User's Guide

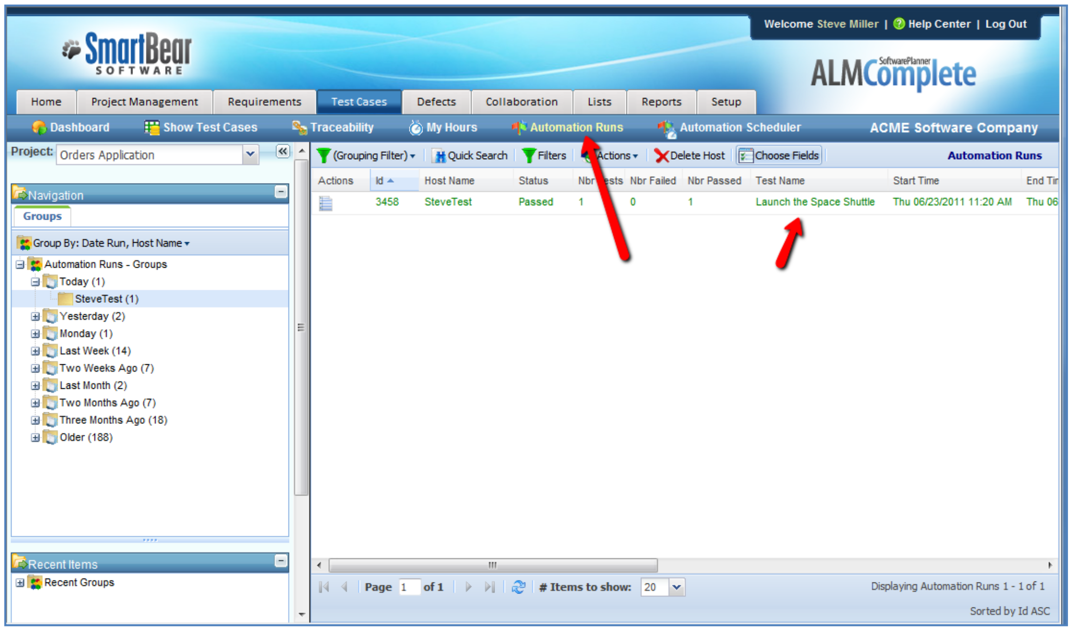
Software Planner
Posting Automation Runs from Custom Test Harnesses

This document provides an overview of how to post Automation Runs from a Custom Test Harness.

Contents

Overview

Companies that wish to post runs from custom test harnesses can use our API to do that. This User's Guide describes how to do that. Once automated tests are posted into ALMComplete or QAComplete, you can query the run history from our user interface (click Test Cases | Automation Runs):



Project: Orders Application

Group By: Date Run, Host Name

- Automation Runs - Groups
  - Today (1)
    - SteveTest (1)
  - Yesterday (2)
  - Monday (1)
  - Last Week (14)
  - Two Weeks Ago (7)
  - Last Month (2)
  - Two Months Ago (7)
  - Three Months Ago (18)
  - Older (188)

