Managing Software Testing Automation Framework with Rational Quality Manager

Rajat Pratap Singh

August 21, 2012

"Rational Support Whitepaper"

Introduction

This white paper highlights the integration of IBM Rational Quality Manager (RQM) with the STAF and STAX Automation Framework. Using this integration, you can centrally manage the client machines and trigger the automation from within Rational Quality Manager. This white paper is useful for those who use Rational Quality Manager as the tool for running and maintaining test execution.

Rational Quality Manager enables quality assurance teams to track all aspects of the quality assurance effort. The central artifact in the tool is a dynamic test plan that contains all information pertaining to the quality assurance effort, such as goals, schedules, milestones, and exit criteria. The central artifact also contains links to associated test cases, requirements, and development work items. Rational Quality Manager further includes modules for requirements management, manual test authoring and execution, test lab management, test execution, reporting and defect management.

Rational Quality Manager is a Web 2.0 application that runs in a browser. Data storage and management occurs on a Jazz Team Server (JTS) with the Quality Management application installed. Rational Quality Manager builds off the IBM Rational Jazz technology platform. The Jazz technology platform is a common server foundation shared by several Rational tools that facilitates information sharing between teams and applications.

Through the Jazz technology platform, Rational Quality Manager can share requirements information with IBM Rational Requirements Composer (RRC) and share work items with IBM Rational Team Concert (RTC). The Jazz platform also provides a universal API for sharing information with other vendor applications.

Rational Quality Manager is an integral part of the Collaborative Lifecycle Management (CLM) solution that provides seamless integrations for application lifecycle management, with reduced cost of ownership and administration, improved cross-product visibility and reporting, and lifecycle integrations.

Software Testing Automation Framework (STAF) is an open source, multi-platform, multi-language framework designed around the idea of reusable components called "services". These services can include process invocation, resource management, logging, and monitoring. STAF removes the tedium of building an automation infrastructure, thus enabling you to focus on building your automation solution. The STAF framework provides the foundation upon which to build higher-level solutions, and provides a plug-able approach supported across a large variety of platforms and languages.

You can use STAF to help solve common industry problems. These problems include frequent product cycles, lack of preparation time, reduced testing time, more platform choices, more programming language choices, and increased National Language requirements. STAF can help in these areas since it is a proven and mature technology, promotes automation and reuse, has broad platform and language support, and provides a common infrastructure across teams.

The **Software Testing Automation Execution engine (STAX)** can help you thoroughly automate the distribution, execution, and results analysis of your test cases.

STAX builds on top of three existing technologies, STAF, XML, and Python, to place great automation power in the hands of testers. STAX also provides a powerful monitoring application that allows you to interact with and monitor the progress of jobs.

Some of the main features of STAX include:

- Support for parallel execution

- User-defined granularity of execution control

- Support for nested test cases

- The ability to control the length of execution time

- The ability to import modules at run-time

- Support for existing Python and Java modules and packages

- The ability to extend both the STAX language and the STAX monitoring application

Using these capabilities, you can build sophisticated scripts to automate your entire test environment, while ensuring maximum efficiency and control.

# Rational Quality Manager, STAF, and STAX

Rational Quality Manager has the capability to integrate with STAF and STAX which enables it manage all the automation from within the tool. An XML file contains the definitions for the STAF clients. The file resides in Rational Quality Manager installation folder. Through this XML file, Rational Quality Manager registers the clients and makes it available in its user interface.

You can centrally manage these clients and trigger the automation on the necessary STAF client. Once the tests complete on the client machine, the results transfer to the Rational Quality Manager user interface.

There are two deployment models for running STAF and STAX; there is an integrated STAX server and an external STAX server. In both cases, the Jazz Team Server and test machines must have the STAF client installed and running.

