



**DevOps Practice, Automated Testing Initiative (ATI)
Automated Test Scripts Usage and Design Documentation**

Version #: 1.3

Date: 03/18/2020



Table of Contents

1	Introduction	6
1.1	Purpose	6
1.2	Objective	6
1.3	Scope.....	6
1.4	Assumptions.....	7
1.5	Definitions, Acronyms, and Abbreviations.....	7
2	Automated Testing Approach	7
3	Test Automation Framework.....	8
3.1	Overview	8
3.2	Supported Browsers.....	8
3.3	Core Framework (Framework Components)	8
3.4	Global Constants/Properties File	9
3.5	Reporting Capabilities	10
4	Installation Instructions	11
4.1	Pre-Requisites	11
4.2	Installation Process	11
4.3	Project Layout (Folder Structure)	13
4.4	Test Case Sheet	13
5	Sample Script Creation.....	13
5.1	Workflow.....	13
5.2	Script Creation	14
6.	Test Script Local Execution.....	15
6.1	Test Script Level	15
6.2	Test Suite Level	16
6.3	Project Level.....	16
7.	Test and Validate Automation Script	16
7.1	Environment Configurations	16
7.1.1	Pre-Requisites:	16
7.2	Retrieve Latest Test Scripts.....	17
7.2.1	Updating Properties File	17
7.3	Test Script Execution.....	18
7.4	Verifying the Logs.....	18



7.5	RQM Integration Process	18
7.5.1	Setting Up RQM	18
7.5.2	Configuring JUnit Selenium Adapter.....	26
7.5.3	Starting JUnit Selenium Adapter.....	29
7.5.4	Executing Test Cases from RQM	30
7.5.5	Executing Test Suite from RQM	35
7.5.6	Defect Reporting Process.....	39
8.	CICD Handoff.....	40
8.1	RQM Execution	40
8.1.1	Prerequisites	40
8.1.2	Configuring Selenium JUnit Adapter.....	40
8.2	CICD Project Checklist	40
8.3	Service Account Creation Process.....	40
9.	CICD Integration.....	40
9.1	Configuration Instructions	40
9.2	Installation Instructions	41
9.3	RQM Integration Process	43
9.4	RQM Command Line Execution Tool	43
9.5	RTC Configuration	43
10.	Project Team Delivery and Acceptance	44
11.	References	44
12.	Version History.....	46



Table of Figures

Figure 1 – Automated Testing Approach	7
Figure 2 – Test Automation Framework	8
Figure 3 – Property File Location	10
Figure 4 – RTC SCM FATCA QI.....	12
Figure 5 – RTC SCM Project Explorer	12
Figure 6 – Project Layout	13
Figure 7 – Test Script Creation Workflow	13
Figure 8 – New Class Creation	14
Figure 9 – Create New Class Completion	14
Figure 10 – Validate New Class	15
Figure 11 – Extend Class to ScriptBase	15
Figure 12 – Create Test Script Method	15
Figure 13 – Test Script Execution.....	16
Figure 14 – Test Suite Execution.....	16
Figure 15 – Project Execution	16
Figure 16 – SAT Config	17
Figure 17 – ConfigManager.....	18
Figure 18 – Create RQM Test Plan	19
Figure 19 – RQM Test Plan Sample	19
Figure 20 – Create RQM Test Case.....	19
Figure 21 – RQM Test Case Sample	20
Figure 22 – Create RQM Test Script.....	21
Figure 23 – Selecting JUnit Adapter for Test Script	21
Figure 24 – RQM Test Script Sample.....	21
Figure 25 – RQM Test Case Sample	22
Figure 26 – Selection RQM Test Script.....	22
Figure 27 – Linking to RQM Test Script to RQM Test Case Sample	22
Figure 28 – Create RQM Test Suite	23
Figure 29 – RQM Test Suite Sample.....	23
Figure 30 – RQM Test Suite Sample.....	23
Figure 31 – Add RQM Test Case to RQM Test Suite	23
Figure 32 – Linking RQM Test Case to RQM Test Suite Sample.....	24
Figure 33 – RQM Test Plan Sample	24
Figure 34 – RQM Test Suite View.....	25
Figure 35 – Linking RQM Test Suite within RQM Test Plan	25
Figure 36 – Mapping RQM Test Script to RQM Test Case.....	25
Figure 37 – Sample RQM Test Script.....	26
Figure 38 – JUnit Selenium Adapter Folder	26
Figure 39 – Creating Bat File in JUnit Selenium Folder	27
Figure 40 – Selenium Properties	27
Figure 41 – Selenium 3.8 Folder Structure	28
Figure 42 – Selenium 3.8.1 Java Folder Structure.....	28



Figure 43 – Selenium 3.8.1 Drivers Folder Structure	28
Figure 44 – Selenium 3.8 Library Within Libs Folder	28
Figure 45 – Complete Set of Selenium 3.8 Library Files.....	29
Figure 46 – JUnit Selenium Adapter Folder	29
Figure 47 – Starting Adapter in CMD window	30
Figure 48 – Adapter Console.....	30
Figure 49 – RQM Test Case Sample	31
Figure 50 – RQM Test Case Execution Mode.....	31
Figure 51 – Run Test Case	31
Figure 52 – Select a Machine to Run This Test	32
Figure 53 – Define Execution Record Detail	32
Figure 54 – Starting the Test.....	32
Figure 55 – Navigating FATCA QI Application	32
Figure 56 – Showing Error Message.....	33
Figure 57 – Test Case Execution Status.....	33
Figure 58 – Test Case Status	33
Figure 59 – Error Message for Fail Test Case	34
Figure 60 – Downloading PNG File.....	34
Figure 61 – Opening Error Image.....	35
Figure 62 – Opening Out File	35
Figure 63 – Opening Out File in Text.....	35
Figure 64 – RQM Test Suite Sample.....	36
Figure 65 – RQM Test Suite Showing Test Cases	36
Figure 66 – Executing Test Suite	36
Figure 67 – Define Record Detail for Test Suite	37
Figure 68 – Identifying an Adapter	37
Figure 69 – Selecting an Adapter	37
Figure 70 – Adapter Selected.....	38
Figure 71 – Executing Test Cases with Test Suite	38
Figure 72 – Test Suite Execution Status	38
Figure 73 – Showing Status of Each Test Case	39
Figure 74 – Defect Reporting Process.....	39



1 Introduction

The IRS DevOps Practice is constantly exploring opportunities to shorten the release cycle, reduce operational spend and deploy applications to multiple environments. To meet these needs the Automated Testing Initiative (ATI) was started in September 2018 to identify and focus on onboarding applications for test automation and evaluate existing test automation frameworks. Currently, the ATI Team is developing an automated testing framework to significantly accelerate repetitive and time-consuming manual testing processes. The developed automation framework will improve accuracy, breadth, and depth of application testing conducted by the Enterprise System Testing (EST) Team.

1.1 Purpose

The purpose of this user guide is to provide technical guidance on how to develop, test and validate the Accenture-built, Java-based framework using Selenium Web Driver. This framework is a Java-based development automation framework for regression testing.

1.2 Objective

The primary objective of this user guide is to highlight the main features and benefits of the Selenium Automation Framework created for the ATI Team. This guide will provide a walkthrough of the framework layout and development testing process as well as the steps required to validate the developed test scripts. The primary drivers for creating the automated testing framework for using Selenium are as follows:

- Reduce test scripting and execution time
- Optimize test team resource utilization
- Enable more test coverage for regression testing
- Easily identify defects not captured by manual regression testing

1.3 Scope

The following identifies the scope of the automation framework user guide:

In Scope:

- Framework components
- Framework installation
- Test script creation
- Local test script execution
- ATI test script validation
- RQM integration
- CICD reference information

Out of Scope:

- CICD processes and procedures
- CICD environment configurations

1.4 Assumptions

This document operates under the following assumptions:

- The reader is familiar with testing frameworks and methodologies
- The reader is familiar with Eclipse, Java, Selenium, and Maven
- The reader is familiar with Rational Quality Manager (RQM)
- Foreign Account Tax Compliance Act - Qualified Intermediary (FATCA QI) was used as a reference point to showcase the created Selenium framework

1.5 Definitions, Acronyms, and Abbreviations

Abbreviations	Description
ATI	Automated Testing Initiative
EST	Enterprise System Test
FATCA QI	Foreign Account Tax Compliance Act - Qualified Intermediary
MVC	Model View Controller
RTC	Rational Team Concert
RQM	Rational Quality Manager
SCM	Source Code Management

2 Automated Testing Approach

The Automated Testing Initiative Team adopted a three-phased approach to develop, execute, validate and maintain the test automation suite.

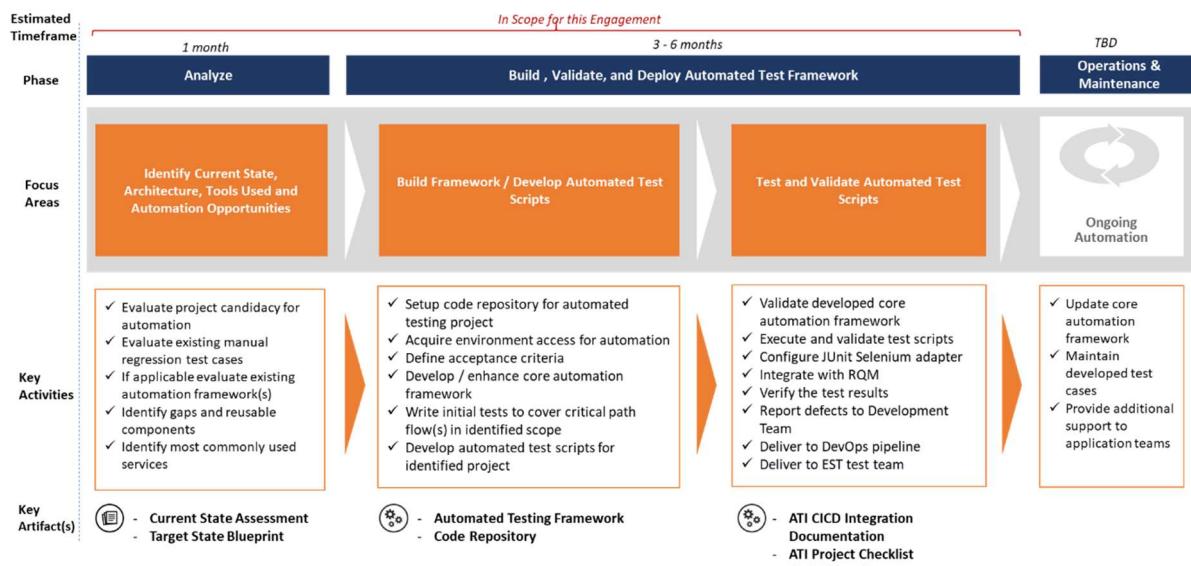


Figure 1 – Automated Testing Approach

- **Analyze** - Focuses on establishing the infrastructure (hardware and software) requirements to develop an automation solution for evaluating projects for automation, evaluating existing manual test cases and identifying gaps and reusable components. Additionally, best practice scenarios and application-specific scenarios are identified.

- **Develop, Validate and Deploy Automated Test Framework** - Consists of primarily three (3) major activities (Build, Validate and Deploy). During this phase the core components of the automation framework are built using Selenium. The ATI Developers then leverage the core components to build, test, and validate test scripts locally. Validated test scripts are then checked into Rational Team Concert (RTC) and handed over to the ATI Testers for validation.
- **Operations and Maintenance** - As per the changes in the various code drops, the test cases require updates on a regular basis due to changes in user interface, business logic and workflow. The changes resulting from updates to the underlying software need to be maintained to keep the scripts relevant. Depending on the changes to the underlying software, impact analysis is conducted to assess the changes. Once changes are analyzed, corresponding updates to components and test scripts are initiated by updating the underlying code/object repository. EST Leadership is currently working to determine who will provide this.

3 Test Automation Framework

3.1 Overview

The ATI Team has developed a framework that assists developers in creating and executing test scripts easily and effectively. Beneficial outcomes such as increased code reusability, higher portability, scalability, reduced script maintenance cost and higher code readability are achieved with this framework. Commonly encountered test scenarios will be grouped into a test suite that will be executed when a successful build is deployed to the test environment(s).

