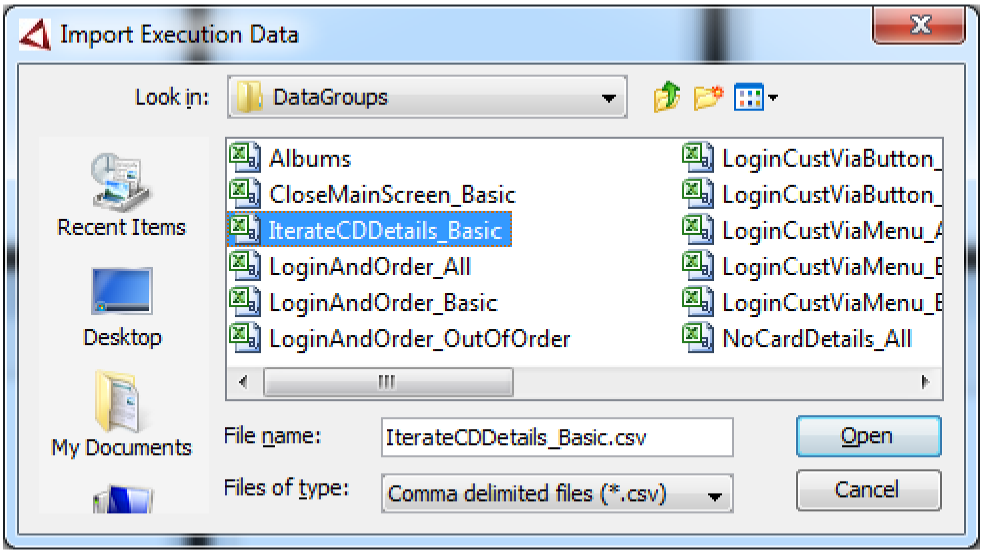
Data Groups

Data Groups
ViewAlbumCover_Basic
ViewCover_Test

8. When the import is displayed the Status bar will report that the data has been imported.

15.2.3.4 Import Execution Data

1. Select Data ➔ Import ➔ Import Execution Data.
2. The Import Execution Data window will open. Navigate to the execution Data.



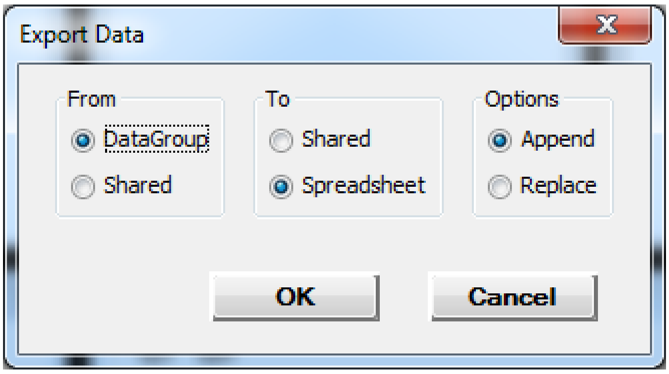
File name: IterateCDDetails_Basic.csv

Files of type: Comma delimited files (*.csv)

3. Select the Execution Data file and click Open
4. The Import Status window will display.
5. The Status bar will display a message indicating that import has been completed.