Architecture

Central Test Automation Tool -- Architecture
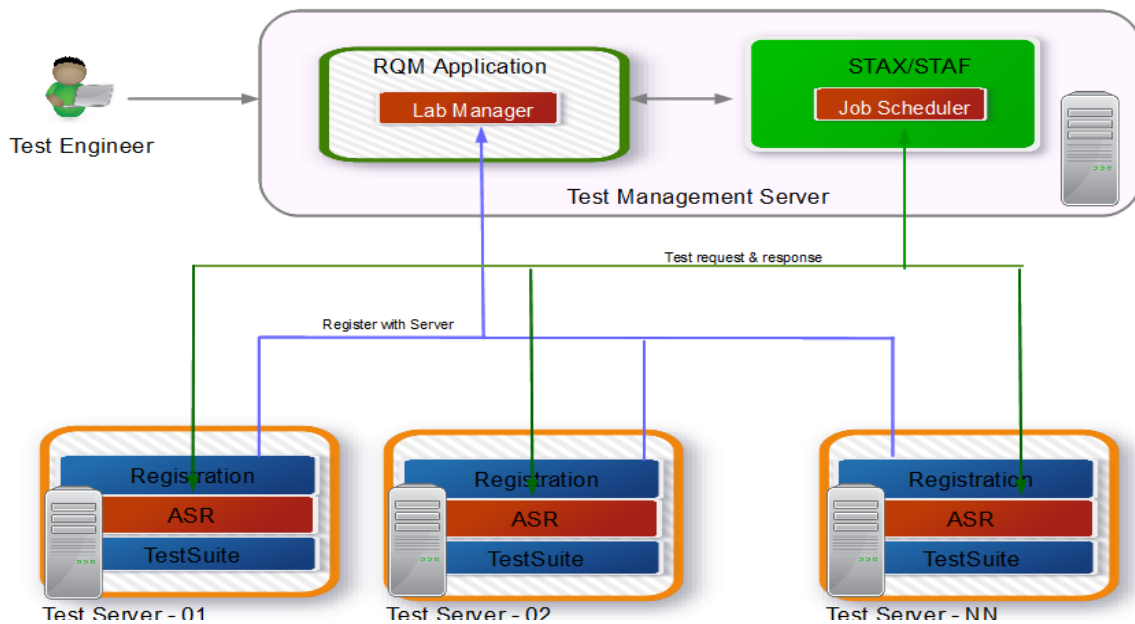
Configure STAF and STAX

1. Install STAF V3.3 and STAX V3.0. You can acquire both with this URL:

    http://staf.sourceforge.net/getcurrent.php

2. Set the **PATH** variable to include the "<STAF home>/bin" location.

3. Set the **PATH** variable to include the "<Java Home>/bin" location.

4. Set the **JAVA_HOME** variable to include the JDK location.

5. Modify the **staf.cfg** file in the "<STAF home>/bin" directory with the IP
    address or host name to add the trust level for a two-way communication.

```
# Turn on tracing of internal errors and deprecated options
trace enable tracepoints "error deprecated"

# Enable TCP/IP connections
interface ssl library STAFTCP option Secure=Yes option Port=6550
interface tcp library STAFTCP option Secure=No  option Port=6500

# Set default local trust
trust machine local://local level 5

# Add default service loader
serviceloader library STAFDSLS

SERVICE STAX LIBRARY JSTAF EXECUTE {STAF/Config/STAFRoot}/services/stax/
SERVICE EVENT LIBRARY JSTAF EXECUTE {STAF/Config/STAFRoot}/services/stax
SET MAXQUEUESIZE 10000

#SERVICE STAX LIBRARY JSTAF EXECUTE {STAF/Config/STAFRoot}/services/stax
#SERVICE EVENT LIBRARY JSTAF EXECUTE {STAF/Config/STAFRoot}/services/sta
```

Enable Rational Quality Manager

The STAX jobs run on a Jazz Team Server server with the Quality Management application. Therefore, there must be a STAX client installation configured on the Jazz Team Server. Configure the STAX adaptor in Rational Quality Manager so that you can run STAX tasks.