3.2 Supported Browsers

The current automated framework provides support across the following browsers and can be extended as needed:

- Internet Explorer v-11
- Google Chrome v-72
- Firefox v-65.0.1

3.3 Core Framework (Framework Components)

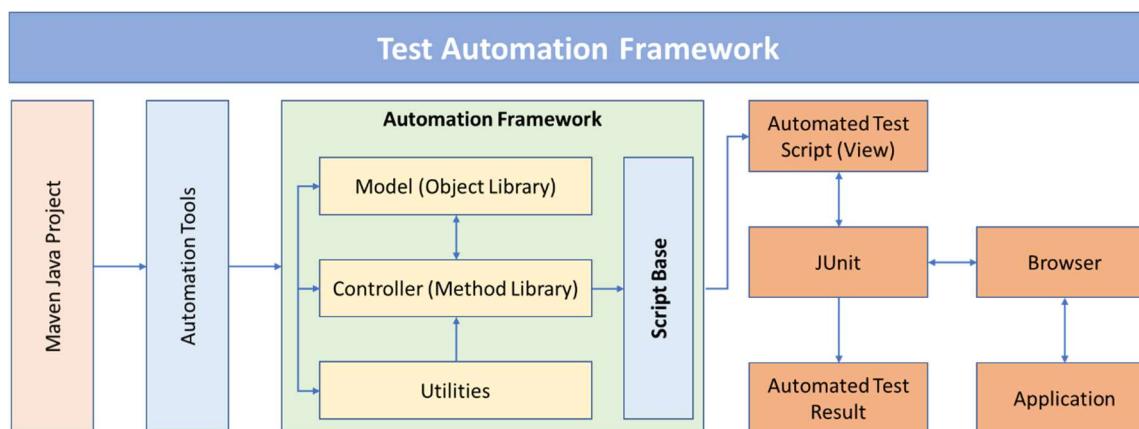


Figure 2 – Test Automation Framework



The Core Framework consists of Java classes and methods that are based on the Model View Controller (MVC) framework methodology. These classes and methods help us invoke and start the execution of the automated test scripts. The framework contains four (4) components inside the source folder:

- **Model:** Identifies static/dynamic objects/elements for the web application for each page. There are various HTML elements which are present on any user interface upon which the tester wants to perform automation operations. The following Web Elements are handled in the current framework:
 - Text Box Handler
 - Radio Button Handler
 - Checkbox Handler
 - List Handler
 - Dropdown Handler
 - Button Handler
 - Link Handler
 - Label Handler
 - Text Area Handler
 - Runtime Dynamic Element Handler
- **Controller:** Implementation of identify and manage objects/elements from model for web application to enable each page to perform its action
- **Script Base:** Automated JUnit test script that calls only methods from controller in a modular based for each step. There is no implementation of any object or any method on the script level.
- **Project Utilities:** This package of the engine consists of all the classes and methods that have project-level utilities (project location, application environment, create folder, current date, future date, past date, etc.) that are required to build the automation framework across the project and develop test scripts for a specific application
 - **Assertion/Verification:** Consists of all the methods that will require verification points for asserting various test
 - **Common Methods:** Predefined methods that contain the list of Java-based methods available for the development of test scripts for a specific application
 - **Configuration Manager:** Class(es) and methods that will require configuration to different environments to build the automation framework across the project and develop test scripts for a specific application

The ATI Team is constantly working to enhance the core features of the developed framework. Updates will be released to the intended audience based on need.

3.4 Global Constants/Properties File

A global constant is one declared at the start or during execution of the code and is accessible to all parts of the program. In the developed automation framework, there is an option to access (set, asset and declare) global constants that are in context having global scope. Moreover, there is a list of global constants that are predefined in the “IResourceConstants.java” file. This gives the tester reference to constants that are application centric entities and are used for file paths.



A “.properties” file is a simple collection of key-value pairs that can be parsed by the Java properties class. Properties files are widely used for many purposes in all kinds of Java applications, often to store configuration or localization data. These files are not required to be complied during execution.

1. Developers/Testers are required to update the SAT and SIT properties file with the service account username and password. **Password must be same for all the users**. This file also contains application URL for the environment. The config files are located under the src/main/resources folder in the Eclipse project.

- user=service_account_username
user1= service_account_username _1
user2= service_account_username _2
password= service_account_password

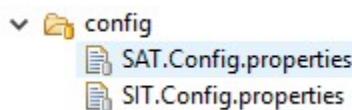


Figure 3 – Property File Location

2. The framework also supports the capability for parallel execution using Selenium Grid. This feature enables the reduction of execution timelines for test suites. By default this feature is disabled. To enable parallel execution, the following configuration variables will need to be updated:

- Browser – Browser preference for execution of suite. The following browsers are supported; default value set as “ie”:

Browser	Value
Internet Explorer	ie
Google Chrome	chrome
Mozilla Firefox	ff

- runOnGrid – Flag for execution of suite in grid mode; default value set as “no”
 - **Note:** The following batch file will need to be started to enable the Selenium Grid nodes for parallel execution, by default the number of parallel nodes is 10:
`C:\EclipseProject\<><project_name>\src\main\resources\driver\grid\Start_Grid.bat`
- hostip – ip address of host(s) running Selenium Grid
- port – port number of host(s) running Selenium Grid

3.5 Reporting Capabilities

Execution summary reporting is created after the overall execution of the automation suite.

A detailed log file is created in 2 formats *.log and *.html and can be retrieved from **<ProjectLocation>\logs locally**. Samples of the execution logs can be found in **Section 11. References**.

Screenshots are also captured on script failure. These screenshots are created as *.png file format and can be retrieved from **<ProjectLocation>\target\screenshot** on the user’s local machine.



4 Installation Instructions

This section contains the instructions for setting up the framework before it is used to start developing or executing the test scripts.

4.1 Pre-Requisites

Software (OSGetservice Request):

- Eclipse IDE Luna 4.4
- JDK 1.8.0.45
- Google Chrome
- Firefox
- Selenium – JAVA 3.8
- RTC Versioning Tool

OL5081 Access Requests:

- TEST USER IMRS (INTERNAL MAVEN REPOSITORY SYSTEM – IMRS)
- IT RATIONALHQJTSRW (JAZZ TEAM SERVER)

Configuration Files:

- Maven Settings XML

4.2 Installation Process

This section defines the instructions for configuring a local development environment to execute the developed test scripts.

1. Create folder ***C:\EclipseProject***
2. Launch Eclipse
3. Default workspace should be ***C:\EclipseProject***
4. Retrieve Latest Source Code (FATCA_QI_SELENIUM_TEST (PRGM_DEVELOPES_CLM_CM))
 - o Connect to RTC Source control repository from Jazz Administrator prospective
 - o Select <Project_Name>_SELENIUM_TEST-SC-comp

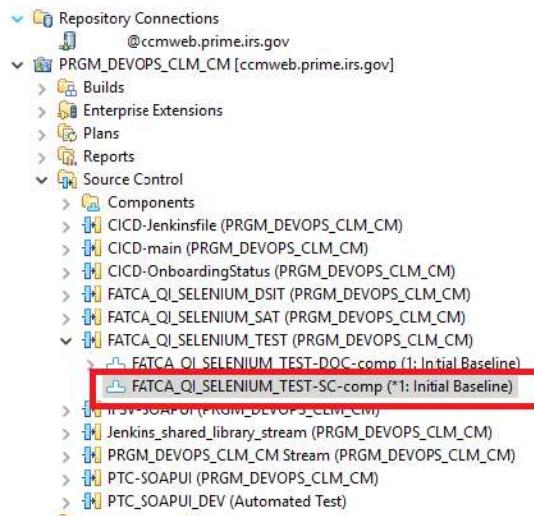


Figure 4 – RTC SCM FATCA QI

- Create a local repository and load (pull down) Selenium Project.
*Please refer to **Section 11. Reference - RTC SCM User Guide** for steps to create the local repository.*
- Open Java EE prospective
- Project should display at Project Explorer Tab

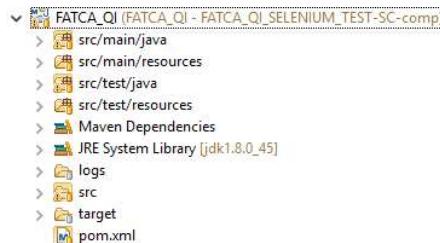


Figure 5 – RTC SCM Project Explorer



4.3 Project Layout (Folder Structure)

The image below provides the overall project hierarchy:

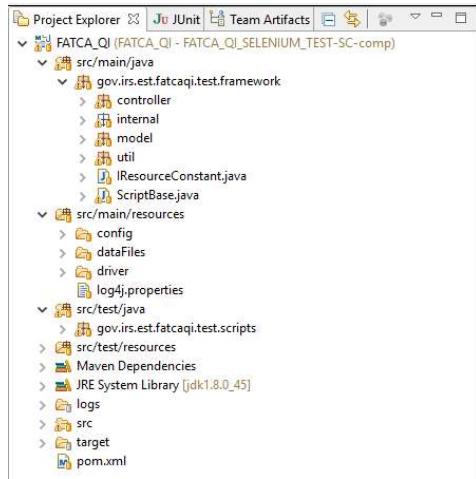


Figure 6 – Project Layout

4.4 Test Case Sheet

This section defines the spreadsheet the ATI Team will use to track the automation status of manual test cases. The template also serves as a mapping of developed test scripts to test cases. Developers and testers can use this spreadsheet to determine which test scripts need to be updated for each test case. A sample of this spreadsheet can be found in [Section 11. Reference](#).

5 Sample Script Creation

This section defines the steps required to create a new test script and upload the script to the RTC SCM.

5.1 Workflow

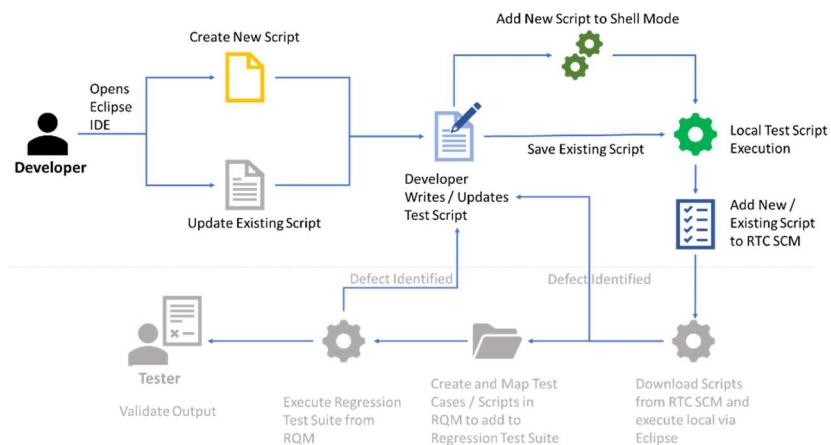


Figure 7 – Test Script Creation Workflow



1. Developer Opens Eclipse IDE and selects from the following two (2) options:
 - Create new script
 - Edit existing script
2. Developer creates/updates test script
3. Developer selects from the following two (2) options to save the test script:
 - If new test script is created, then the script needs to be added to shell mode
 - If script already exists, then save
4. Developer checks-in (adds) new/existing script to RTC SCM

5.2 Script Creation

1. Right-click on the package where the new script will be created
2. Select New -> Class

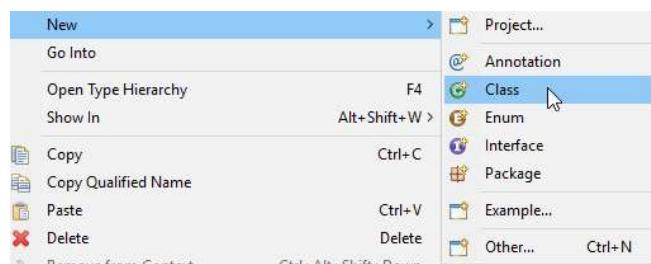


Figure 8 – New Class Creation

3. Enter the Class name
4. Click the Finish button

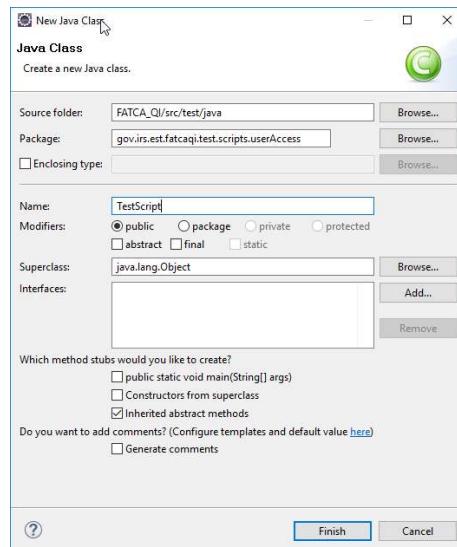
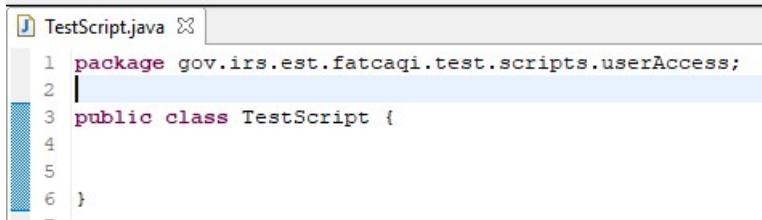


