

Rational. Rhapsody



IBM® Rational® Rhapsody® TestConductor Add On



Adapter for Rational Quality Manager - Howto

Rhapsody®

**IBM® Rational® Rhapsody®
TestConductor Add On**

Adapter for Rational Quality Manager - Howto

Release 2.4.3



License Agreement

No part of this publication may be reproduced, transmitted, stored in a retrieval system, nor translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the prior written permission of the copyright owner, BTC Embedded Systems AG.

The information in this publication is subject to change without notice, and BTC Embedded Systems AG assumes no responsibility for any errors which may appear herein. No warranties, either expressed or implied, are made regarding Rhapsody software and its fitness for any particular purpose.

Trademarks

IBM® Rational® Rhapsody®, IBM® Rational® Rhapsody® Automatic Test Generation Add On, and IBM® Rational® Rhapsody® TestConductor Add On are registered trademarks of IBM Corporation.

All other product or company names mentioned herein may be trademarks or registered trademarks of their respective owners.

© Copyright 2000-2010 BTC Embedded Systems AG. All rights reserved.

1. Introduction

This document gives a short overview about the Rhapsody TestConductor Adapter for Rational Quality Manager. It describes (shortly) how to

- register the TestConductor adapter user interface plugin in RQM (2.1),
- configure and start the adapter (2.2),
- create TestConductor Tests in Rhapsody (3.1),
- create TestCases (using the TestConductor Adapter) in RQM (3.2) and
- execute and evaluate these tests (4).

This document concludes with a troubleshooting and FAQ section (5).

2. Preparing the Adapter UI and the Adapter

The Rhapsody TestConductor adapter for RQM consists of two parts. The first is a user interface plugin which has to be registered in RQM on the RQM server machine (see 2.1). The second is the Rhapsody TestConductor adapter itself which has to be started on the test machine on which the Rhapsody TestConductor tests are to be executed (see 2.2).

2.1 Registering the UI Plugin

Since the Rhapsody TestConductor adapter is currently not part of the standard RQM installation, the adapter has to be registered with RQM manually. To do this the following tasks are required:

1. Copy the file
“<RhapsodyInstallPath>/TestConductor/RQM_TestConductorAdapter/btc-profile.ini” into the folder
“<RQMInstallPath>/server/conf/jazz/provision_profiles”.

(Ensure that the path specified in the file directs to the folder
“<RhapsodyInstallPath>/TestConductor/RQM_TestConductorAdapter”
2. If RQM is running, shut RQM down.
3. Delete the content of the folder “<RQMInstallPath>/server/tomcat/work/”.
4. Restart the RQM Server

After that an additional entry “Rhapsody TestConductor” should be available in the type section of the Test Script editor dialog.

2.2 Configuring and Starting the RQM TestConductor Adapter

The TestConductor Adapter is located inside the folder

TestConductor/RQM_TestConductorAdapter in your Rhapsody installation.

The adapter is started via the file `start.bat`¹ analogous to the CommandLine Adapter shipped with RQM²:

```
start.bat -repository https://<rqmserver>:<port>/jazz -user <userid>
-passwd <password> -adapter <adapter> \
    [-projectArea <project area>] [-sleepTime <sleep time>] \
    [-configFile <configuration file>]
```

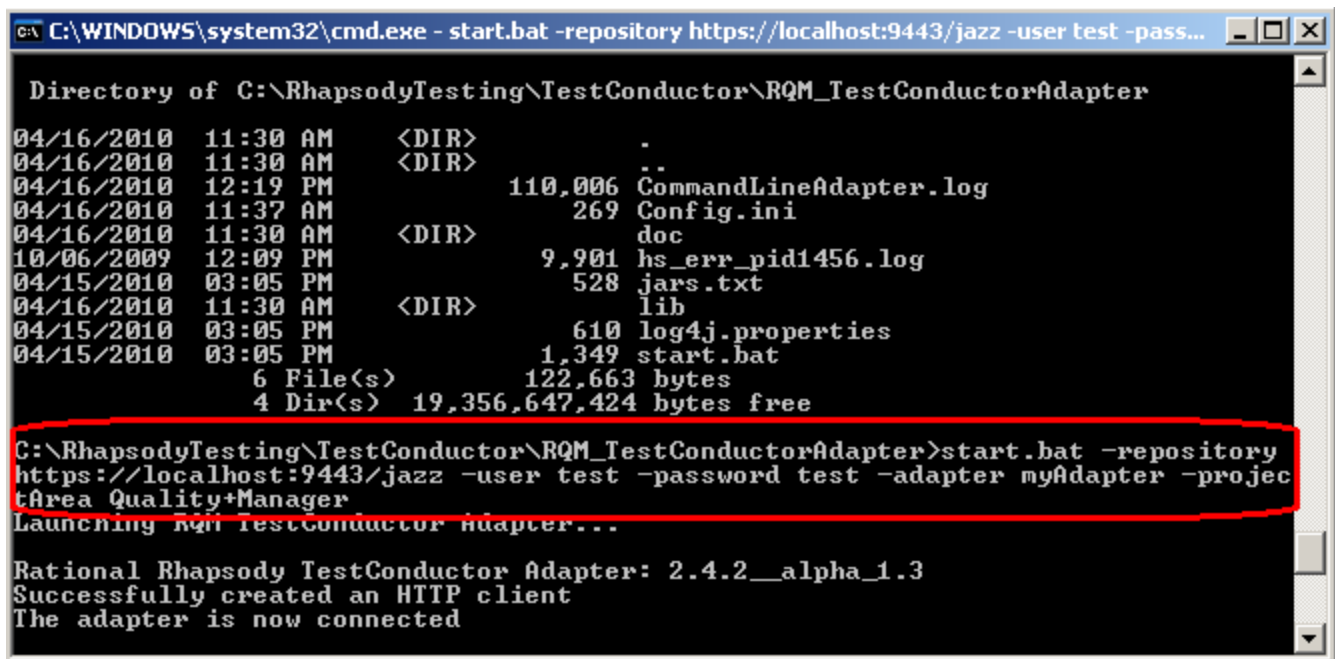
¹ Currently there is no official Linux support of the TestConductor Adapter.