1. Modify the **integration_config.xml** file on the Jazz Team Server to define the STAX Adaptor. Here is the file location:

    <Jazz Team Server>\server\conf\qm

```xml
- <adapter>
    <name>stax_adapter</name>
    <version>1.0</version>
    <APIVersionSupported>1.0</APIVersionSupported>
    <inventoryExtensionId>com.ibm.rational.test.lm.spi.stax.inventory</inventoryExtensionId>
    <automationExtensionId>com.ibm.rational.test.lm.spi.stax.automation</automationExtensionId>
    <pluginId>com.ibm.rational.test.lm.spi.stax</pluginId>
  - <instance>
      <instanceId>stax_adapter</instanceId>
      <hostname>localhost</hostname>
      <port>0</port>
      <offline>FALSE</offline>
      <staxPath>C:\stax</staxPath>
      <scanInterval>180</scanInterval>
    </instance>
  </adapter>
```

2. After modifying the changes in the XML file, restart the Jazz Team Server.

Register STAX Adapter with Jazz Team Server

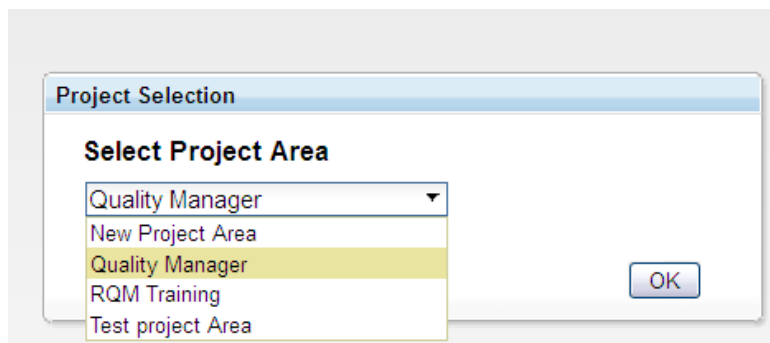1. Start the STAF server on the Jazz Team Server machine.

2. Start Jazz Team Server.

3. Log into your project area in Rational Quality Manager using this URL:

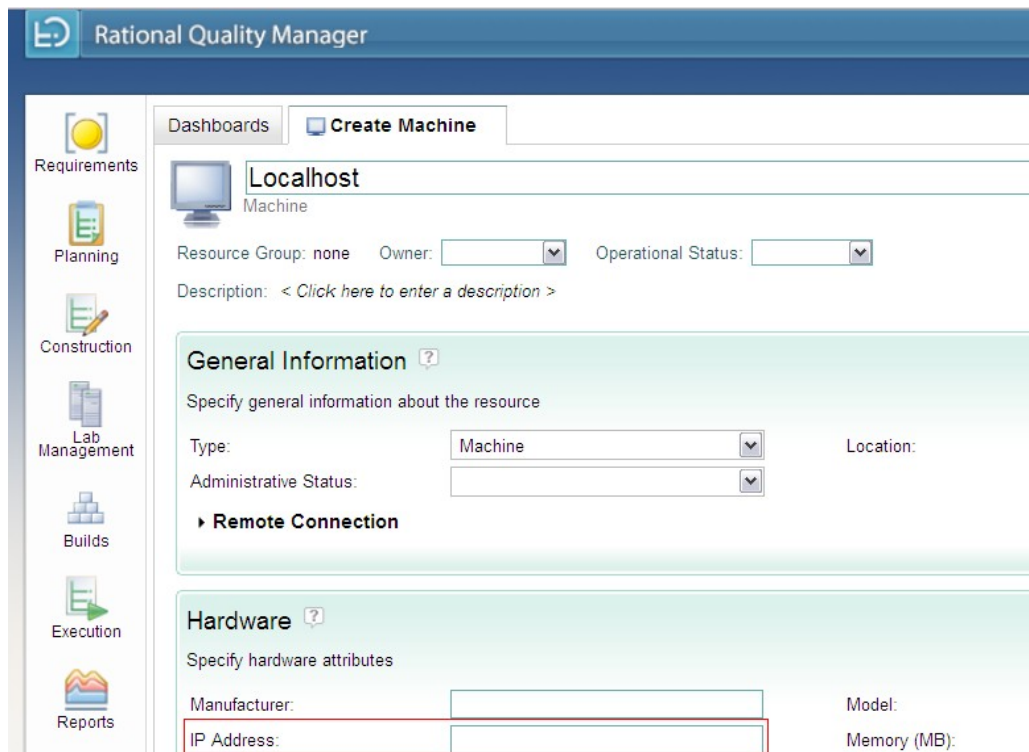   **https://<server>:9443/jazz/web/console**