Figure 9 – Create New Class Completion

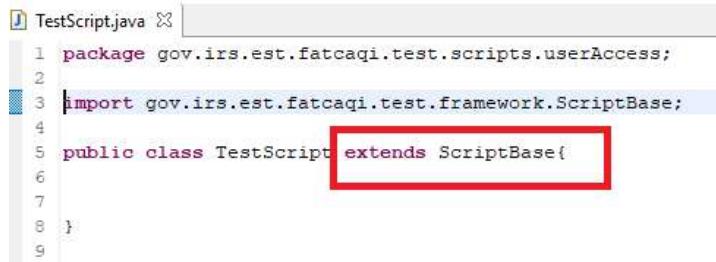
5. Ensure that the new Java class was created



```
TestScript.java
1 package gov.irs.est.fatcaqi.test.scripts.userAccess;
2
3 public class TestScript {
4
5 }
6
```

Figure 10 – Validate New Class

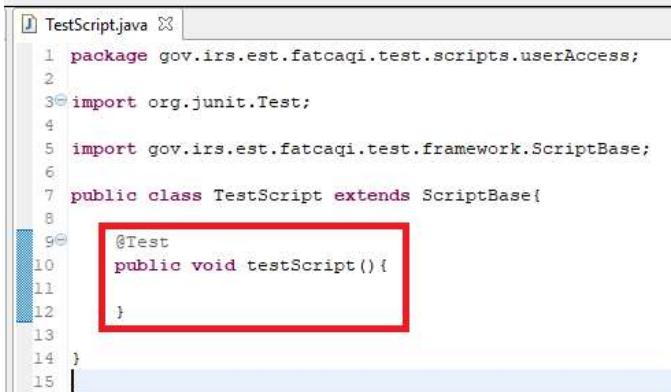
6. Extend the class to ScriptBase



```
TestScript.java
1 package gov.irs.est.fatcaqi.test.scripts.userAccess;
2
3 import gov.irs.est.fatcaqi.test.framework.ScriptBase;
4
5 public class TestScript extends ScriptBase{
6
7 }
8
9
```

Figure 11 – Extend Class to ScriptBase

7. Create the test Script method to the class and add JUnit annotation above the method



```
TestScript.java
1 package gov.irs.est.fatcaqi.test.scripts.userAccess;
2
3 import org.junit.Test;
4
5 import gov.irs.est.fatcaqi.test.framework.ScriptBase;
6
7 public class TestScript extends ScriptBase{
8
9     @Test
10    public void testScript(){
11
12    }
13
14 }
15
```

Figure 12 – Create Test Script Method

8. To create a new script, follow the pattern as <ProjectName.PageName.Action>.

i.e. fatca_qi().userLoginPage().enterUserName(User.USER_NAME);

6. Test Script Local Execution

This section will describe the three (3) options available to developers to execute the test cases.

6.1 Test Script Level

1. User should be able to select each script and run them individually
2. Right-click on the test script -> Run As -> JUnit Test



3. This will execute on the selected script

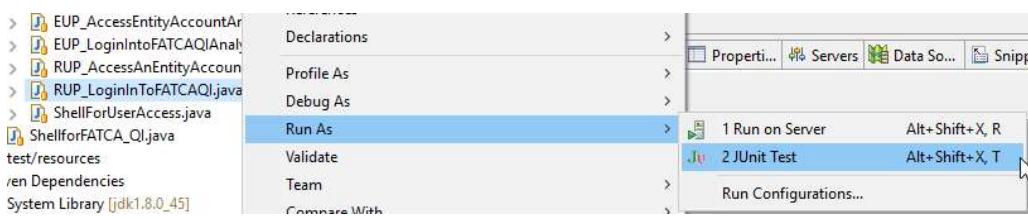


Figure 13 – Test Script Execution

6.2 Test Suite Level

1. User should be able to select each Test Suite script <ShellForUserAccess> run them on batch mode
2. Right-click on the test suite script-> Run As -> JUnit Test
3. This will only execute all the scripts within the Test Suite



Figure 14 – Test Suite Execution

6.3 Project Level

1. User should be able to select project Test Suite script <Shellfor[APPLICATION_NAME]> and run it on batch mode
2. Right-click on the project level test suite -> Run As -> JUnit Test
3. This will execute the scripts from the project



Figure 15 – Project Execution

7. Test and Validate Automation Script

7.1 Environment Configurations

This section defines the prerequisites required for a tester to execute the Selenium automation scripts using Eclipse and RQM.

7.1.1 Pre-Requisites:

ATI Testers will require the same access level as developers detailed in **Section 4.1 Pre-Requisites**. ATI Testers will also require the following roles/configuration updates:

- Testers will require **Release Engineer, Development Team Member** role in RTC
- Access to RQM Project Areas (i.e. PRGM_DEVOPS_CLM_QM)
 - Tester will require **Test Lead** role in associated project areas



- Single Sign On (SSO) configuration
 - Refer to Single Sign On (SSO) Procedures.pdf in **Section 11. References** for access to RQM project Area
- JUnit Selenium Adapter configuration
- RQM Selenium Adapter – Requested from Rational Support Team
- User needs to create the folder “**C:\DriverForProg**”
- User needs to create the following folders:
 - **C:\DriverForProg\RQMSeleniumAdapter**
 - **C:\DriverForProg\Log4J**
 - **C:\DriverForProg\POI3.7**
 - **C:\DriverForProg\SeleniumJAVA_3.8**
- RQM Selenium Adapter should be placed in **C:\DriverForProg\RQMSeleniumAdapter**
- Submit OSGetService installation request for log4j library files
- Log4j library files should be placed in **C:\DriverForProg\Log4J**
- Submit OSGetService installation request for Apache POI 3.7 library files
- Apache POI 3.7 library files should be placed in **C:\DriverForProg\POI3.7**
- User needs to request SeleniumJAVA_3.8 software from OSGetservice
- SeleniumJAVA_3.8 software should be placed in **C:\DriverForProg\SeleniumJAVA_3.8**

For questions or issues please contact it.devops.automated.testing@irs.gov

7.2 Retrieve Latest Test Scripts

Steps to configure a local project space and retrieve the latest automation code base can be found in section **4.2 Installation Process** and **3.4 Global Constants/Properties File**

7.2.1 Updating Properties File

1. Navigate to **C:\EclipseProject\FATCA_QI\src\main\resources\config**

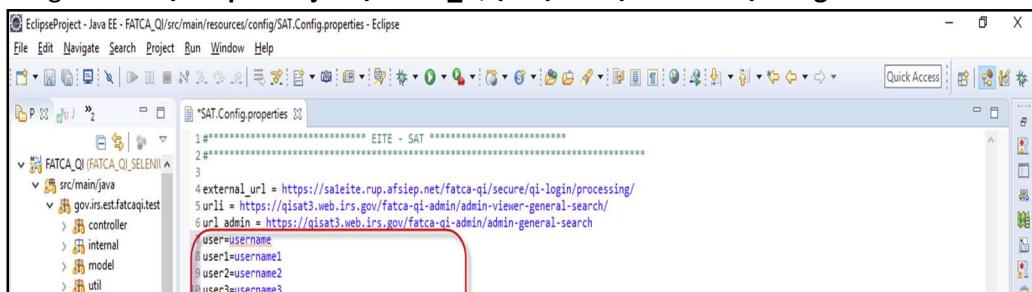


Figure 16 – SAT Config

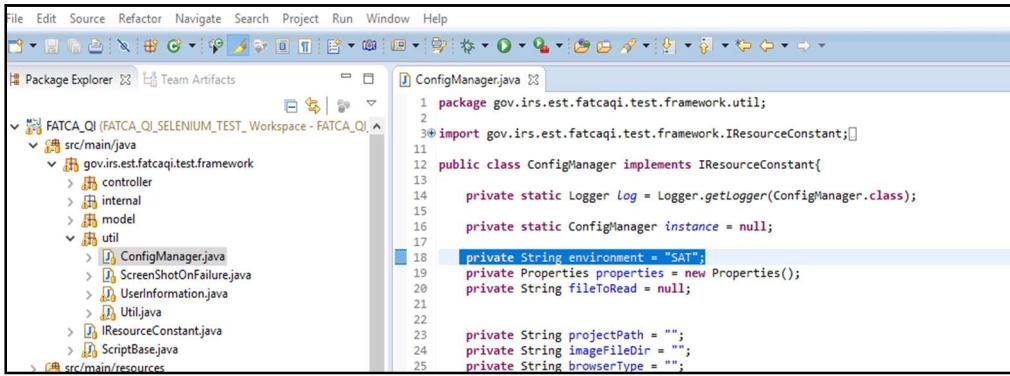
2. Select the corresponding environment properties files to update:
 - a. SAT.Config.properties for SAT (EITE) (**Default Environment**)
 - b. SIT.Config.properties for DSIT(SIT)
3. Update the following fields:
 - a. external_url - Provides the framework for the external url for the application being tested
 - b. url - Provides the framework for the internal url for the application being tested

- c. url admin - Provides the framework for the internal admin url for the application being tested
- d. userX – Provides the framework for the associated user test account login id. The framework currently supports up to ten (10) users.
- e. password – Provides the framework for the associated password for the test accounts
Note: All user accounts must have the same password
- f. browser – Provides the framework for the default browser to be used for test execution

Browser	Value
Internet Explorer	ie
Google Chrome	chrome
Mozilla Firefox	ff

Note: All browser values must be lowercase

4. Navigate to **C:\EclipseProject\FATCA_QI\src\main\java\gov\irs\est\fatcaqi\test\framework\util**
5. Select the ConfigManager.java file



```

File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Team Artifacts
FATCA_QI (FATCA_QI_SELENIUM_TEST_ Workspace - FATCA_QI)
src/main/java
  gov.irs.est.fatcaqi.test.framework
    controller
    internal
    model
    util
      ConfigManager.java
      ScreenShotOnFailure.java
      UserInformation.java
      Util.java
      ResourceConstant.java
      ScriptBase.java
src/main/resources
ConfigManager.java
1 package gov.irs.est.fatcaqi.test.framework.util;
2
3 import gov.irs.est.fatcaqi.test.framework.IResourceConstant;
4
5 public class ConfigManager implements IResourceConstant{
6
7     private static Logger log = Logger.getLogger(ConfigManager.class);
8
9     private static ConfigManager instance = null;
10
11     private String environment = "SAT";
12     private Properties properties = new Properties();
13     private String fileToRead = null;
14
15     private String projectPath = "";
16     private String imageFileDir = "";
17     private String browserType = "";
18
19
20
21
22
23
24
25

```

Figure 17 – ConfigManager

6. Update the private String environment variable based on the environment selection set in step 2

Environment	Value
SAT (EITE)	private String environment = "SAT"
DSIT (SIT)	private String environment = "SIT"

Note: Only one environment variable can be set per execution.

7.3 Test Script Execution

EST testers can execute the test script following the steps detailed in **Section 6.0 Test Script Local Execution**

7.4 Verifying the Logs

EST Testers can validate logs and screenshots following the steps detailed in **Section 3.5 Reporting Capabilities**

7.5 RQM Integration Process

The following section will detail how to integrate the developed test framework with RQM.

7.5.1 Setting Up RQM

1. Navigate to <https://rqmweb.prime.irs.gov:9443/jazz/web/console>



2. Select the RQM Project Area - *PRGM_DEVOPS_CLM_QM*
3. Create RQM Test Plan
 - a. Click “Create Test Plan” section under the Planning menu

The screenshot shows the 'Create Test Plan' dialog box. The 'Test Plan Name' field contains '<Enter New Test Plan Name>'. The 'Creating using template' dropdown is set to 'Default test plan template'. The 'State' dropdown is set to 'Draft'. The 'Action' dropdown is set to 'Change State'. The 'Originator' dropdown is set to 'Paudyal Ananda'. The 'Owner' dropdown is set to 'Unassigned'. The 'Priority' dropdown is set to 'Unassigned'. The 'Description' field has the placeholder text '<Click here to enter a description>'. The 'Summary' section includes an 'Overview of the test plan' and a 'Quality Task' section with icons. The 'Categories' section shows a single category.

Figure 18 – Create RQM Test Plan

- b. Enter the test plan name in the <Enter New Test Plan Name> field
- c. Click Save

The screenshot shows a list of test plans. One test plan is selected, showing its details: '16748: DevOps FATCA-QI FY 2019'. The 'State' is 'Draft', 'Action' is 'Change State', 'Originator' is 'Paudyal Ananda', and 'Owner' is 'Unassigned'. A progress bar indicates 'Total: 0:0 h', 'Estimated: 0%', and 'Progress: 0%'. The 'Test Case Execution (Record) Progress' section shows a progress bar at 0%.