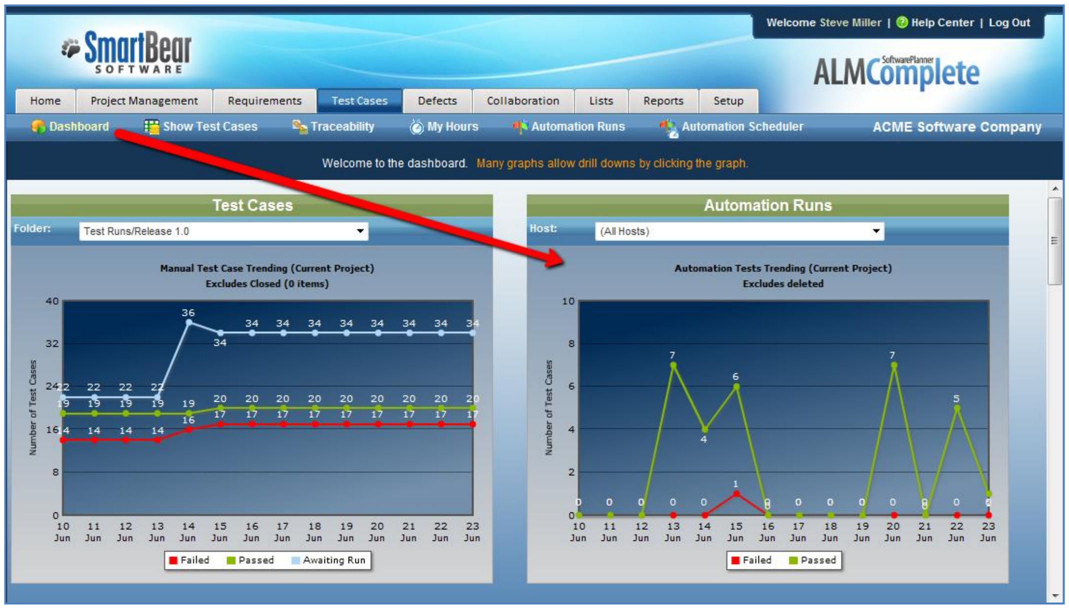
Automation Runs

Actions	Id	Host Name	Status	Nbr Tests	Nbr Failed	Nbr Passed	Test Name	Start Time	End Ti
	3458	SteveTest	Passed	1	0	1	Launch the Space Shuttle	Thu 06/23/2011 11:20 AM	Thu 06

Page 1 of 1 # Items to show: 20 Displaying Automation Runs 1 - 1 of 1

Sorted by Id ASC

You can also view test runs via a dashboard:



Welcome to the dashboard. Many graphs allow drill downs by clicking the graph.

Folder: Test Runs/Release 1.0

Manual Test Case Trending (Current Project)
Excludes Closed (0 items)

Host: (All Hosts)

Automation Tests Trending (Current Project)
Excludes deleted

Pre-Requisites

Before beginning , you need collect a couple pieces of information:

- **DeptId** – You can find this by going to **Setup | Project (Open & Create)**
- **ProjId** – You can find this by going to **Setup | Project (Open & Create)**
- **UserId** – You can setup a new user for the integration or use an existing one. Determine the userid by going to **Setup | Security | Users**

Want to see how to do this? Watch this video:
http://www.softwareplanner.com/movies.asp?Topic=CTAPrereq

Object Overview

To connect test runs for a custom test harness, you will be working with 3 basic objects:

1. **AutomationHosts** – This identifies the machine (or host) you are running the tests on. If you are always running the hosts on a single machine, you will need only one Automation Host. If you are running on multiple machines, you will need an AutomationHost for each machine.
2. **AutomationTestItems** – This is a group of automated test case you wish to run, for example you may have an AutomationTestItem for Regression Tests, another for Smoke Tests, etc. You can even create an individual AutomationTestItem for each granular level automated test case, if you wish to report on it at that level.
3. **AutomationRuns** – This summarizes your run, it links back to an AutomationHost and AutomationTestItem and summarizes how many passed, failed, and how long the run took.

Using the API

Our API allows you to add and query each of the objects discussed above. To hook up a harness you will normally go through this process:

Add your AutomationHost

Add your AutomationHost using our API (**AutomationHosts_Add**). If you are always going to be running tests on the same machine, you only need to do this once. If you are running on different machines each time, you will do add a host when you need to. Let's imagine your machine name is **vmPerl** and you will run your tests on that machine. To add that, invoke the AutomationHosts_Add method passing in these parameters (along with your security authentication info):

- **Host Name** – vmPerl
- **Description** – (Optional) Can be anything (e.g. This machine is where we run our Perl automated scripts)

- **CreateUserId and UpdateUserId** – Use the UserId of the account created for running these scripts (discussed in Pre-Requisites section above).
- **DateCreated and DateUpdated** – These should auto populate.

Getting the AutomationHostId

Once you have added an AutomationHost, you will need the ID of that to be used in the other API methods, so you will invoke the **AutomationHosts_Load** method to retrieve it. You will send it your security authentication info along with your ProjId in the **<ProjIds>** parameter (we discussed the ProjId in the Pre-Requisites section above. Once invoked, this will return a list of Automation Hosts (with the HostName and AutomationHostId of each), so you will capture the AutomationHostId of the one you just added).