² The following parameter description is extracted from the readme.txt of the commandline adapter.

where:

rqmserver: is the hostname/IP address of the RQM server
port: is the port where the RQM server is running
userid: is a registered user ID within RQM that has the license to run an adapter
password: is the password of the userID used
adapter: is a user given adapter name
project area: is the name of the project being logged into, default is "Quality+Manager"
sleep time: is the polling interval between polling for tasks, default is 30 seconds
log file: the file to which to log the output messages, default is CommandLineAdapter.log
configuration file: file to store and read the settings for this adapter, default is Config.ini

The adapter will generate a configuration file (Config.ini file or the file specified via -configFile option) that will contain the registration information. This will be re-used when the adapter is re-started.



```
C:\WINDOWS\system32\cmd.exe - start.bat -repository https://localhost:9443/jazz -user test -pass...

Directory of C:\RhapsodyTesting\TestConductor\RQM_TestConductorAdapter

04/16/2010  11:30 AM    <DIR>          .
04/16/2010  11:30 AM    <DIR>          ..
04/16/2010  12:19 PM             110,006  CommandLineAdapter.log
04/16/2010  11:37 AM             269    Config.ini
04/16/2010  11:30 AM    <DIR>          doc
10/06/2009  12:09 PM             9,901  hs_err_pid1456.log
04/15/2010  03:05 PM             528    jars.txt
04/16/2010  11:30 AM    <DIR>          lib
04/15/2010  03:05 PM             610  log4j.properties
04/15/2010  03:05 PM             1,349  start.bat
        6 File(s)              122,663 bytes
        4 Dir(s)  19,356,647,424 bytes free

C:\RhapsodyTesting\TestConductor\RQM_TestConductorAdapter>start.bat -repository
https://localhost:9443/jazz -user test -password test -adapter myAdapter -project
Area Quality+Manager
Launching RQM TestConductor Adapter...

Rational Rhapsody TestConductor Adapter: 2.4.2__alpha_1.3
Successfully created an HTTP client
The adapter is now connected
```

Fig. 1: An appropriate –dependent from your environment and configuration– call would be:
“start.bat -repository https://localhost:9443/jazz -user test -password test -adapter myAdapter
-projectArea Quality+Manager”

Note that inside the RQM installation in the adapters folder there is a RQMCommandLineAdapter.zip file which contains a readme.txt file describing the mandatory and optional parameters in more detail.

Note that the RQM TestConductor Adapter currently is only equipped with a start script for windows (there is no linux start script yet).

3. Executing Rhapsody Testconductor Tests via RQM

To execute Rhapsody TestConductor Tests in RQM at first Tests have to be specified in a Rhapsody model using TestConductor (3.1). Then in Rational Quality Manager TestScripts can be created (3.2) which are linked via the TestConductor Adapter with the tests in the Rhapsody model. The TestScripts are again part of a (RQM) TestCase which can be executed in RQM.

3.1 Preparing the Tests (the Rhapsody Part)

(It is advisable to have a basic knowledge about Rhapsody and TestConductor before creating TestConductor tests. Both tools are shipped with documentation and tutorial which are accessible via the help tab of Rhapsody.)

1. At first you need a Rhapsody model with a design model which you want to test. Let's assume you are using the sample model CppCashRegister (Samples/CppSamples/TestConductor/CppCashRegister).
2. Inside the model rightclick the class CashRegister in the package CashRegisterPkg and select "Create TestArchitecture" from the context menu.