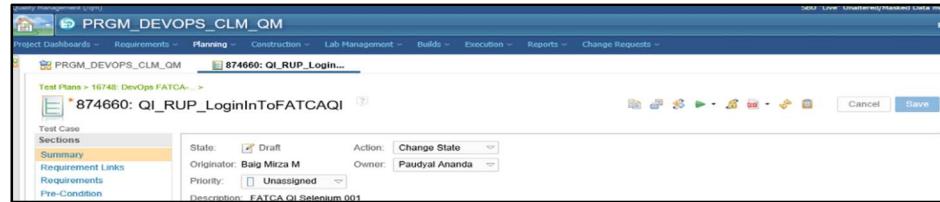
Figure 19 – RQM Test Plan Sample

4. Create RQM Test Case
 - a. If browser is closed, then
 - Navigate to <https://rwmweb.prime.irs.gov:9443/jazz/>
 - Navigate to the RQM Project Area *PRGM_DEVOPS_CLM_QM*
 - Click “Create Test Case” section under Construction menu
 - b. Enter the test case name in the <Enter New Test Case Name> field

The screenshot shows the 'Create Test Case' dialog box. The 'Test Case Name' field contains '<Enter New Test Case Name>'. The 'Creating using template' dropdown is set to 'Default test case template'. The 'State' dropdown is set to 'Draft'. The 'Action' dropdown is set to 'Change State'. The 'Originator' dropdown is set to 'Paudyal Ananda'. The 'Owner' dropdown is set to 'Unassigned'. The 'Description' field has the placeholder text '<Click here to enter a description>'.

Figure 20 – Create RQM Test Case

- c. Click Save



A screenshot of a software application window titled "PRGM_DEVOPS_CLM_QM". The main title bar also includes "Test Plans > 10740: DevOps FATCA..." and "874660: QI_RUP_Login...". The interface is a test case management tool with a navigation menu at the top and a central form for editing a test case. The form contains fields for State (set to Draft), Action (Change State dropdown), Originator (Baig Mirza M), Owner (Paudyal Ananda), Priority (Unassigned dropdown), and Description (FATCA QI Selenium 001). On the left, there is a sidebar with sections like Test Case, Sections, Summary, Requirement Links, Requirements, and Pre-Condition. A toolbar with various icons is located at the top right of the form area.

Figure 21 – RQM Test Case Sample



5. Create RQM Test Script

- If browser is closed, then
 - Navigate to <https://rwmweb.prime.irs.gov:9443/jazz/>
 - Navigate to the RQM Project Area **PRGM_DEVOPS_CLM_QM**
 - Click “Create Test Script” section under Construction menu
- Enter the test script name in the <Enter New Test Script Name> field

Figure 22 – Create RQM Test Script

- Select JUnit Selenium from the drop-down selection menu as the Type

Figure 23 – Selecting JUnit Adapter for Test Script

- Set Test Class to:
gov.irs.est.fatcaqi.test.scripts.userAccess.RUP_LoginInToFATCAQI
- Set Classpath to:
C:/EclipseProject/FATCA_QI/target/testclasses;C:/EclipseProject/FATCA_QI/target/classes
- Set Java System Properties to:
localDir=C:/EclipseProject/FATCA_QI
- Click Save

Figure 24 – RQM Test Script Sample

6. Link test script to test case

- If Browser is closed, then
 - Navigate to <https://rwmweb.prime.irs.gov:9443/jazz/>
 - Navigate to the RQM Project Area **PRGM_DEVOPS_CLM_QM**



- Otherwise, navigate to test case that created above steps and click Test Scripts section from left

The screenshot shows the RQM interface for a test case titled "874660: QI_RUP_LoginInToFATCAQI". The "Test Scripts" section is selected in the left sidebar. It displays a message: "View the automated or manual test scripts that are associated with this test case. Scripts can be reused. Note: you can drag test scripts from another browser to add them." Below this, there is a table header for "Test Scripts" with columns: ID, Name, Script Type, Owner, and Modified. A note below the table says "No items found." At the bottom right of the screen, there are "Add", "Add and Close", and "Close" buttons.

Figure 25 – RQM Test Case Sample

- Click **+** to identify the test scripts to be added to the test case
- Select the test scripts to be added
- Click “Add and Close” button

The screenshot shows a list of test scripts. The top bar includes the project name "263978", the test case name "QI_RUP_LoginInToFATCAQI_TS", and the status "Draft". The list area is empty, showing "Showing 1-1 of 1 items". At the bottom right, there are "Add", "Add and Close", and "Close" buttons.

Figure 26 – Selection RQM Test Script

- Click Save

The screenshot shows the RQM interface for the same test case. A green success message at the top states "Saved successfully at: 08:41:39". The "Test Scripts" section now lists one item: "QI_RUP_LoginInToFATCAQI_TS". The rest of the test case details remain the same as in Figure 25.

Figure 27 – Linking to RQM Test Script to RQM Test Case Sample

7. Create RQM Test Suite

- If Browser is closed, then
 - Navigate to <https://rwmweb.prime.irs.gov:9443/jazz/>
 - Navigate PRGM_DEVOPS_CLM_QM
 - Click “Create Test Suite” section from Construction menu
- Enter the test suite name in the <Enter New Test Suite Name> field



Figure 28 – Create RQM Test Suite

c. Click Save

Figure 29 – RQM Test Suite Sample

8. Add Test Case to Test Suite

a. If Browser is closed, then

- Navigate to <https://rwmweb.prime.irs.gov:9443/jazz/>
- Navagte to the RQM Project Area *PRGM_DEVOPS_CLM_QM*
- Navigate to Test Suite created in above step

Figure 30 – RQM Test Suite Sample

b. Click Test Cases in the left navigation pane

Figure 31 – Add RQM Test Case to RQM Test Suite



- c. Click + to identify the test cases to be added to the test suite
- d. Select the test cases to be added to the test suite
- e. Click “Add and Close”

The screenshot shows the RQM interface for a 'Test Suite' named '1092: FATCA QI User Access Test Suite'. The 'Test Cases' section is selected on the left sidebar. A single test case, '874660 GL_RUP_LoginToFATCAQI_TS', is listed with its ID and name. The main panel displays the test suite's configuration, including execution options like 'Run the suite in a sequence' and 'Run this suite in parallel'. The status bar at the bottom indicates 'Saved successfully at: 09:18:50'.

Figure 32 – Linking RQM Test Case to RQM Test Suite Sample

9. Link Test Suite with Test Plan

- a. If Browser is closed, then
 - Navigate to <https://rwmweb.prime.irs.gov:9443/jazz/>
 - Navigate to the RQM Project Area *PRGM_DEVOPS_CLM_QM*
- b. Navigate to Test Plan created in section 7.5.1 (step 3).

The screenshot shows the RQM interface for a 'Test Plan' named '16748: DevOps FATCA-QI FY 2019'. The 'Summary' section is selected on the left sidebar. The main panel displays the test plan's configuration, including its state (Draft), originator (Paudyal Ananda), and priority (Unassigned). It also shows the 'Test Case Execution (Record) Progress' and 'Test Suite Execution (Record) Progress' sections. The status bar at the bottom indicates 'The test suites ass... show details'.

Figure 33 – RQM Test Plan Sample



c. Click Test Suite

The screenshot shows the RQM Test Suite View for plan 16748: DevOps FATCA-QI FY 2019. The left sidebar has 'Test Suites' selected. The main area displays a table of test suites with columns: ID, Name, Created By, Weight, and Modified. A note at the bottom says 'Showing 1-13 of 13 items'.

ID	Name	Created By	Weight	Modified
952	FATCA_QI_Framework_2.0	Baig Mirza M	101	Aug 14, 2019
1004	FATCA_QI_Demo_With_Project_Team	Paudyal Ananda	4	Jun 4, 2019
1082	FATCA_QI_AccountAction Test Suite	Paudyal Ananda	6	Aug 7, 2019
1083	FATCA_QI_Admin Test Suite	Paudyal Ananda	14	Aug 14, 2019
1084	FATCA_QI_ApplyForApplication Test Suite	Paudyal Ananda	9	Aug 7, 2019
1085	FATCA_QI_CertificationCCG Test Suite	Paudyal Ananda	10	Aug 8, 2019
1086	FATCA_QI_CertificationOptions Test Suite	Paudyal Ananda	22	Aug 8, 2019
1087	FATCA_QI_ManageContactInformation Test Suite	Paudyal Ananda	9	Aug 7, 2019
1088	FATCA_QI_Renewal Test Suite	Paudyal Ananda	12	Aug 8, 2019
1089	FATCA_QI_TerminationOptions Test Suite	Paudyal Ananda	6	Aug 7, 2019
1090	FATCA_QI_UpdateAccountInformation Test Suite	Paudyal Ananda	3	Aug 8, 2019
1091	FATCA_QI_UpdatePAIAgreementInformation Test Suite	Paudyal Ananda	6	Aug 8, 2019
1092	FATCA_QI_User Access Test Suite	Paudyal Ananda	4	8 minutes ago

Figure 34 – RQM Test Suite View

- d. Click **+** to identify the test cases to be added to the test suite
- e. Select test suite to be added to test plan
- f. Click “Add and Close”

The screenshot shows the RQM Test Plan for plan 16748: DevOps FATCA-QI FY 2019. The left sidebar has 'Test Cases' selected. The main area displays a table of test cases with columns: ID, Name, Created By, Weight, and Modified. A note at the bottom says 'Showing 1-13 of 13 items'.

ID	Name	Created By	Weight	Modified
952	FATCA_QI_Framework_2.0	Baig Mirza M	101	Aug 14, 2019
1004	FATCA_QI_Demo_With_Project_Team	Paudyal Ananda	4	Jun 4, 2019
1082	FATCA_QI_AccountAction Test Suite	Paudyal Ananda	6	Aug 7, 2019
1083	FATCA_QI_Admin Test Suite	Paudyal Ananda	14	Aug 14, 2019
1084	FATCA_QI_ApplyForApplication Test Suite	Paudyal Ananda	9	Aug 7, 2019
1085	FATCA_QI_CertificationCCG Test Suite	Paudyal Ananda	10	Aug 8, 2019
1086	FATCA_QI_CertificationOptions Test Suite	Paudyal Ananda	22	Aug 8, 2019
1087	FATCA_QI_ManageContactInformation Test Suite	Paudyal Ananda	9	Aug 7, 2019
1088	FATCA_QI_Renewal Test Suite	Paudyal Ananda	12	Aug 8, 2019
1089	FATCA_QI_TerminationOptions Test Suite	Paudyal Ananda	6	Aug 7, 2019
1090	FATCA_QI_UpdateAccountInformation Test Suite	Paudyal Ananda	3	Aug 8, 2019
1091	FATCA_QI_UpdatePAIAgreementInformation Test Suite	Paudyal Ananda	6	Aug 8, 2019
1092	FATCA_QI_User Access Test Suite	Paudyal Ananda	4	8 minutes ago

Figure 35 – Linking RQM Test Suite within RQM Test Plan

After creating test cases in RQM, testers will need to link those test cases with Eclipse's JUnit test script

The screenshot shows the RQM Test Script View for test case 874660: QI_RUP_LoginInToFATCAQI. The left sidebar has 'Test Scripts' selected. The main area displays a table of test scripts with columns: ID, Name, Script Type, Owner, and Modified. A note at the bottom says 'Showing 1-1 of 1 items'.

ID	Name	Script Type	Owner	Modified
263978	QI_RUP_LoginInToFATCAQI_TS	JUnit Selenium	Baig Mirza M	Aug 7, 2019

Figure 36 – Mapping RQM Test Script to RQM Test Case

Note: Variables can be set by RQM within the Java System Properties field. For example, the browser type and environment variables can be set using the following:

localDir=C:/EclipseProject/FATCA_QI,BrowserType=chrome,ENV=SIT

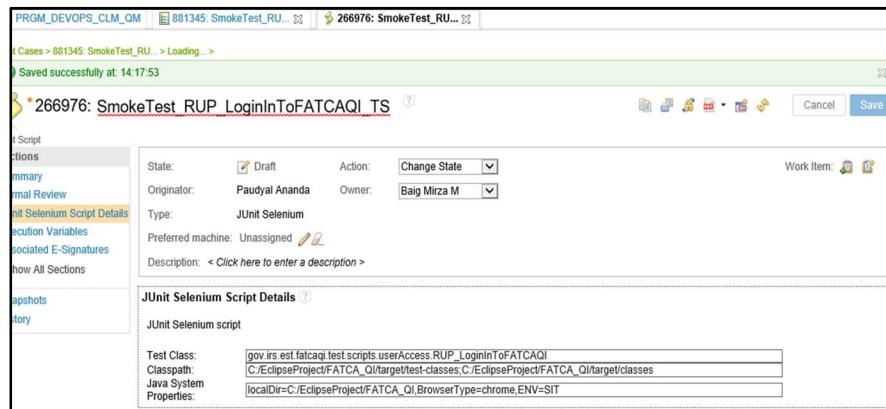


Figure 37 – Sample RQM Test Script

Note: There should not be any spaces between each of the variables set

7.5.2 Configuring JUnit Selenium Adapter

This section will detail how to configure the JUnit Selenium Adapter for execution.