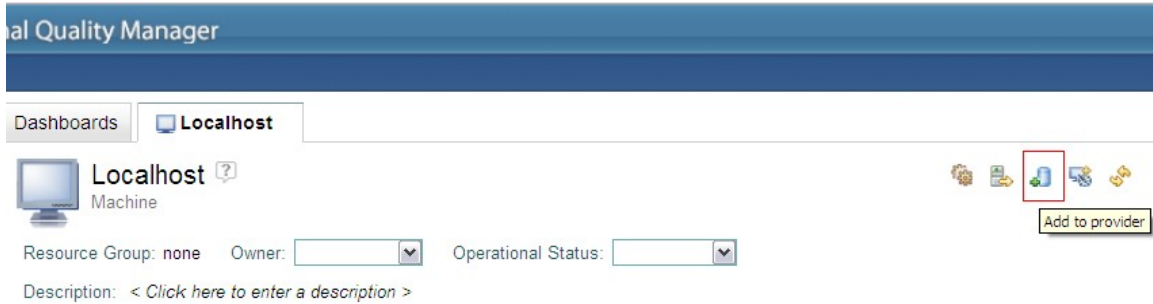   The "server" value of the URL is the IP address or domain name of the server.

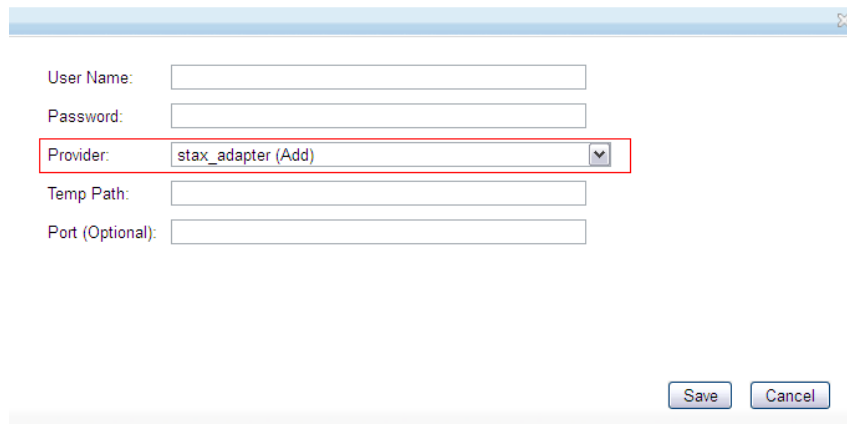4. Create a lab machine in Rational Quality Manager for the STAX client.

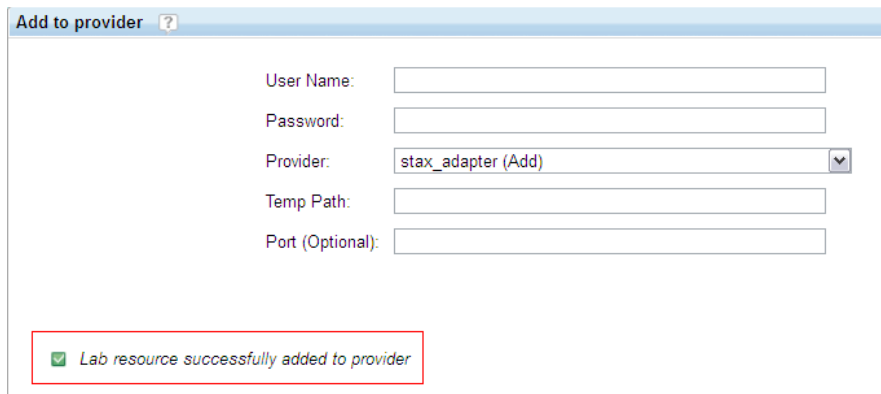5. Enter the name and IP address for the client machine and save the changes.