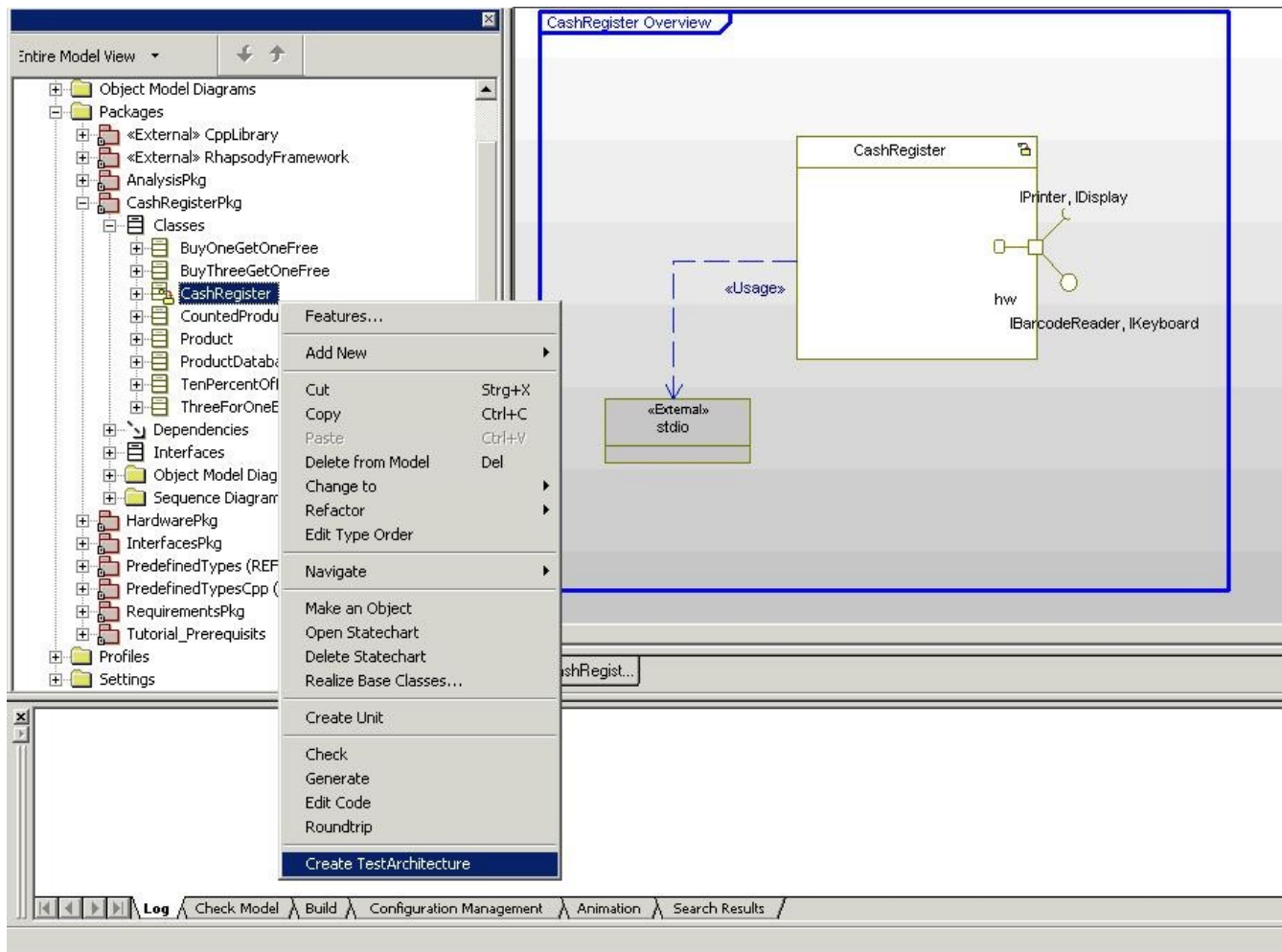


Fig. 2: Select "Create TestArchitecture" on class CashRegister

Then you have to agree to have the TestingProfile added to your model and a new

- (Test)package “TPkg_CashRegister” is created in the model.
3. Rightclick the TestContext “TCon_CashRegister” in the TestPackage and select “Create SDTestCase”.

An SDTestCase is created. Now fill the TestScenario of the TestCase with content (e.g. like in Fig. 3).

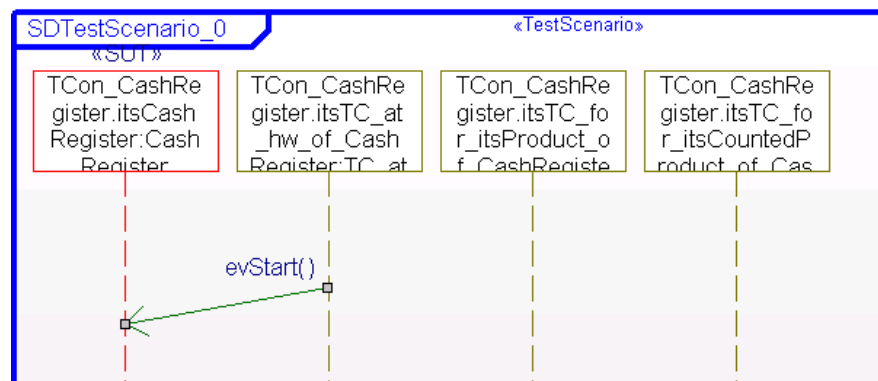


Fig. 3: A very simple TestScenario for an SDTestCase

4. Create more TestCases if you like.
5. Rightclick on the TestContext and select “Update TestContext” (the test model is populated with necessary driver operations)
6. Rightclick the TestContext and select “Build TestContext” (the TestContext and the TestCases are built and afterwards ready to be executed).
7. If the preceding steps were successful save the model. Be sure remember the names of the TestCases (and TestContext and TestPackage) as you need these later to reference them from inside of RQM.

3.2 Preparing the Tests (the RQM Part)

(It is advisable to have basic knowledge about Rational Quality Manager integrating Rhapsody TestConductor Tests into RQM. RQM is shipped with online help files and sample files.)

1. Start RQM on your RQM server machine (may be your local machine)
2. Open the website https://<rqm_server>:9443/jazz/web/console and login with your user login (after initial install only the ADMIN/ADMIN exists, you have to create a user with the correct rights before able to work with RQM, please see the RQM documentation for further details)
3. At first you need a **Test Plan**, create one if you do not have one already