1. Navigate to RQM Selenium Adapter V 6.0.600 folder under
C:\DriverForProg\RQM.Selenium.Adapter

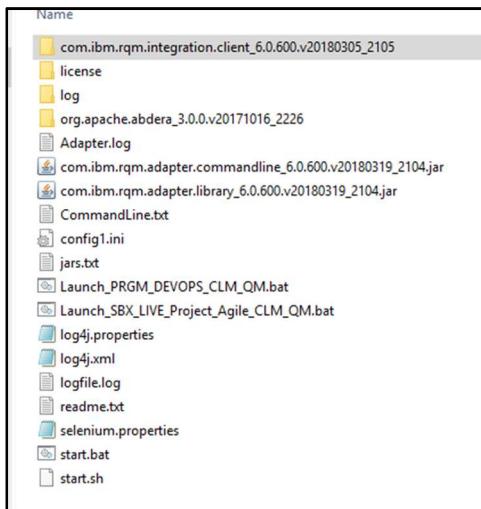


Figure 38 – JUnit Selenium Adapter Folder

2. Create a .bat file for each RQM project area
 - a. Navigate to the Selenium JUnit adapter folder where the JUnit adapter files were stored
(C:\DriverForProg\RQM.Selenium.Adapter)
 - b. Create a text file named “Launch_PRGM_DEVOPS_CLM_QM.bat” in the same folder as the Selenium JUnit adapter files

Name	Date modified	Type	Size
com.ibm.rqm.integration.client_6.0.600.v20180...	5/1/2019 3:05 PM	File folder	
license	5/1/2019 3:05 PM	File folder	
log	4/18/2019 1:08 PM	File folder	
logs	5/1/2019 3:05 PM	File folder	
org.apache.abdera_3.0.0.v20171016_2226	5/1/2019 3:05 PM	File folder	
Adapter.log	5/14/2019 7:45 AM	Text Document	65 KB
com.ibm.rqm.adapter.commandline_6.0.600.v2...	4/4/2018 4:33 PM	Executable Jar File	130 KB
com.ibm.rqm.adapter.library_6.0.600.v20180319...	4/4/2018 4:33 PM	Executable Jar File	122 KB
CommandLine.txt	11/19/2018 9:38 AM	Text Document	1 KB
config1.ini	4/18/2019 1:09 PM	Configuration settings	1 KB
config2.ini	5/14/2019 7:45 AM	Configuration settings	1 KB
jars.txt	4/4/2018 4:33 PM	Text Document	2 KB
Launch_PRMG_DEVOPS_CLM_QM.bat	4/18/2019 1:11 PM	Windows Batch File	1 KB
Launch_SBX_LIVE_Project_Agile_CLM_QM.bat	3/28/2019 3:19 PM	Windows Batch File	1 KB
log4j.properties	12/6/2018 4:33 PM	PROPERTIES File	2 KB
log4j.xml	11/13/2018 5:01 PM	XML File	1 KB
readme.txt	4/4/2018 4:33 PM	Text Document	10 KB
selenium.properties	5/1/2019 9:55 AM	PROPERTIES File	4 KB
start.bat	4/4/2018 4:33 PM	Windows Batch File	2 KB
start.sh	4/4/2018 4:33 PM	SH File	2 KB

Figure 39 – Creating Bat File in JUnit Selenium Folder

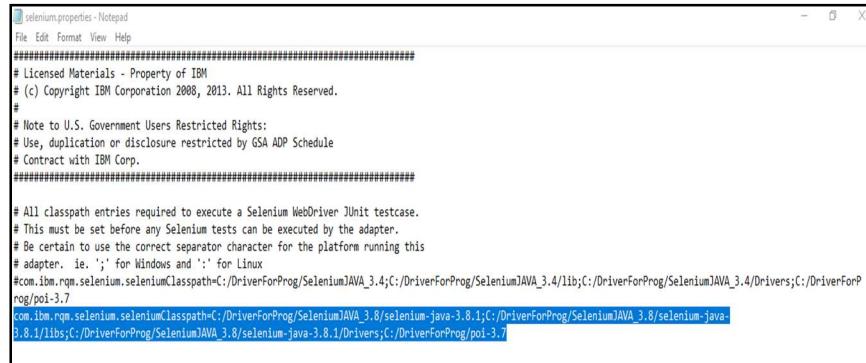
- c. Open the text file and paste the following:

```
set path=%PATH%;C:\Program Files\Java\jdk1.8.0_45\bin;
start.bat -repository https://rqmweb.prime.irs.gov:9443/jazz -user XXXXX -password
XXXXX -projectArea PRGM_DEVOPS_CLM_QM -adapterName XXXXXSeleniumAdapter
-configFile config2.ini
```

Note: Use your SEID as user and LAN credential as password

- d. Update the Selenium.properties file located **C:\DriverForProg\RQM.SeleniumAdapter\Selenium.properties** with the following content:

```
com.ibm.rqm.selenium.seleniumClasspath=C:/DriverForProg/SeleniumJAVA_3.8/selen
ium-java-3.8.1;C:/DriverForProg/SeleniumJAVA_3.8/selenium-java-
3.8.1/libs;C:/DriverForProg/SeleniumJAVA_3.8/selenium-java-
3.8.1/Drivers;C:/DriverForProg/poi-3.7
```



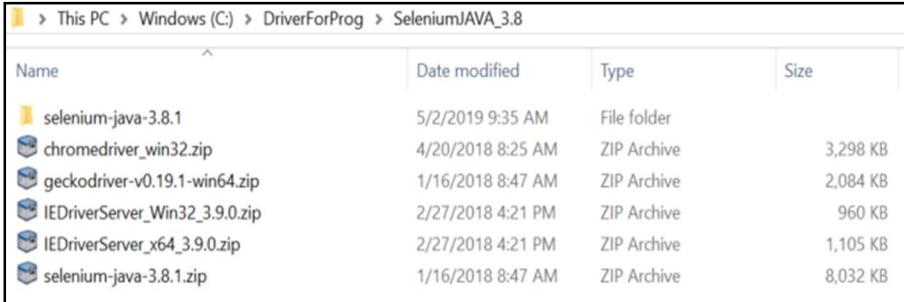
```
selenium.properties - Notepad
File Edit Format View Help
#####
# Licensed Materials - Property of IBM
# (c) Copyright IBM Corporation 2008, 2013. All Rights Reserved.
#
# Not to U.S. Government Users Restricted Rights:
# Use, duplication or disclosure restricted by GSA ADP Schedule
# Contract with IBM Corp.
#####

# All classpath entries required to execute a Selenium WebDriver JUnit testcase.
# This must be set before any Selenium tests can be executed by the adapter.
# Be certain to use the correct separator character for the platform running this
# adapter. ie. ';' for Windows and ':' for Linux
#com.ibm.selenium.seleniumClasspath=C:/DriverForProg/SeleniumJAVA_3.4;C:/DriverForProg/SeleniumJAVA_3.4/lib;C:/DriverForProg/SeleniumJAVA_3.4/Drivers;C:/DriverForProg/poi-3.7
com.ibm.selenium.seleniumClasspath=C:/DriverForProg/SeleniumJAVA_3.8/selenium-java-3.8.1;C:/DriverForProg/SeleniumJAVA_3.8/selenium-java-
3.8.1/libs;C:/DriverForProg/SeleniumJAVA_3.8/selenium-java-3.8.1/Drivers;C:/DriverForProg/poi-3.7
```

Figure 40 – Selenium Properties

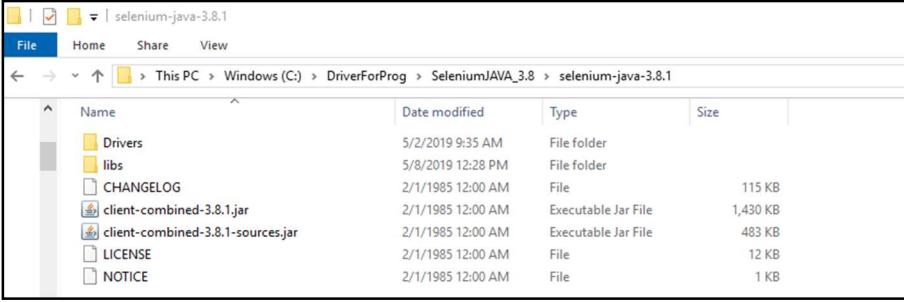
e. Verify folder contents

Note: All drivers should be placed in the C:\DriverForProg\SeleniumJAVA_3.8\ as seen below:



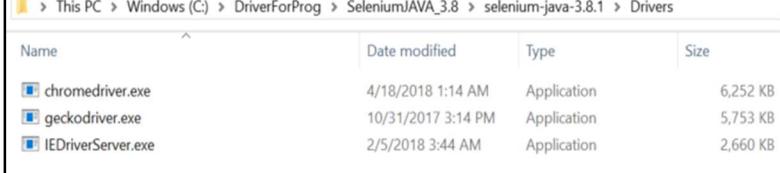
Name	Date modified	Type	Size
selenium-java-3.8.1	5/2/2019 9:35 AM	File folder	
chromedriver_win32.zip	4/20/2018 8:25 AM	ZIP Archive	3,298 KB
geckodriver-v0.19.1-win64.zip	1/16/2018 8:47 AM	ZIP Archive	2,084 KB
IEDriverServer_Win32_3.9.0.zip	2/27/2018 4:21 PM	ZIP Archive	960 KB
IEDriverServer_x64_3.9.0.zip	2/27/2018 4:21 PM	ZIP Archive	1,105 KB
selenium-java-3.8.1.zip	1/16/2018 8:47 AM	ZIP Archive	8,032 KB

Figure 41 – Selenium 3.8 Folder Structure



Name	Date modified	Type	Size
Drivers	5/2/2019 9:35 AM	File folder	
libs	5/8/2019 12:28 PM	File folder	
CHANGELOG	2/1/1985 12:00 AM	File	115 KB
client-combined-3.8.1.jar	2/1/1985 12:00 AM	Executable Jar File	1,430 KB
client-combined-3.8.1-sources.jar	2/1/1985 12:00 AM	Executable Jar File	483 KB
LICENSE	2/1/1985 12:00 AM	File	12 KB
NOTICE	2/1/1985 12:00 AM	File	1 KB

Figure 42 – Selenium 3.8.1 Java Folder Structure

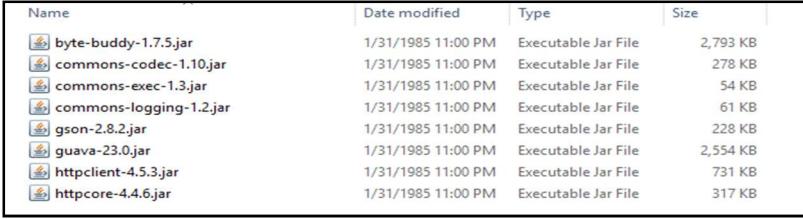


Name	Date modified	Type	Size
chromedriver.exe	4/18/2018 1:14 AM	Application	6,252 KB
geckodriver.exe	10/31/2017 3:14 PM	Application	5,753 KB
IEDriverServer.exe	2/5/2018 3:44 AM	Application	2,660 KB

Figure 43 – Selenium 3.8.1 Drivers Folder Structure

Note: The Drivers folder must contain the three (3) browser drivers as illustrated above.

Selenium Java 3.8 does not include all the library files to upload images into RQM during execution of automation scripts. Libs folders contains only libraries listed below.