15.2.3.5 Export Action Data

1. Select Data ➔ Export ➔ Export Action Data
2. The Export Data panel will open



From: ● DataGroup ○ Shared

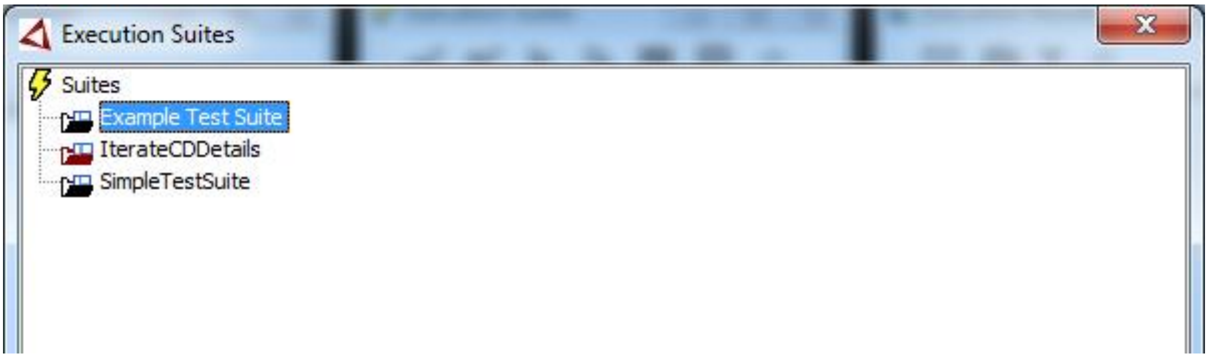
To: ○ Shared ● Spreadsheet

Options: ● Append ○ Replace

3. Choose the relevant options and click ok
4. The Data Groups panel will open, select a Data Group, right click and chose Select from the context menu.
   4.1. If exporting to the Shared location, a Popup indicting that Export is occurring will display, when completed the Status bar will show **Local Data exported to server**.
   4.2. If exporting to a spreadsheet the **Export Action Data** panel opens , navigate to your Data Groups folder and click **Save**.

15.2.3.6 Export Execution Data

1. Select Data → Export → Export Execution Data
2. The Execution Suites window will Open



3. Select the Execution Suite to be exported, right Click and select Export Execution Data from the context menu.
4. The Export Execution Data panel will display. Navigate to the appropriate folder and save the file.

15.2.4 Scripts Menu Functions in detail

15.2.4.1 Register Scripts

1. Select Scripts → Register.
2. The Register Scripts panel will open to the top folder of the datastore containing the scripts.
3. Navigate to the location where the script is stored.
4. Select one or more scripts by using clicking on a single script or using Ctrl or Shift key and arrow keys to select the scripts required.
5. Click Open to register the selected scripts.
6. On completion of the script registration process, a message appears on the Status bar indicating that script registration was successful.
7. The script and all the Actions it contains are now registered in the Framework.

15.2.4.2 Convert Script or Project

Script or project conversion occurs when the script is first included in the TAF Pro project. It is the responsibility of the Scripting Team and the process is described in the Scripting Guide.

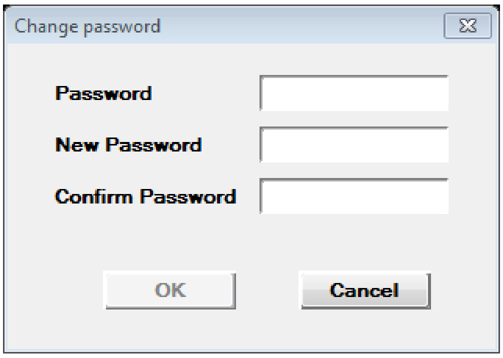
15.2.5 User Menu Functions in detail

15.2.5.1 Change Password

When a user is set up for TAF Pro, the password is generally get to a default password that is the same for all users. When the user first accesses TAF Pro, they should change their password in line with security policies for the company.

To change the password for the logged in user:

1. Select ***User*** **→** ***Change Password*** form the Menu Bar
2. The Change Password window is displayed below



3. ***Enter*** the old or ***existing password***
4. ***Enter*** the ***new password***
5. ***Re-enter*** the ***new password***
6. Click ***OK*** to save changes

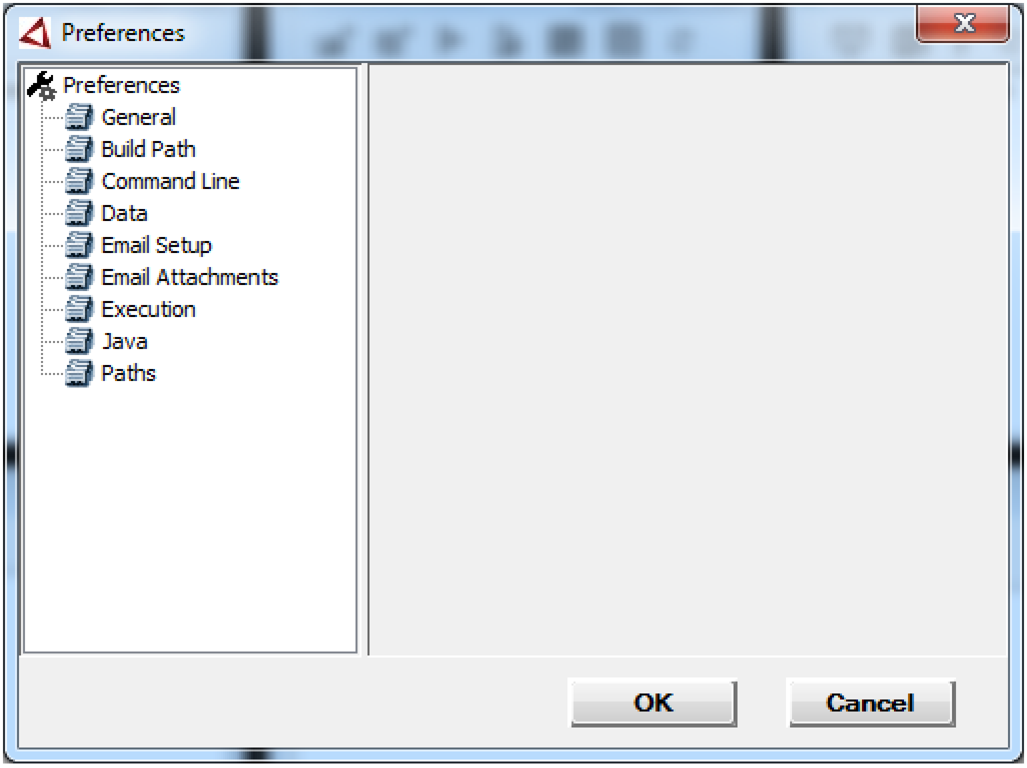
The new password will apply from the next login.