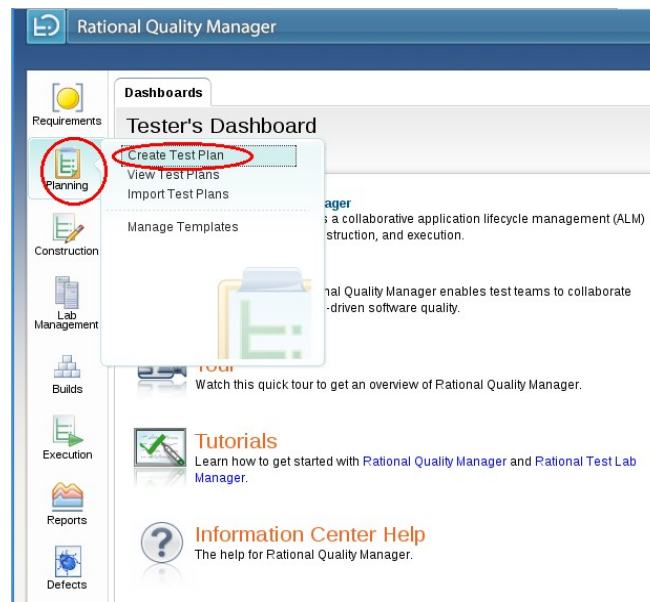


Fig. 4: Create a Test Plan

4. In the next step you create **Test Cases** as part of the Test Plan.

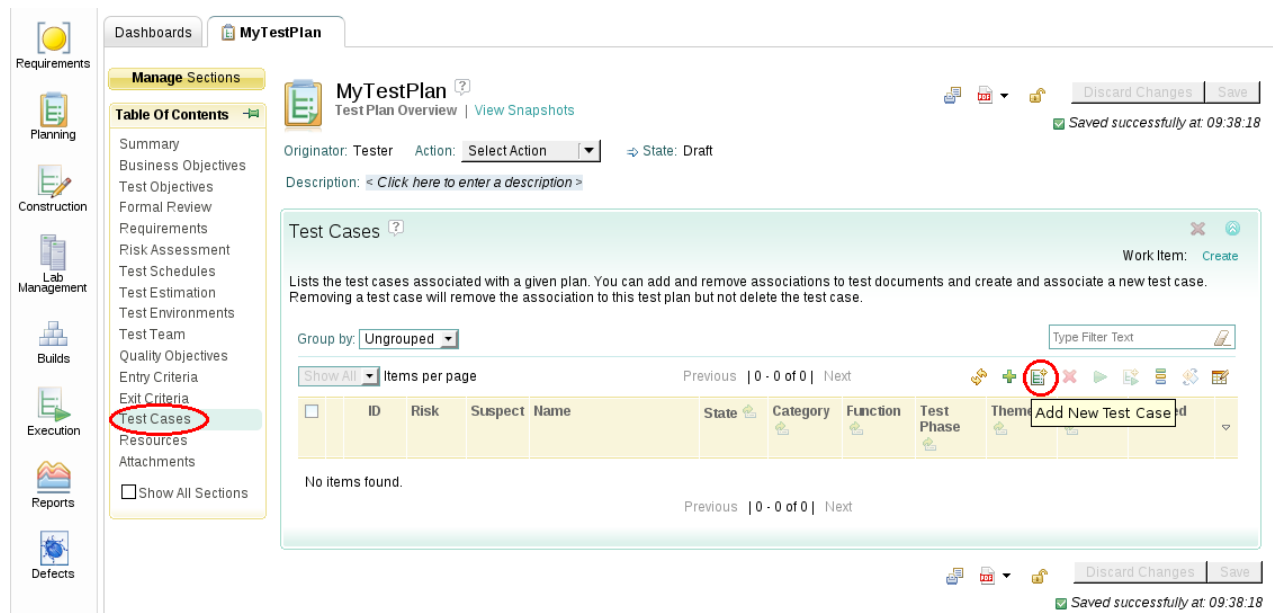


Abbildung 5: Inside the Test Plan overview you can open the Test Cases section and create a new Test Case

5. In the next step you create a **Test Script**

6. Inside the Test Plan overview you can open the Test Cases section and create a new Test Case

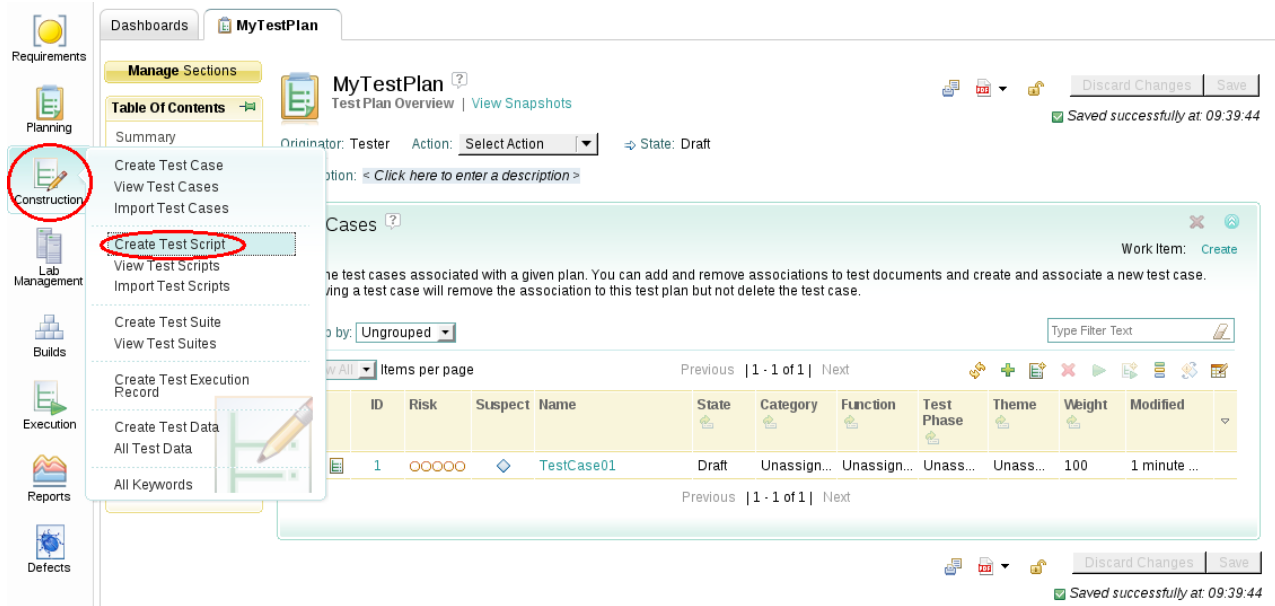


Fig. 6: Creating a Test Script (not yet connected to a Test Case)

- a) Specify a Test Script **name**
- b) Select as **type** “Rhapsody TestConductor”