Name	Date modified	Type	Size
byte-buddy-1.7.5.jar	1/31/1985 11:00 PM	Executable Jar File	2,793 KB
commons-codec-1.10.jar	1/31/1985 11:00 PM	Executable Jar File	278 KB
commons-exec-1.3.jar	1/31/1985 11:00 PM	Executable Jar File	54 KB
commons-logging-1.2.jar	1/31/1985 11:00 PM	Executable Jar File	61 KB
gson-2.8.2.jar	1/31/1985 11:00 PM	Executable Jar File	228 KB
guava-23.0.jar	1/31/1985 11:00 PM	Executable Jar File	2,554 KB
httpclient-4.5.3.jar	1/31/1985 11:00 PM	Executable Jar File	731 KB
httpcore-4.4.6.jar	1/31/1985 11:00 PM	Executable Jar File	317 KB

Figure 44 – Selenium 3.8 Library Within Libs Folder

The following jar files need to be added into the libs folder to allow for uploading into RQM:

- commons-io-2.5.jar
- commons-logging-1.1.1.jar
- hamcrest-core-1.3.jar
- hamcrest-library-1.3.jar
- JUnit-4.12.jar
- log4j-1.2.13.jar

Name	Date modified	Type	Size
byte-buddy-1.7.5.jar	2/1/1985 12:00 AM	Executable Jar File	2,793 KB
commons-codec-1.10.jar	2/1/1985 12:00 AM	Executable Jar File	278 KB
commons-exec-1.3.jar	2/1/1985 12:00 AM	Executable Jar File	54 KB
commons-io-2.5.jar	4/21/2017 1:01 PM	Executable Jar File	204 KB
commons-logging-1.1.1.jar	12/8/2015 11:14 AM	Executable Jar File	60 KB
commons-logging-1.2.jar	2/1/1985 12:00 AM	Executable Jar File	61 KB
json-2.8.2.jar	2/1/1985 12:00 AM	Executable Jar File	228 KB
guava-23.0.jar	2/1/1985 12:00 AM	Executable Jar File	2,554 KB
hamcrest-core-1.3.jar	4/21/2017 1:01 PM	Executable Jar File	44 KB
hamcrest-library-1.3.jar	3/10/2017 8:04 AM	Executable Jar File	52 KB
httpclient-4.5.3.jar	2/1/1985 12:00 AM	Executable Jar File	731 KB
httpcore-4.4.6.jar	2/1/1985 12:00 AM	Executable Jar File	317 KB
junit-4.12.jar	4/21/2017 1:01 PM	Executable Jar File	308 KB
log4j-1.2.13.jar	7/30/2015 11:23 AM	Executable Jar File	350 KB

Figure 45 – Complete Set of Selenium 3.8 Library Files

7.5.3 Starting JUnit Selenium Adapter

1. Navigate to the JUnit Selenium Folder C:\DriverForProg\RQMSeleniumAdapter\

com.ibm.rqm.integration.client_6.0.600.v20180305_2105
license
log
org.apache.abdera_3.0.0.v20171016_2226
Adapter.log
com.ibm.rqm.adapter.commandline_6.0.600.v20180319_2104.jar
com.ibm.rqm.adapter.library_6.0.600.v20180319_2104.jar
CommandLine.txt
config1.ini
jars.txt
Launch_PRGM_DEVOPS_CLM_QM.bat
Launch_SBX_LIVE_Project_Agile_CLM_QM.bat
log4j.properties
log4j.xml
logfile.log
readme.txt
selenium.properties
start.bat
start.sh

Figure 46 – JUnit Selenium Adapter Folder

2. Double-click on `Launch_PRCM_DEVOPS_CLM_QM.bat`

```
maven-3.3.9\bin;C:\Program Files (x86)\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn;;C:\Program Files (x86)\Microsoft SQL Server\130\Tools\Binn;;C:\Program Files (x86)\Microsoft SQL Server\130\Tools\ManagementStudio\;;C:\Program Files\Java\jdk1.8.0_45\bin;

C:\Users\q8nb\Desktop\ProgFolder\RQBNB_RQM Selenium Adapter>start.bat -repository https://rqmweb.prime.irs.gov:9443 -user RQ8NB -password DDeone22# -projectArea PRGM_DEVOPS_CLM_QM -adapterName RQ8NBseleniumAdapter -configFile config3.ini
Launching Adapter...

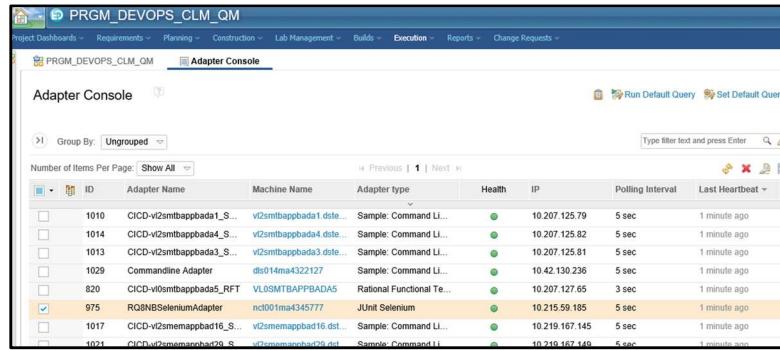
java version "1.8.0_111"
Java(TM) SE Runtime Environment (build 1.8.0_111-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.111-b14, mixed mode)
Adapter is reconnecting. Using saved configuration information from config3.ini
Attempting to create SSL_TLS context
java.security.NoSuchAlgorithmException: SSL_TLS SSLContext not available
Unable to create SSL_TLS context, trying TLS
Successfully created an HTTP client
Security consideration: Adapters enable you to run programs remotely. When you use adapters, ensure that you run them using a restricted user account that has access to only the required resources.
Using userid/password to authenticate.
Attempting to create SSL_TLS context
java.security.NoSuchAlgorithmException: SSL_TLS SSLContext not available
Unable to create SSL_TLS context, trying TLS
Authentication scheme Negotiate not supported
The adapter is now connected
"Created JUnit Selenium Adapter"
```

Figure 47 – Starting Adapter in CMD window

3. Verify Selenium adapter is connected to RQM

4. If browser is closed, then

- Navigate to <https://rqmweb.prime.irs.gov:9443/jazz/web/console>
- Select the RQM Project Area - *PRGM_DEVOPS_CLM_QM*
- Otherwise, navigate to RQM Adapter Console under the execution menu



The screenshot shows the RQM Adapter Console interface. At the top, there's a navigation bar with links like 'Project Dashboards', 'Requirements', 'Planning', 'Construction', 'Lab Management', 'Builds', 'Execution', 'Reports', and 'Change Requests'. Below that is a sub-navigation bar specific to the Adapter Console. The main area is a table titled 'Adapter Console' with columns: ID, Adapter Name, Machine Name, Adapter type, Health, IP, Polling Interval, and Last Heartbeat. There are 10 rows of data, each representing an adapter. Row 975, which is highlighted with a yellow background, corresponds to the 'RQ8NBseleniumAdapter' entry in the previous log output.

ID	Adapter Name	Machine Name	Adapter type	Health	IP	Polling Interval	Last Heartbeat
1010	CICD-v2smthappbad1_S...	v2smthappbad1.dte...	Sample: Command Li...	green	10.207.125.79	5 sec	1 minute ago
1014	CICD-v2smthappbad4_S...	v2smthappbad4.dte...	Sample: Command Li...	green	10.207.125.82	5 sec	1 minute ago
1013	CICD-v2smthappbad3_S...	v2smthappbad3.dte...	Sample: Command Li...	green	10.207.125.81	5 sec	1 minute ago
1029	Commandline Adapter	ds014ma432127	Sample: Command Li...	green	10.42.130.236	5 sec	1 minute ago
820	CICD-v0smthappbad5_RFT	VLOSMTBAPPBAD5	Rational Functional Te...	green	10.207.127.65	3 sec	1 minute ago
975	RQ8NBseleniumAdapter	nc001ma345777	JUnit Selenium	green	10.215.59.185	5 sec	1 minute ago
1017	CICD-v2simemappbad16_S...	v2simemappbad16.dte...	Sample: Command Li...	green	10.219.167.145	5 sec	1 minute ago
1021	CICD-v2smemappbad29_S...	v2smemappbad29.dte...	Sample: Command Li...	green	10.219.167.149	5 sec	1 minute ago

Figure 48 – Adapter Console

7.5.4 Executing Test Cases from RQM

1. If browser is closed, then

- Navigate to <https://rqmweb.prime.irs.gov:9443/jazz/web/console>
- Select the RQM Project Area - *PRGM_DEVOPS_CLM_QM*
- Otherwise, navigate to Construction menu and select RQM test case created in previous steps



The screenshot shows a test case record in RQM. The test case ID is 874660: QI_RUP_LoginInToFATCAQI. The summary section includes fields for State (Draft), Action (Change State), Originator (Baig Mirza M), Owner (Paudyal Ananda), Priority (Unassigned), and Description (FATCA QI Selenium 001). The Summary tab is selected, showing a table for Test Case Preparation where all items are marked as Unassigned. A weight field is set to 1. There are tabs for Test Case Sections, Requirements Links, Requirements, Pre-Condition, Expected Results, Notes, Test Scripts, Test Case Execution Records, Test Case Design, Execution Variables, Snapshots, and History.

Figure 49 – RQM Test Case Sample

2. Click on

The screenshot shows the same test case record in RQM, but the execution mode is active. The 'Run' button is highlighted in orange at the top right of the interface. The rest of the interface is identical to Figure 49, showing the test case details and preparation status.

Figure 50 – RQM Test Case Execution Mode

3. Click on Run

The screenshot shows the 'Run Test Case' dialog box. It starts with a step 1: Define Record Details. It offers two options: 'Use an Existing Test Case Execution Record' (selected) or 'Generate a New Test Case Execution Record'. Below this is a table for selecting items per page, showing one item selected: '816588 RUP_LoginIn...'. The 'Test Script' dropdown is set to 'Machine:•' and lists 'QI_RUP_LoginInToFATCAQI_TS' and 'VLOSMTBAPPBADA5'. The 'Test Cell' dropdown is set to 'CICD' and lists 'vlosmtbappbada5_SELENIUM'. The 'Build Record' and 'Deployment Plan' dropdowns are both set to 'Unassigned'. At the bottom, there are checkboxes for 'Schedule execution', 'Create a result without executing', and 'Modify execution variable values', along with 'Remember selected deployment plan for future run'. Navigation buttons 'Previous', 'Next', 'Finish', and 'Cancel' are at the bottom right.