Add the AutomationTestItems

Once you have your AutomationHostId, you will want to add an AutomationTestItem for the group of test cases you are running (or you could add one for each individual test case if you want to get really granular). Let's imagine you wanted to create a Test Item that represented all the test cases in your Regression Suite (100 test cases for example) and you only wanted to report on that from a summary level. Use the **AutomationTestItems_Add** method to add this. You will send it your security authentication info and fill in these parameters:

- **AutomationHostId** – Discussed above, represents the machine it was run on
- **AutomationHostName** - Discussed above, represents the machine it was run on (vmPerl)
- **TestName** – This would be the name of the test (e.g. Full Regression)
- **TestType** – Type of test (e.g. Perl Script)
- **FullPath** – (Optional) Where the automated script resides (e.g. \\MyTestPath\FullRegression.perl)
- **AutomationTestParentId** – Set to 0 unless you want this to be a child of other tests, then set it to the AutomationTestItemId of the parent test.
- **ParentFilePath** – (Optional) Leave it blank
- **Description** – Description of the test (e.g. Runs a full regression for XYX system)
- **AutomationType** – Can be anything (e.g. Perl)
- **ProjId** – ProjId discussed in Pre-Requisites section
- **CreateUserId and UpdateUserId** – Use the UserId of the account created for running these scripts (discussed in Pre-Requisites section above).
- **DateCreated and DateUpdated** – These should auto populate.

Getting the AutomationTestItemId

Once you have added an AutomationTestItem, you will need the ID of that to be used in the other API methods, so you will invoke the **AutomationTestItems_Load** method to retrieve it. You will send it your security authentication info along with your ProjId in the **<ProjIds>** parameter (we discussed the ProjId in the Pre-Requisites section above. Once invoked, this will return a list of

AutomationTestItems (with the TestName and AutomationTestItemId of each), so you will capture the AutomationTestItemId of the one you just added).

Add the AutomationRuns

Once you have your AutomationHostId and AutomationTestItemId, you will want to add an AutomationTestRun each time you run an AutomationTestItem, so you can run the same AutomationTestItem many times and each time will be logged to a different AutomationRun. This summarizes your run, it links back to an AutomationHost and AutomationTestItem and summarizes how many passed, failed, and how long the run took.

Let's imagine you wanted to run an existing AutomatedTestItem, you would invoke the **AutomationTestRuns_Add** method to add this. You will send it your security authentication info and fill in these parameters:

- **AutomationHostId** – Discussed above, represents the machine it was run on
- **AutomationHostName** - Discussed above, represents the machine it was run on (vmPerl)
- **AutomationTestItemId** – Discussed above, represents the Automation test item.
- **Status** – Normally you would set this to Passed or Failed, depending on the status of the run.
- **FullTestName** – Set this to the same name as the AutomatedTestItem
- **NbrTests** – The number of tests in this run (e.g. 100)
- **NbrPassed** – The number of tests that passed in this run (e.g. 95)
- **NbrFailed** – The number of tests that failed in this run (e.g. 5)
- **DurationInSecs** – How many seconds this test took to run (eg. 120 would be 2 minutes)
- **FileId** – If you wanted to attach a file to this run (like a Run Report), you can invoke our Attachments_Add method to add the file and the Attachments_GetAttachmentsList method to retrieve the FileId of the item uploaded. Then you can use this FileId here. If you don't care about that, set it to 0.
- **StartTime** – The date/time the test began
- **EndTime** – The date/time the test ended
- **AutomationType** – Can be anything (e.g. Perl)
- **CreateUserId and UpdateUserId** – Use the UserId of the account created for running these scripts (discussed in Pre-Requisites section above).
- **DateCreated and DateUpdated** – These should auto populate.

Seeing it in Action

Let's imagine that you wanted to record the run results of a test called **Full Regression** on a host named **vmPerl**. Let's imagine the automation run was a Perl script we built. Watch this movie to see how the tables in the database are affected and how you can view the results inside of ALMComplete or QAComplete:

http://www.softwareplanner.com/Movies.asp?Topic=CTADemo