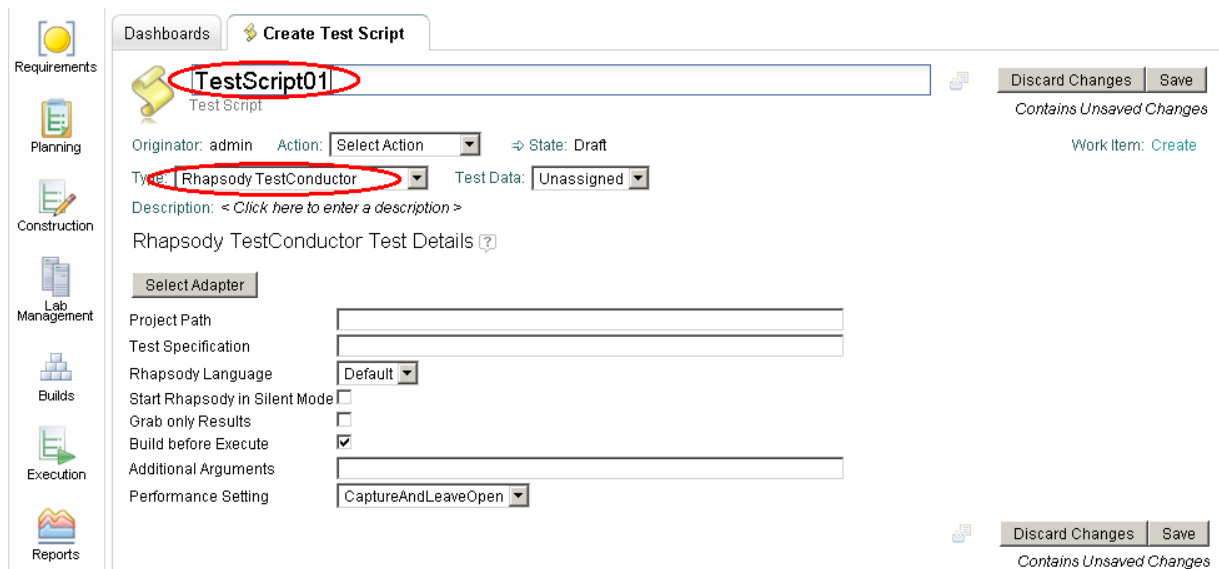


Fig. 7: Specify the name of the Test Script (here "TestScript01"), select the Type "Rhapsody TestConductor"

- c) After changing the type, the mask with the test script options changes
 1. “**Project Path**” is a mandatory setting and contains the path to the Rhapsody project on the test machine. The project path can be set manually in this field or automatically filled via the adapter dialog (see d.)).
 2. “**Test Specification**” is a mandatory field and contains the full path of the test element in the Rhapsody project.³ The test specification can be set manually in this field or

3 Valid elements are TestPackages, TestContexts and TestCases. To avoid a potential name clash, TestCases can be annotated with trailing “()”.

- automatically filled via the adapter dialog (see d.)).
3. Via “**Rhapsody Language**” the starting of a certain Rhapsody language version can be forced.^{4 5}
 4. If “**Start Rhapsody in Silent Mode**” is checked Rhapsody will be started without GUI to improve performance and reduce action on the display.⁶
 5. If “**Grab only Results**” is checked, the Rhapsody TestConductor test will not be updated, built and executed, but only the existent results are imported into RQM (requires results in model).
 6. If “**Build before Execute**” is not checked, the Rhapsody TestConductor test will not be updated and built before execution (requires uptodate build in model).
 7. In “**Additional Settings**” additional parameters can be passed to Rhapsody and TestConductor. These settings will be attached as they are to the Rhapsody call.
 8. With the setting “**Performance Setting**” the performance can be improved if running a series of tests by reusing the Rhapsody instance and the loaded model instead of starting and closing Rhapsody for each test.
 - With the (default) setting “*CaptureAndLeaveOpen*” the adapter will take over an potential existent Rhapsody. If the active project is the same as the project specified in the RQM test script, the active project will be reused, otherwise the active project will be closed and the correct project will loaded. If there is no Rhapsody active, a new Rhapsody instance will be started. In the first case the settings “Rhapsody Language” and “Start Rhapsody in Silent Mode” have no effect, in the second these settings will be considered. In both cases Rhapsody and the project will remain open after finishing the test.
 - The setting “*CaptureAndClose*” has the same effect as the setting described above, except that the adapter will close Rhapsody after finishing the test.
 - With the setting “*NoCapture*” the adapter will not take over an existent Rhapsody instance and it will close Rhapsody after finishing the test. In fact the setting requires that no Rhapsody instance is active and will result in error otherwise.
- d) The button “**Select Adapter**” (optional) opens a wizard dialog (see Fig 8) in which
1. a TestConductor adapter can be chosen from a list of currently running adapters (see fig 9).
 2. the Project path of the test project can be resolved on the test machine (see fig 10).
 3. the test specification can be selected among a list of all available test elements of the chosen project (see fig 11).
- (To resolve the list of tests the adapter has to contact a Rhapsody instance. Starting Rhapsody or connecting to existent instances will be done according the setting “PerformanceSetting”. Note that this may take some time especially if Rhapsody has to be started on the target machine for this task).

4 It is necessary e.g. for ADA models to start RiAda

5 Please not the effects of the performance settings on this feature.

6 Note that not “RhapsodyCI” is used but the standard Rhapsody application with the parameter -hiddenui

TestScript01
Test Script

Originator: admin Action: Select Action State: Draft

Type: Rhapsody TestConductor Test Data: Unassigned

Description: [Click here to enter a description](#)

Rhapsody TestConductor Test Details

Select Adapter

Project Path:

Test Specification:

Rhapsody Language: Default

Start Rhapsody in Silent Mode: ☐

Grab only Results: ☐

Build before Execute: ☒

Additional Arguments:

Performance Setting: CaptureAndLeaveOpen

Discard Changes Save

Contains Unsaved Changes

Work Item: [Create](#)

Discard Changes Save

Contains Unsaved Changes

Fig 8: Click on "Select Adapter" to invoke the adapter dialog

< Enter New Test Script Name >
Test Script

Originator: admin Action: Select Action State: Draft

Type: Rhapsody TestConductor Test Data: Unassigned

Import Test Script: Use local test resources

Step 1: Select an adapter

Select from: ☐ Test Cells ☒ Machines

Group by: Ungrouped Type Filter Text

Show All Items per page

Previous | 1 - 1 of 1 | Next

ID	Machine Name	Host Name	Adapter Name	Health	Reserved	Jobs
<input checked="" type="checkbox"/> 1	orr-rqh-rqm	orr-rqh-rqm	RQM Adapter	●		0