Figure 51 – Run Test Case

4. Click on Machine to select your adapter



5. Select your adapter then click OK button

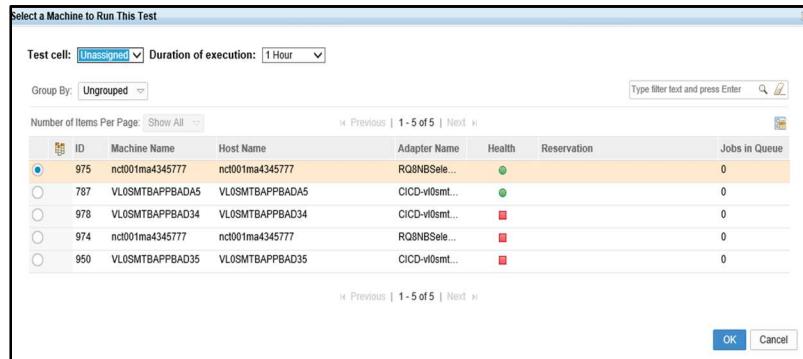


Figure 52 – Select a Machine to Run This Test

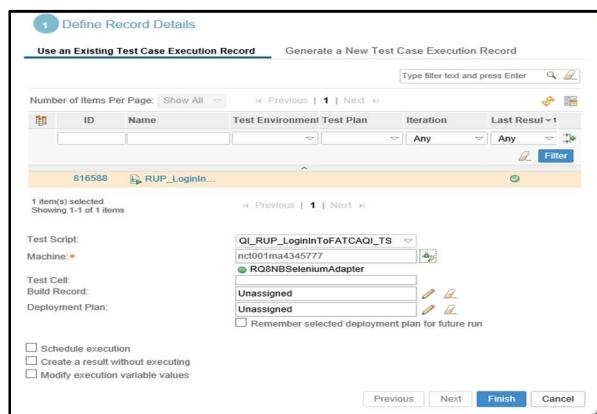


Figure 53 – Define Execution Record Detail

6. Click Finish

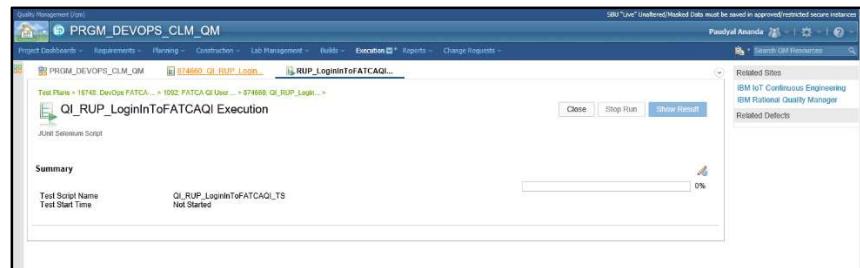


Figure 54 – Starting the Test

7. Automation will begin executing

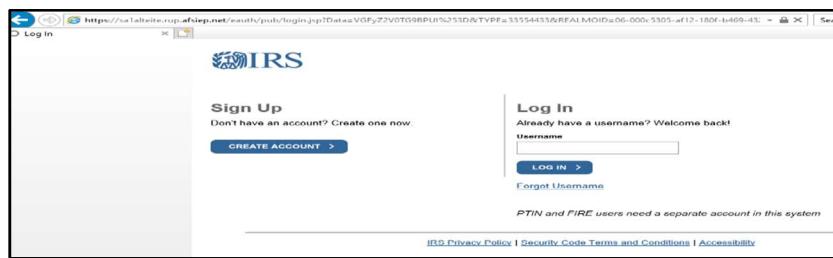


Figure 55 – Navigating FATCA QI Application

8. Wait for test execution to complete



9. Showing error message during execution

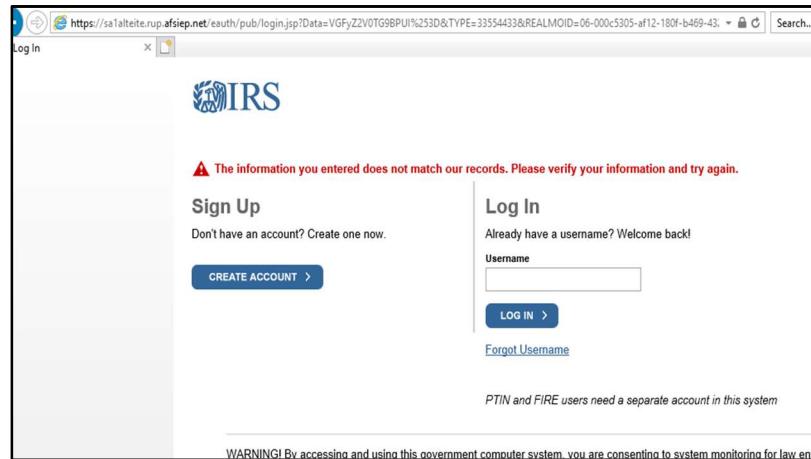


Figure 56 – Showing Error Message

10. Test Execution is complete and click on Show Result

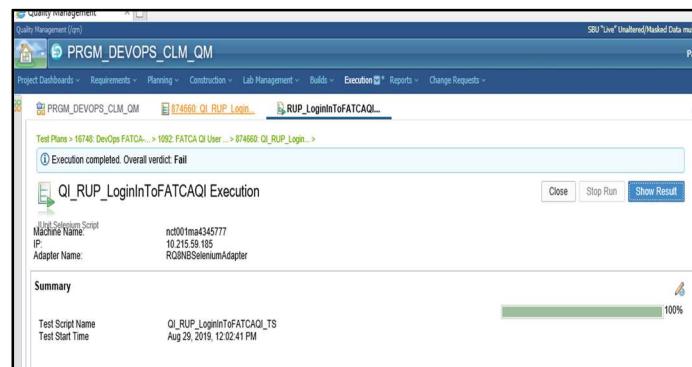


Figure 57 – Test Case Execution Status

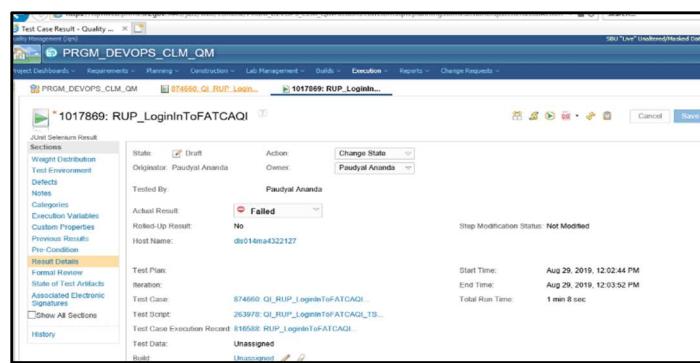
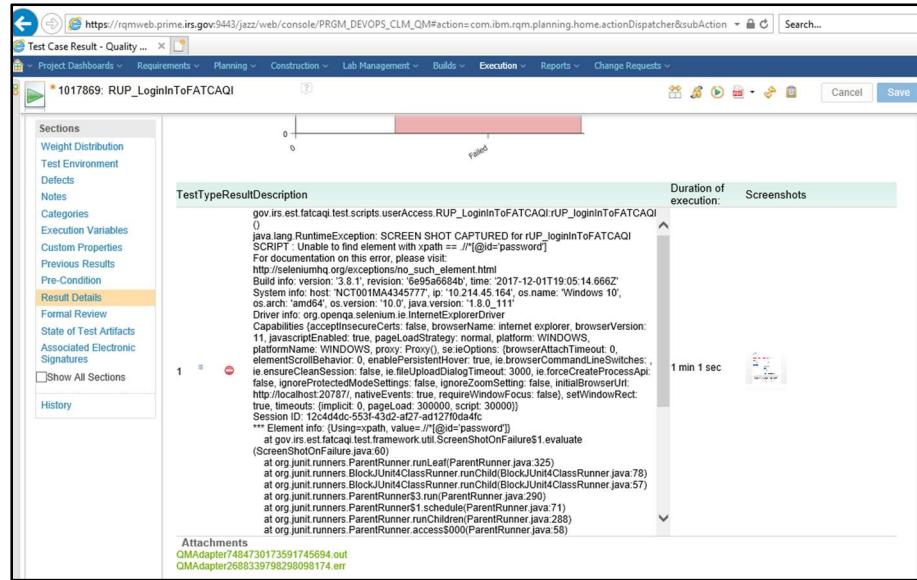


Figure 58 – Test Case Status

11. RQM is displaying a failed status for the test case

Note: Changes were made to the test script so that it will fail during execution. The error message and error image are then uploaded into RQM as an artifact.



The screenshot shows the RQM Test Case Result - Quality dashboard. A test case titled "1017869: RUP_LoginInToFATCAQI" has failed. The status bar at the top indicates "Failed". The "TestTypeResultDescription" section contains the following error message:

```

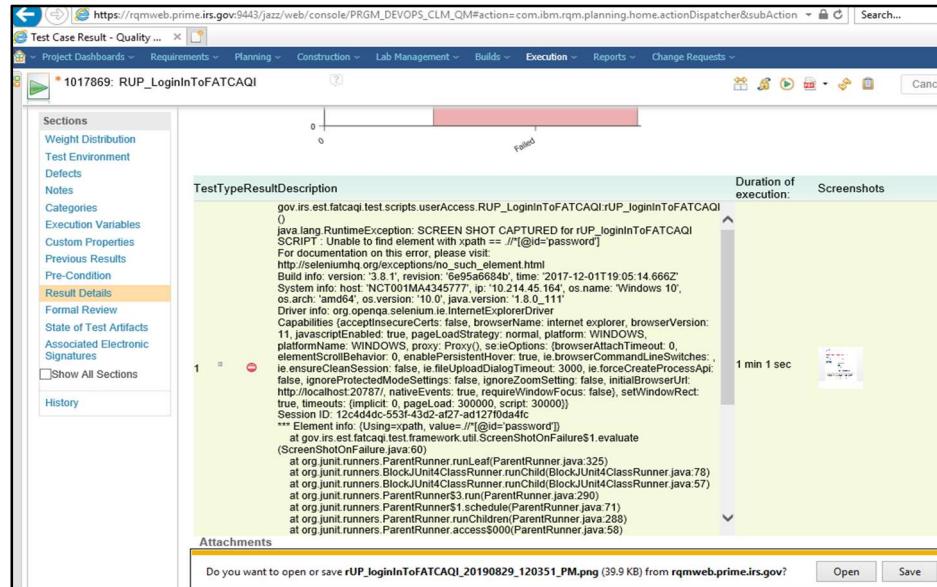
gov.irs.est.fatcaqi.test.scripts.userAccess.RUP_LoginInToFATCAQI.rUP_LoginInToFATCAQI
()
java.lang.RuntimeException: SCREEN SHOT CAPTURED for rUP_LoginInToFATCAQI
SCRIPT : Unable to find element with xpath == '/@id=password'
For documentation on this error, please visit:
http://seleniumhq.org/exceptions/no_such_element.html
Build info: version: '3.8.1', revision: '6955a664b', time: '2017-12-01T19:05:14.666Z'
System info: host: 'NCT001MA434577', ip: '10.214.45.164', os name: 'Windows 10',
os arch: 'amd64', java version: '10.0', java version: '1.8.0_111'
Driver info: org.openqa.selenium.ie.InternetExplorerDriver
Capabilities {acceptInsecureCerts: false, browserName: internet explorer, browserVersion:
11, javascriptEnabled: true, pageLoadStrategy: normal, platform: WINDOWS,
platformName: WINDOWS, proxy: Proxy{, seIeOptions: {browserAttachTimeout: 0,
elementScrollBehavior: 0, enablePersistentHover: true, forceCreateProcessSwitches:
false, ignoreProtectedModeSettings: false, ignoreZoomSetting: false, initialBrowserUrl:
http://localhost:20787/, nativeEvents: true, requireWindowFocus: false}, setWindowRect:
true, timeouts: {implicit: 0, pageLoad: 300000, script: 30000}}, Session ID: 12c4d40c-553f-43c2-a2f7-d1270d4fc
Session ID: 12c4d40c-553f-43c2-a2f7-d1270d4fc
*** Element info: (Using <xpath>, value: //[@id=password])
    at gov.irs.est.fatcaqi.test.framework.util.ScreenShotOnFailure$1.evaluate
(ScreenShotOnFailure.java:60)
    at org.junit.runners.ParentRunner.runLeaf(ParentRunner.java:325)
    at org.junit.runners.BlockJUnit4ClassRunner.$runChild(BlockJUnit4ClassRunner.java:78)
    at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:57)
    at org.junit.runners.ParentRunner$3.run(ParentRunner.java:290)
    at org.junit.runners.ParentRunner$3.schedule(ParentRunner.java:71)
    at org.junit.runners.ParentRunner.runChildren(ParentRunner.java:288)
    at org.junit.runners.ParentRunner.access$000(ParentRunner.java:58)

```

The "Attachments" section shows two files: QMAdapter7484730173591745694.out and QMAdapter2683330789280098174.png.

Figure 59 – Error Message for Fail Test Case

12. The test case has failed, and error message and error image are captured on the test script level within RQM. Double-click on the image screenshot then click open; it will display error during execution.



The screenshot shows the RQM Test Case Result - Quality dashboard. A test case titled "1017869: RUP_LoginInToFATCAQI" has failed. The status bar at the top indicates "Failed". The "TestTypeResultDescription" section contains the same error message as Figure 59. At the bottom of the "Attachments" section, there is a link: "Do you want to open or save rUP_LoginInToFATCAQI_20190829_120351_PM.png (39.9 KB) from rqmweb.prime.irs.gov?".

Figure 60 – Downloading PNG File

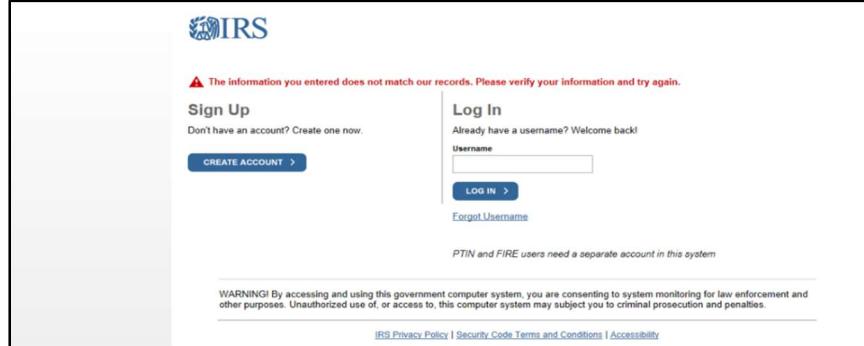


Figure 61 – Opening Error Image

13. Download QMAdapter7484730173591745694.out file and open in Notepad, you will see more detail of execution

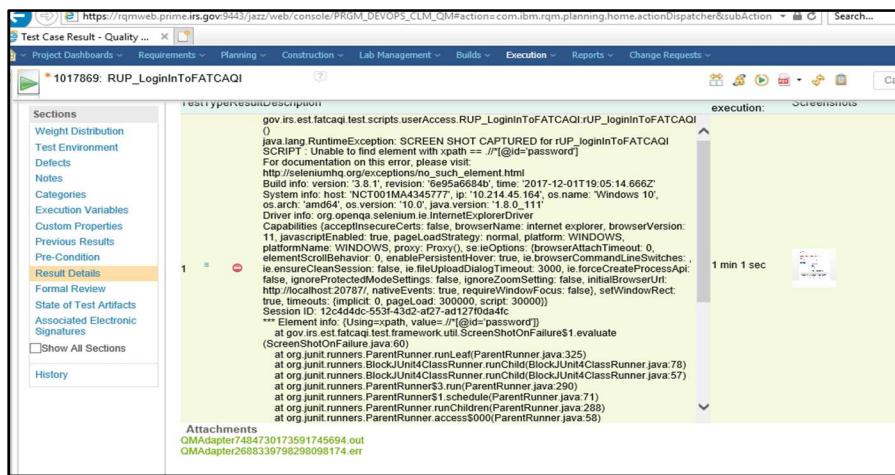


Figure 62 – Opening Out File

```
QMAdapter7484730173591745694.out
QMAdapter268833979829098174.err
```