15.2.5.2 Preferences

User preferences are local settings for each user. These settings are applied to all environments that the user is assigned too.

Selects User > Preferences to display the preferences window.

The User Preferences window is displayed.



Open each group of preferences by double clicking on the relevant node.

15.2.5.3 General

The following General Preferences are available:

- **Maximum Rows to Execute** – Allows the user to change the default maximum number of rows executed when running a test. This setting ensures that, in the design phase Test Execution is limited to a few lines rather than executing the entire Data Group. This value can be overridden at test execution time.
- **Maximum Action Data - Auto Display** – Sets the number of rows that will display automatically when the action or data group is selected from the Execution Suites panel. When there are large data groups, this setting helps reduce delays loading data. To display all data in the data group the use can right click on the action or data group and choose Show Action Data or Show Data
- **Data Import Progress Indicator** – When importing data a new message indicating progress is displayed every time this number of rows is imported.
- **Show Windows at Startup -** The default is to show all TAF Pro windows on startup. If this option is deselected no windows are active when TAF Pro opens.
- **Defaults -** Reset the user preference general options to factory settings.

Click Apply to save changes made to General preferences

15.2.5.4 Build Path

The Build Path preference panel allows the user to add extra libraries for use by IBM Rational Functional Tester Scripts. TAF Pro and IBM Rational Functional Tester have a standard set of libraries that are used. This facility allows the user to include either custom jar files or import 3rd party jars to achieve specific functions within the script. Adding the jar files here will make them available to TAF Pro. Any 3rd party or custom built jar files also need to be imported to IBM Rational Functional Tester to complete the usage capability.

1. To add custom built or 3rd party jars to the Build Path Preferences:
   1.1. Click on Build Path in the preferences window
   1.2. Right Click in the Build Path panel and select New
   1.3. Navigate to the location of the required jar file
   1.4. Select the file and Click Open
   1.5. The new file is displayed in the build path panel.
2. To remove redundant or unneeded jars select the file to highlight
   2.1. Right click on the jar file and select Remove

15.2.5.5 Command Line

There are three subsections to Command Line preferences

Name: Set the default name and description for the command line build file. These fields must be added to when the CLI build file is created so the recommended values are to use a standard suffix in the name eg **'_Build'** and an a stand introduction in the Description eg **'Build Description: '**

Paths: Set the default location for the Test Plans and Results eg {TAF ProjectPath}\CLI\Plans, {TAF ProjectPath}\CLI\Results.

Options: Select the Report Format and the Report Category. The only available Report Format currently is **csv**. The report Categories are **All, Pass, Fail or Warning**.