Previous | 1 - 1 of 1 | Next

< Previous Next > Finish

Fig. 9: Select your adapter

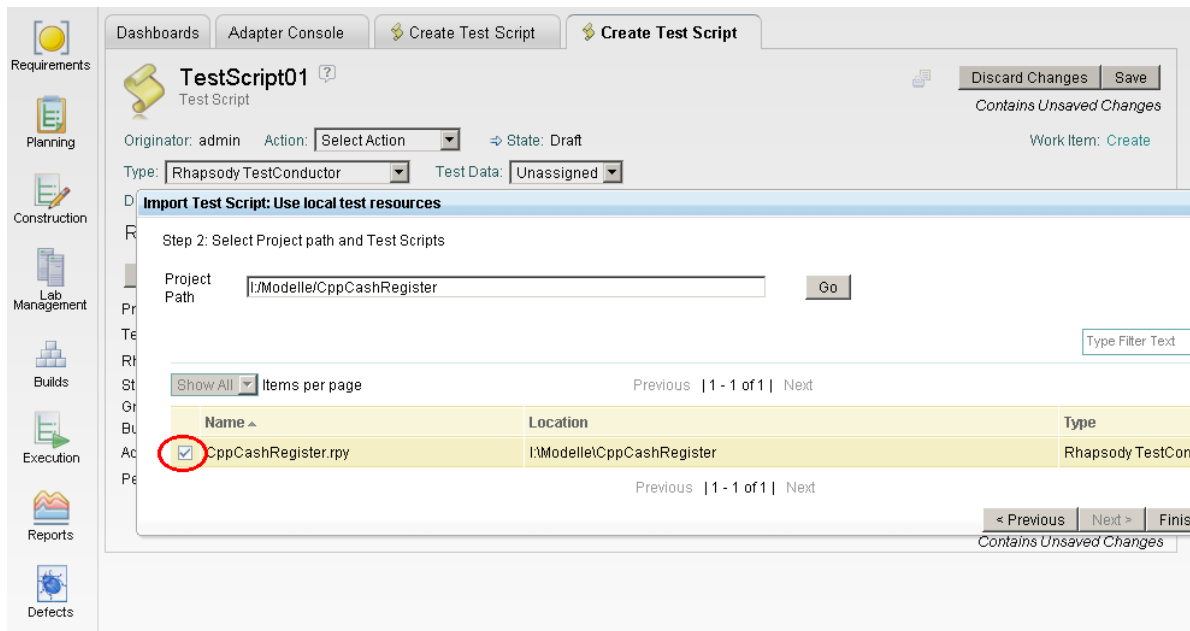


Fig. 10: Specify the (local) path, press "Go", wait, then select the model's project file, press "Next" to proceed with selecting the Rhapsody test element or press "Finish" if you want to specify the test manually.

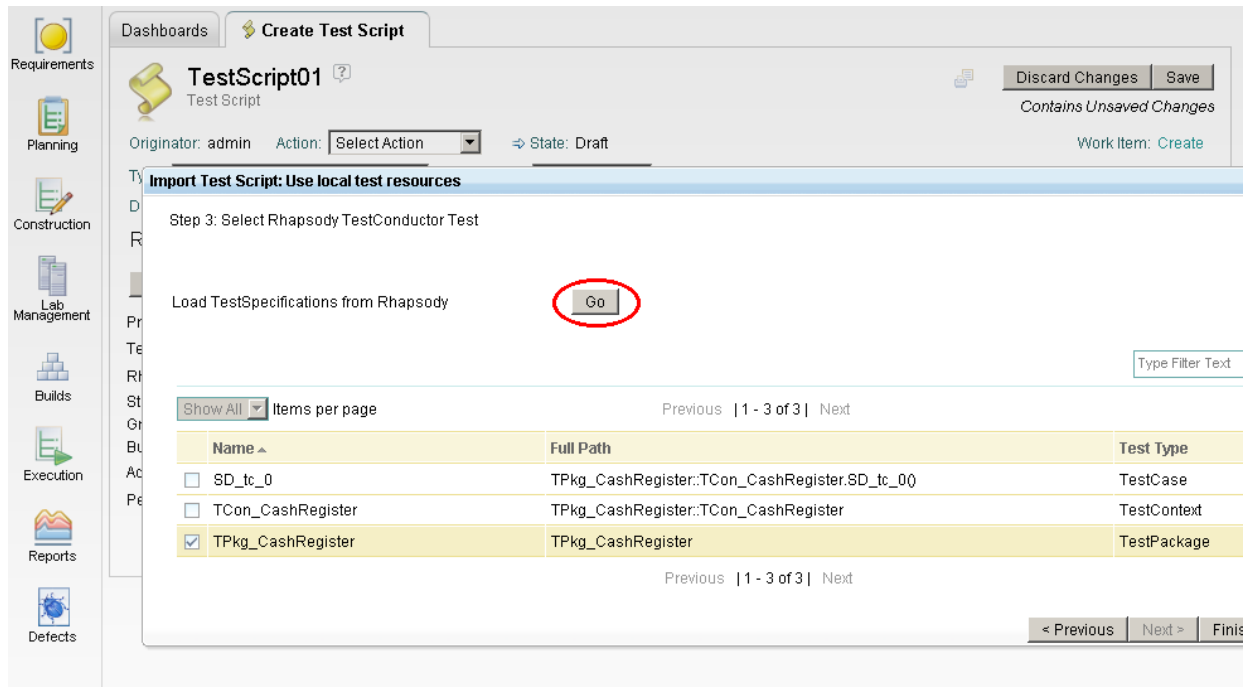


Fig 11: Click on "Go" to import the available tests from Rhapsody (this may take a while if Rhapsody has to be started on the target machine), select one and press finish to transfer your selection into the RQM Test Script specification.