6. Click the **Add to provider** button to add the STAX adapter to Rational Quality Manager.

7. Select the configured provider and click Save:

8. Wait until you get the confirmation stating that the lab resource successfully added to the provider.

Execution of STAX jobs from Rational Quality Manager

1. Place the facade file under the STAX path entered in the **integration_config.xml** file.

The facade file ports the STAX XML files for execution from Rational Quality Manager. Here is what the default facade task looks like:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE STAX SYSTEM "stax.dtd">
<STAX>
       <defaultcall function="myLegacyTask"/>

       <function name="myLegacyTask" scope="local">
               <function-prolog>Legacy STAX Task</function-prolog>

               <function-epilog>
                       This is a description of My Legacy Task.
               </function-epilog>

               <function-list-args>
                       <function-required-arg name="target">The target
machine</function-required-arg>
                       <function-optional-arg name="arg1"
default="'default1'">The first arg description</function-optional-arg>
                       <function-optional-arg name="arg2"
default="'default2'">The second arg description</function-optional-arg>
               </function-list-args>

               <sequence>
                       <!-- Import and invoke the legacy task -->
                       <import machine="STAXJobXMLMachine"
file="'%s/../../legacy/myOldTask.xml' % STAXJobXMLFile" import-
include="myOldFunction"/>
                       <call
function="'myOldFunction'">[target,arg1,arg2]</call>
                       <return>0</return>
               </sequence>
       </function>
</STAX>
```

2. In Rational Quality Manager, click **Lab Management > All Automations**. This will list the STAX task coming from the facade file.

3. Select the automation task. Select the client for the execution and click **Run**.

4. The execution loads the default arguments mentioned in the facade file. You can modify the arguments depending the on the requirement.

Properties

arg1	= default1
arg2	= default2

OK   Cancel

5. Click **OK** to start the execution. Scroll down to see the running automations with their results.

Running Automations

Group by: Ungrouped

Type Filter Text

10   Items per page   Previous | 1 - 10 of 117 | Next

ID	Name	Start Time	Duration	End Time	State	Result	Machine	Job Log	Cancel
31	Legacy STA...	1 minute ago			Running	Waiting	Localhost	Log	✖
30	Legacy STA...	Jul 26, 2012	51s 65...	Jul 26, 2012	Completed	Fail		Log	✖
29	Legacy STA...	Jul 25, 2012	24s 73...	Jul 25, 2012	Completed	Pass		Log	✖
28	Legacy STA...	Jul 25, 2012	14s 79...	Jul 25, 2012	Completed	Fail		Log	✖
27	Legacy STA...	Jul 25, 2012	27s 45...	Jul 25, 2012	Completed	Fail		Log	✖
26	Legacy STA...	Jul 25, 2012	26s 96...	Jul 25, 2012	Completed	Fail		Log	✖
25	Legacy STA...	Jul 25, 2012	23s 75...	Jul 25, 2012	Completed	Fail		Log	✖
24	Legacy STA...	Jul 25, 2012	20s 39...	Jul 25, 2012	Completed	Fail		Log	✖
23	Legacy STA...	Jul 25, 2012	10s 90...	Jul 25, 2012	Completed	Fail		Log	✖
22	Legacy STA...	Jul 25, 2012	17s 75...	Jul 25, 2012	Completed	Fail		Log	✖

Previous | 1 - 10 of 117 | Next

6. Click **Log** to see any errors in the execution.

Troubleshooting and best practices

Here are some troubleshooting measures and best practices for the integration of Rational Quality Manager and STAX.

1. Ensure that the **staf.cfg** file for the automation client contains the IP address or host name with the appropriate trust level for Jazz Team Server.

2. The **STAFProc** must run on the Jazz Team Server and the automation client.

3. The **<MainTask>.xml** file must follow proper schema for the function calls.

4. The **<MainTask>.xml** file must have proper read, write, and execute permission on the Jazz Team Server machine.

References

- See the "*Rational Quality Manager and STAF STAX*" topic in the **CLM Information Center**:

  http://www.ibm.com/support/docview.wss?uid=swg27014016

- Software Testing Automation Framework on SourceForge:

  http://staf.sourceforge.net/

Copyrights, trademarks, and disclaimers

IBM, the IBM logo, ibm.com, and Rational are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS DOCUMENT, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS DOCUMENT OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS DOCUMENT IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REDOCUMENTS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.