15.2.5.6 Data

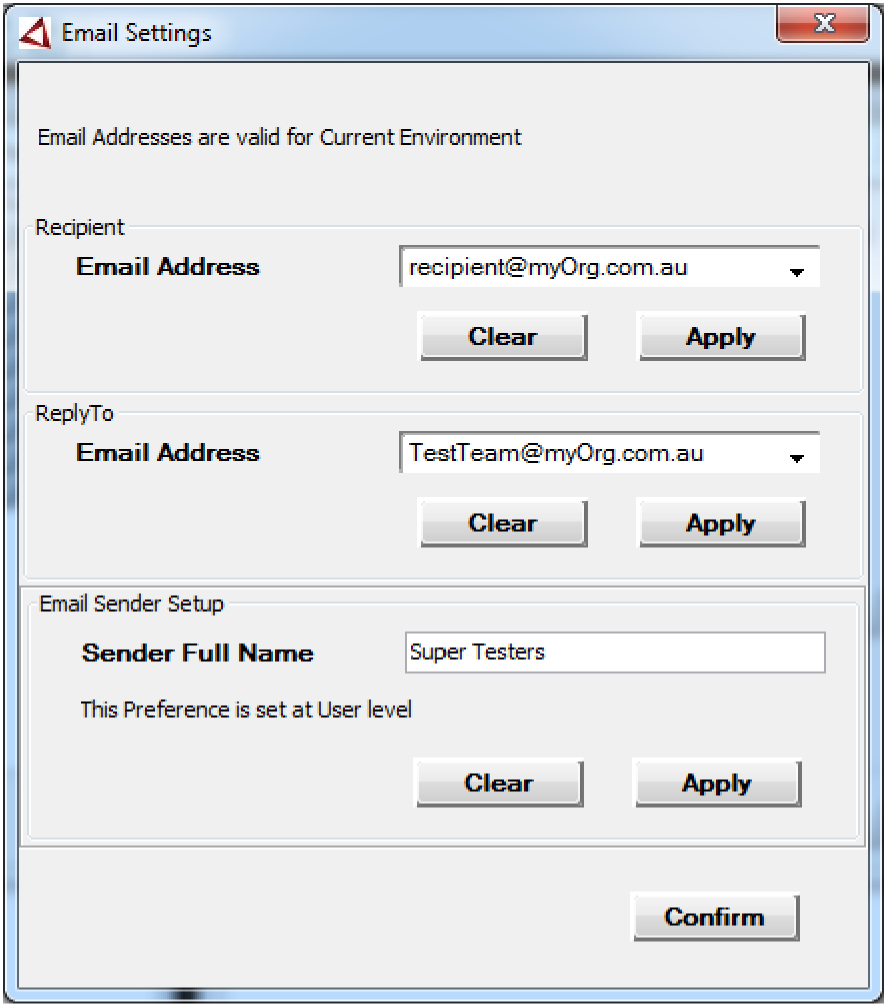
Two options are available for default processing of data import and export.

Append or Replace: determines whether existing data is added to overwritten.

File Format: 'csv' or 'xls'.

15.2.5.7 Email Setup

Two email addresses can be setup for the current user environment as well as providing the users full name.



Email Settings

Email Addresses are valid for Current Environment

Recipient

Email Address: recipient@myOrg.com.au

ReplyTo

Email Address: TestTeam@myOrg.com.au

Email Sender Setup

Sender Full Name: Super Testers

This Preference is set at User level

These emails addresses are used on all emails generated by the user in the current environment unless overridden when used. Email addresses can be entered directly or selected from a drop down list of previously used email addresses. Click **Apply** to save each address.

The **Clear** button removes all previously used email addresses. The **Confirm** button finalises the changes.

15.2.5.8 Email Attachments

Sets a maximum size (in megabytes) for email attachments sent from this environment.

15.2.5.9 Execution

Determines whether the Execution toolbar opens or is minimised when execution is started.

15.2.5.10 Java

Set the path to the java.exe file used in TAF Pro. This allows the user to select the Java version used.

15.2.5.11 Paths

Displays the paths used in the configuration of this environment. These can only be modified when logging in however they can be checked here to ensure they are correct.

15.2.6 Help Menu Function in Detail

15.2.6.1 Documentation

There are four selections provided to open the User Guide, Scripter Guide, Installation & Configuration Guide or Release Notes in a PDF document.

15.2.6.2 Log an Issue

Opens a panel allowing the use to report an issue or request support information from TestPro. Email must be setup for this facility to operate.

15.2.6.3 About

Describes Version Number, Licence, Patent and component information relating to TAF Pro.