- e) Save the Test Script (Repeat if necessary for other Test Scripts)
7. Now the (RQM) TestScripts have to be linked with the (RQM) Test Cases.
 - a) Open the Test Case in RQM
 - b) Add existing Test Script

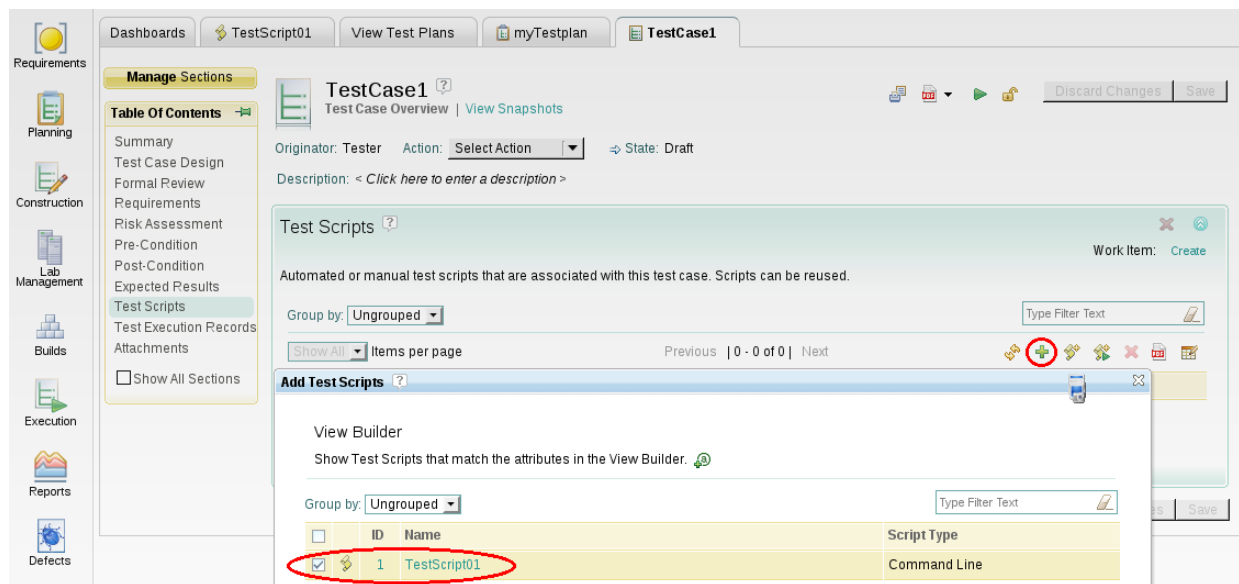


Fig. 13: In the Test Case click "add existing Test Scripts" and select your previously created Test Script

- c) Save the Test Case

4. Executing and Evaluating

You can execute a (selected) Test Case (prepared as described in the preceding section) or Test Script by clicking on the green triangle at the list of Test Cases of a Test Plan or at the list of Test Scripts of a Test Case.

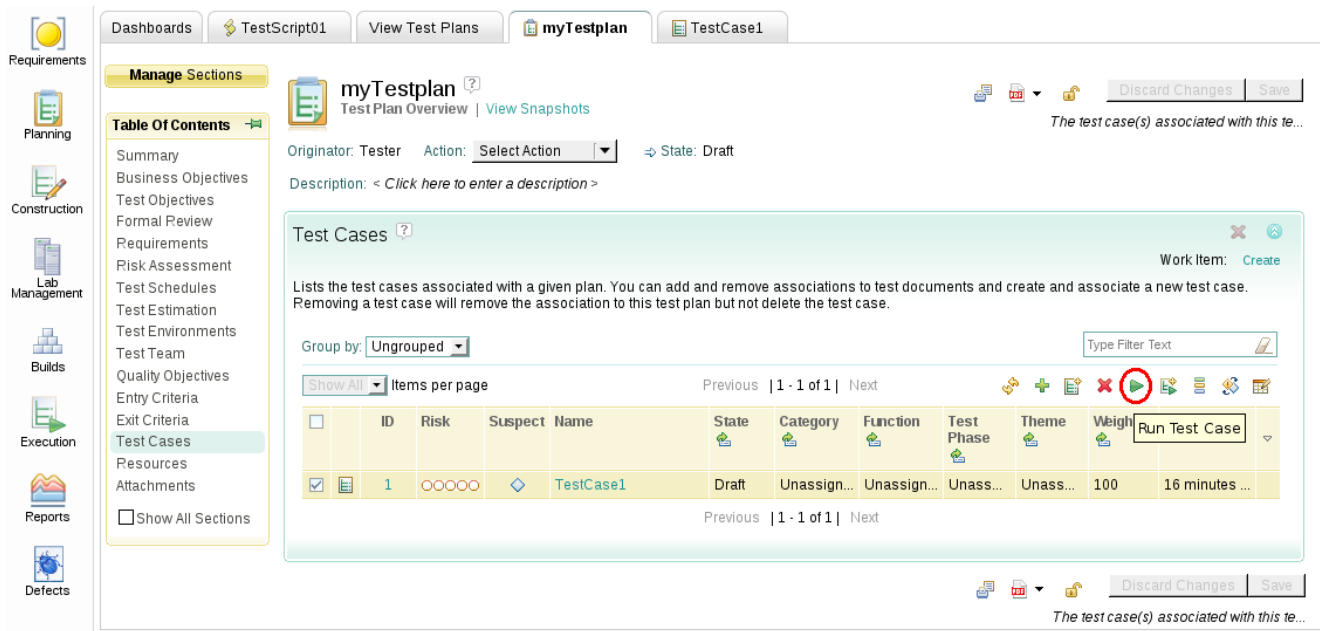


Fig. 14: In the list of Test Cases of the Test Plan select the Test Case and click “Run Test Case”

After starting the test RQM contacts the corresponding adapter and shows information about it. If the adapter is online and ready, the user can start the test.

In the background the TestConductor adapter starts Rhapsody, loads the model, tries to resolve the specified test, updates, builds and executes the test and imports the results into RQM.

After the test has finished, you can evaluate the results.

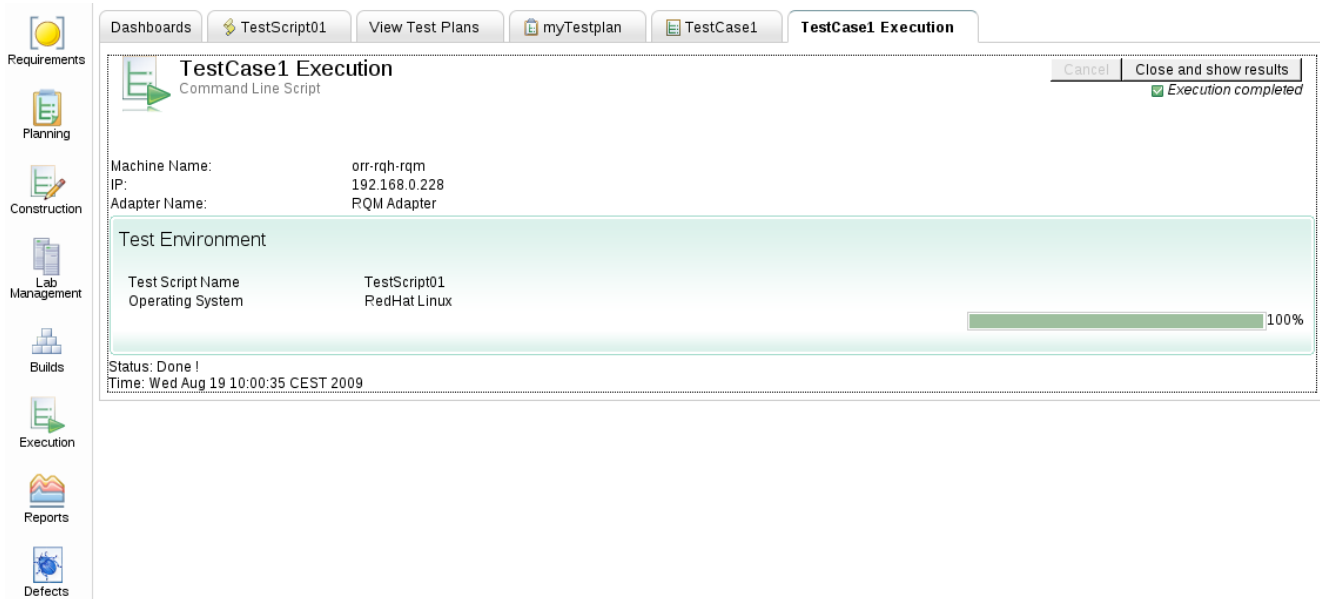


Fig. 15: Test execution has finished (without error), the results are ready to be reviewed.

If the test was executed successfully (result is passed or failed), there is a test result file (and if configured so also coverage result file) in the result file section. If an error occurred only a log file is imported. This file contains the course of the test and also possible errors. All files can be shown by clicking on them.

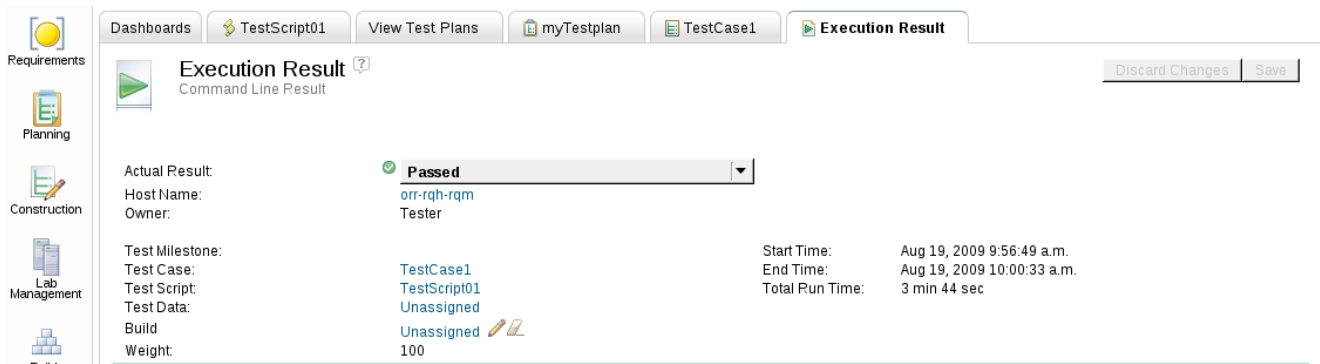


Fig. 16: The Test is passed.



Fig. 17: The result files of a (passed) test.

5. Troubleshooting

- If the **tests in RQM do not finish** or there is some other kind of weird behavior (I) please check in your task manager whether there are Rhapsody processes still active in the system. Since the adapter calls Rhapsody per default with hidden ui, the user does not see any Rhapsody instances which were wrongly not closed after a test run previously.
- If the **tests in RQM do not finish** or there is some other kind of weird behavior (II) please check in your task manager whether there are unaccounted java processes still active in the system and if so terminate them (after verifying they are really unaccounted). (Currently the java adapter process still remains active after terminating the adapter console. This java process may interference with a newly started adapter)
- If you want to have (model) **coverage information** imported to RQM, be sure you have switched Model Coverage on in the TestCase properties (inside Rhapsody).
- **The adapter does not start, complaining about a missing or wrong entry of the Rhapsody API in the class path:** In the file `start.bat` check and adjust the value of the variable `RhapsodyPath` (line 7) according to your Rational Rhapsody installation.
- In the rare case of an ambiguity between a TestCase and a TestContext name (if a TestContext contains another nested TestContext with the same name of a TestCase inside the outer TestContext –this constellation is not advised), the adapter chooses the TestContext before the TestCase. If you want to execute the TestCase you can add “()” behind the Test's name, to indicate you mean the TestCase instead of the equally named TestContext.