15.3 Appendix C – HP VAPI-XP Test Script

The following script code should be copied and pasted into the “Test Script” in HP Quality Center to replace the script that is generated.

```
' TAFPro VAPI-XP Test Scrip v0.1
'
' ----------------------------------------------------
Sub Test_Main(Debug, CurrentTestSet, CurrentTSTest, CurrentRun)
 ' *** VBScript Limitation ! ***
 ' "On Error Resume Next" statement suppresses run-time script errors.
 ' To handle run-time error in a right way, you need to put "If
Err.Number <> 0 Then"
 ' after each line of code that can cause such a run-time error.
 On Error Resume Next
 ' clear output window
 TDOutput.Clear
 vBuildFailed = 0
 If Not Debug Then
  ' Load the local variables with the QC parameters
  With CurrentTSTest.Params
   For i = 0 To .Count - 1
    If StrComp(Trim(.ParamName(i)), "ReportPath", 1) = 0 Then
     vLogPath = .ParamValue(i)
    End if
    If StrComp(Trim(.ParamName(i)), "Executable", 1) = 0 Then
     vExecutable = .ParamValue(i)
    End if
    If StrComp(Trim(.ParamName(i)), "TAFLOG", 1) = 0 Then
     vTafLog = .ParamValue(i)
    End if
   Next
  End With
 End If
 'Run the TAFPro batch file
 vCheckBatchFile = CheckFileExists(vExecutable)
 If vCheckBatchFile = True Then
  XTools.run vExecutable, "", -1
 Else
  Err.Number = 1
  Err.Description = "Batch File not found, check the executable
parameter."
  vBuildFailed = 1
 End If
 ' Check the output text ANT Build failure
 vText = TDOutput.Text
 MyPos = Instr(4, vText, "BUILD FAILED", 1)
 if MyPos <> 0 Then
  Err.Number = 1
  Err.Description = "ANT Build process failed"
  vBuildFailed = 1
 End if
 ' Get the latest log file from the log path
 If vBuildFailed = 0 Then
  Dim fNewest
  set
oFolder=createobject("scripting.filesystemobject").getfolder(vLogPath)
```

```
  For Each aFile In oFolder.Files
   If fNewest = "" Then
     Set fNewest = aFile
   Else
     If fNewest.DateCreated < aFile.DateCreated Then
       Set fNewest = aFile
     End If
   End If
  Next
 ' Check the log file for any failures (ignore first line)
 Set fso = CreateObject("Scripting.FileSystemObject")
 vRunLog = vLogPath & "\" & fNewest.Name
 vFailed = 0
 Set fLogFile = fso.OpenTextFile(vRunLog, 1)
 vLineIn = fLogFile.ReadLine ' Read First line
 Do While (fLogFile.AtEndOfStream <> True)
   vLineIn = fLogFile.ReadLine
   'TDOutput.Print(vLineIn)
   vTextPosition = Instr(1, vLineIn, "Fail", 1)
   If vTextPosition <> 0 Then
    vFailed = 1
   End If
 Loop
 fLogFile.Close
  ' Check if Test failed and set error message
  If vFailed <> 0 Then
   Err.Number = 1
   Err.Description = "Test Failed, check test log for details"
  End If
  ' Upload the TAFPro command line log file to the execution run
  TDHelper.UploadAttachment vRunLog, CurrentRun
  ' Upload the RFT Log file
  Set fs = CreateObject("Scripting.FileSystemObject")
  Set MainFolder = fs.GetFolder(vTafLog)
  For Each fldr In MainFolder.SubFolders
    If fldr.DateLastModified > LastDate Or IsNull(LastDate) Then
      LastFolder = fldr.Name
      LastDate = fldr.DateLastModified
    End If
  Next
  'MsgBox LastFolder
  vRFTLogFile = vTafLog & "\" & LastFolder & "\rational_ft_log.html"
  TDHelper.UploadAttachment vRFTLogFile, CurrentRun
 End If ' Build Fail If
 If Not Debug Then
 End If
 ' handle run-time errors
 If Err.Number <> 0 Then
  TDOutput.Print "Run-time error [" & Err.Number & "] : " &
Err.Description & Erl
  ' update execution status in "Test" mode
  If Not Debug Then
   CurrentRun.Status = "Failed"
   CurrentTSTest.Status = "Failed"
  End If
 End If
 ' Write TDOutput to file and attach
 Const ForAppending = 8
 Set objFSO = CreateObject("Scripting.FileSystemObject")
 Set objTextFile = objFSO.OpenTextFile("c:\tdoutput.txt",
ForAppending, True)
 objTextFile.WriteLine(TDOutput.Text)
 objTextFile.Close
 wait 1
 TDHelper.UploadAttachment "c:\tdoutput.txt", CurrentRun
 objFSO.Deletefile("c:\tdoutput.txt")
End Sub
Function CheckFileExists(filespec)
 Dim fso, msg
 Set fso = CreateObject("Scripting.FileSystemObject")
 If (fso.FileExists(filespec)) Then
  CheckFileExists = true
 Else
 CheckFileExists = false
 End If
End Function
```