File Edit Format View Help
2019-08-29 12:02:51.011 [INFO] [main] - gov.in.est.fatcaqi.test.framework.controller.ApplicationController
(ApplicationController.java:101) [main] - startApplication(FATCAQI)
2019-08-29 12:02:51.018 [INFO] [main] - gov.in.est.fatcaqi.test.framework.controller.ApplicationController
(ApplicationController.java:984) - startApplication(<https://sealeite.rup.afilis.net/fatca-qii/secure/qi-login/processing/>)
2019-08-29 12:03:32.592 [INFO] [main] - gov.in.est.fatcaqi.test.framework.controller.UserLoginController
(UserLoginController.java:109) [main] - startApplication(<http://localhost:20787/>, nativeEvents: true, requireWindowFocus: false), setWindowRect: true, timeout: 10000, (method: 0, pageLoadTimeout: 30000, script: 30000)
Session ID: 12c4d4dc-553f-43d2-af27-ad127ff0da4fc
*** Element info (Using<xpath>, value= //@id='password')
at gov.in.est.fatcaqi.test.framework.util.ScreenShotOnFailure\$1.evaluate
(ScreenShotOnFailure.java:69)
at org.junit.runners.BlockJUnit4ClassRunner.runLeaf(BlockJUnit4ClassRunner.java:325)
at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:78)
at org.junit.runners.BlockJUnit4ClassRunner.runChild(BlockJUnit4ClassRunner.java:57)
at org.junit.runners.ParentRunner\$3.run(ParentRunner.java:290)
at org.junit.runners.ParentRunner\$2.evaluate(ParentRunner.java:288)
at org.junit.runners.ParentRunner\$3.run(ParentRunner.java:71)
at org.junit.runners.ParentRunner.runChildren(ParentRunner.java:288)
at org.junit.runners.ParentRunner.access\$000(ParentRunner.java:58)

Figure 63 – Opening Out File in Text

7.5.5 Executing Test Suite from RQM

Prerequisite: JUnit Selenium adapter is up and running

1. If browser is closed, then
 - a. Navigate to <https://rqmweb.prime.irs.gov:9443/jazz/web/console>
 - b. Select the RQM Project Area - **PRGM_DEVOPS_CLM_QM**
 - c. Otherwise, navigate to Construction menu and select RQM Test Suite created in previous steps



The screenshot shows the RQM Test Suite interface. The main title bar says "Test Suite: 1092: FATCA QI User Access Test Suite". The left sidebar has sections like "Test Suite", "Sections", "Summary", "Test Suite Design", "Formal Review", "Pre-Condition", "Post-Condition", "Expected Results", "Risk Assessment", "Test Cases", "Test Suite Execution Records", "Attachments", "Execution Variables", "Show All Sections", "Manage Sections", "Snapshots", and "History". The "Test Cases" section is currently selected. The main content area shows the "Test Suite" configuration with fields for "State" (Draft), "Action" (Change State), "Originator" (Paudyal Ananda), "Owner" (Unassigned), "Priority" (Unassigned), "Description" (< Click here to enter a description >), and "Test Suite Execution (Record) Progress" (Total: 0/0 h, Estimated: 0%, Progress: 0%, Total: 0). Below this is the "Summary" section with fields for "Categories" (Unassigned), "Function" (Unassigned), "Test Phase" (Unassigned), "Estimate" (Weight: 4 Points), and "Quality Task" (with icons for edit, delete, and attach).

Figure 64 – RQM Test Suite Sample

2. Click on Test Case

This screenshot shows the "Test Cases" section of the RQM Test Suite. The left sidebar is identical to Figure 64. The main content area shows the "Test Cases" configuration with fields for "Run this suite in a sequence" (selected), "Pass execution variables between scripts" (unchecked), "Stop suite execution if any test case does not pass" (unchecked), and "Run this suite in parallel" (unchecked). The "Test cell Unassigned" section is shown. Below this is a table titled "Number of Items Per Page: Show All" with 4 items per page. The table lists four test cases: 874660 (ID 1) - QI_RUP_LoginInToFATCAQI_TS, 874661 (ID 2) - QI_EUP_AccessEntityAccountAnalyst_TS, 874662 (ID 3) - QI_EUP_LoginInfoFATCAQIAnalyst_TS, and 874664 (ID 4) - QI_RUP_AccessAnEntityAccount_TS.

Figure 65 – RQM Test Suite Showing Test Cases

3. Click on

This screenshot shows the RQM Test Suite interface with the "Test Suite" tab selected. The "Run Test Suite (Ctrl+Shift+X)" button is highlighted with a cursor. The rest of the interface is similar to Figure 64, showing the "Test Suite" configuration and summary.

Figure 66 – Executing Test Suite



4. Select test cases for execution

The screenshot shows the 'Define Record Details' dialog box. At the top, it lists 'Test Suite Execution Record' as 'New Test Suite Execution Record'. Below this, 'Test Plan', 'Iteration', 'Test Environment', and 'Test Cell' are all set to 'Unassigned'. A table below shows four test cases: 'OL_RUP_LoginInToFATCAQI' (Machine: nct001ma4345777), 'OL_EUP_AccessEntityAccount...' (Machine: VL0SMTBAPPBAD5), 'OL_EUP_LoginInfoFATCAQIAna...' (Machine: nct001ma4345777), and 'OL_RUP_AccessAnEntityAccount' (Machine: VL0SMTBAPPBAD5). The status column for all cases is green. At the bottom, there are sections for 'Build Record' (set to 'Unassigned'), scheduling (checkboxes for execution mode, parallel run, etc.), and execution status ('In Progress'). Buttons for 'Finish' and 'Cancel' are at the bottom right.

Figure 67 – Define Record Detail for Test Suite

5. Click Machine

The screenshot shows the 'Run Test Suite' dialog box. It has a sidebar with project navigation. The main area is titled 'Define Record Details' with 'Test Suite Execution Record' set to 'New Test Suite Execution Record'. Under 'Test Case', four checkboxes are checked: 'OL_RUP_LoginInToFATCAQI', 'OL_EUP_AccessEntityAccount...', 'OL_EUP_LoginInfoFATCAQIAna...', and 'OL_RUP_AccessAnEntityAccount'. The 'Machine' dropdown is set to 'VL0SMTBAPPBAD5'. Below this, a table shows the selected items: '4 items(s) Selected (Showing 1-4 of 4 items)'. The table includes columns for 'Test Case', 'Test Script', 'Machine', 'Test Cell', 'Health', and 'Skip Execution'. The 'Machine' column for all rows is highlighted in orange. At the bottom, there are sections for 'Build Record' (set to 'Unassigned'), scheduling, and execution status ('In Progress'). Buttons for 'Finish' and 'Cancel' are at the bottom right.

Figure 68 – Identifying an Adapter

6. Select JUnit Selenium Adapter

The screenshot shows the 'Select JUnit Selenium Adapter' dialog box. It displays two adapter options: '974 nct001ma4345777' and '950 VL0SMTBAPPBAD35'. Both have their 'Machine' dropdown set to 'RQ8NBSe...'. The status column for both entries is red. At the bottom, there are buttons for 'OK' and 'Cancel'.

Figure 69 – Selecting an Adapter



7. Click OK

Test Suite Execution Record: New Test Suite Execution Record
Test Plan: Unassigned
Iteration: Unassigned
Test Environment: Unassigned
Test Cell: Unassigned

Number of Items Per Page: Show All | Previous | 1 | Next | Type filter text and press Enter

Test Case	Test Script	Machine	Test Cell	Health	Skip Execution
QI_RUP_LoginInToFATCAQI	QI_RUP_LoginInToFATCAQI_T!	nct001ma4345777		●	
QI_EUP_AccessEntityAccount...	QI_EUP_AccessEntityAccountAv	nct001ma4345777		●	
QI_EUP_LoginInfoFATCAQIAvA...	QI_EUP_LoginInfoFATCAQIAvA	nct001ma4345777		●	
QI_RUP_AccessAnEntityAccount	QI_RUP_AccessAnEntityAccoun	nct001ma4345777		●	

4 item(s) selected
Showing 1-4 of 4 items

Build Record: Unassigned

Schedule execution:
Run this suite in parallel
Pass execution variables between scripts (sequential mode only)
Stop suite execution if any test case does not pass
Update test suite steps to include a valid script if one is needed
Create a result without executing
Modify execution variable values

8/29/2019 12:42:57 In Progress

Finish Cancel

Figure 70 – Adapter Selected

8. Click Finish

Test Suite Execution Record: FATCA QI User Access Test Suite

Execution Mode: Sequential
Test Start Time: Not Started
Test Cases Completed: 0
Test Cases Running: 0
Test Cases Not Run: 4

Execution Ord+ Name	Test Environment Test Script	Owner	Host Name	Health	Status	Test Case Progress	Result	Modified Test Start Time
1 RUP_LoginT...	QI_RUP...	Paudyal A	nct001...	●	Not Started	0%	QI...	1 mil

Showing 1-1 of 1 items

Figure 71 – Executing Test Cases with Test Suite

9. Verify Test Suite status in RQM

Test Suite Result: 1018016: X

Test Suite: * 1018016: FATCA QI User Access Test Suite

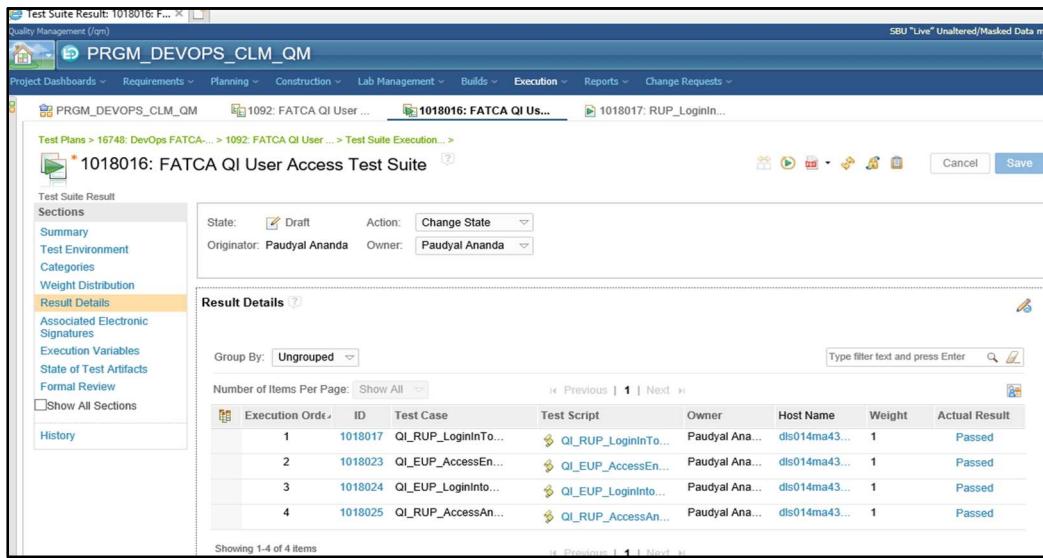
Test Suite Result Sections: Summary, Test Environment, Categories, Weight Distribution, Result Details, Associated Electronic Signatures, Execution Variables, State of Test Artifacts, Formal Review, Show All Sections, History.

Summary:

Actual Result: Passed
Start Time: Aug 29, 2019, 1:51:25 PM
End Time: Aug 29, 2019, 1:55:34 PM
Total Run Time: 4 min 9 sec
Rolled-Up Result: No
Execution Mode: Sequential
Execution Variables: Not passed between script(s)
Suite execution was stopped: No
Test Cell: Unassigned
Test Plan: Unassigned
Build: Unassigned
Iteration: Unassigned
Test Suite: FATCA QI User Access Test Suite
Test Suite Execution Record: FATCA QI User Access Test Suite

Figure 72 – Test Suite Execution Status

10. Verify test case status within Test Suite



The screenshot shows the Quality Management (qm) software interface. The main title is "Test Suite Result: 1018016: F...". The navigation bar includes "Project Dashboards", "Requirements", "Planning", "Construction", "Lab Management", "Builds", "Execution", "Reports", and "Change Requests". Below the navigation is a breadcrumb trail: "Test Plans > 16748: DevOps FATCA-... > 1092: FATCA QI User ... > Test Suite Execution... > 1018016: FATCA QI User Access Test Suite". The left sidebar has sections like "Test Suite Result Sections", "Summary", "Test Environment", "Categories", "Weight Distribution", "Result Details" (which is selected), "Associated Electronic Signatures", "Execution Variables", "State of Test Artifacts", "Formal Review", and "History". The main content area shows "Result Details" with a table:

Execution Order	ID	Test Case	Test Script	Owner	Host Name	Weight	Actual Result
1	1018017	QI_RUP_LoginInTo...	QI_RUP_LoginInTo...	Paudyal Ananda	dls014ma43...	1	Passed
2	1018023	QI_EUP_AccessEn...	QI_EUP_AccessEn...	Paudyal Ananda	dls014ma43...	1	Passed
3	1018024	QI_EUP_LoginInTo...	QI_EUP_LoginInTo...	Paudyal Ananda	dls014ma43...	1	Passed
4	1018025	QI_RUP_AccessAn...	QI_RUP_AccessAn...	Paudyal Ananda	dls014ma43...	1	Passed

Figure 73 – Showing Status of Each Test Case

7.5.6 Defect Reporting Process

If a script failed during execution via Eclipse, the AT Tester will communicate the defect to the AT Development Team via email. The image below describes the process workflow.

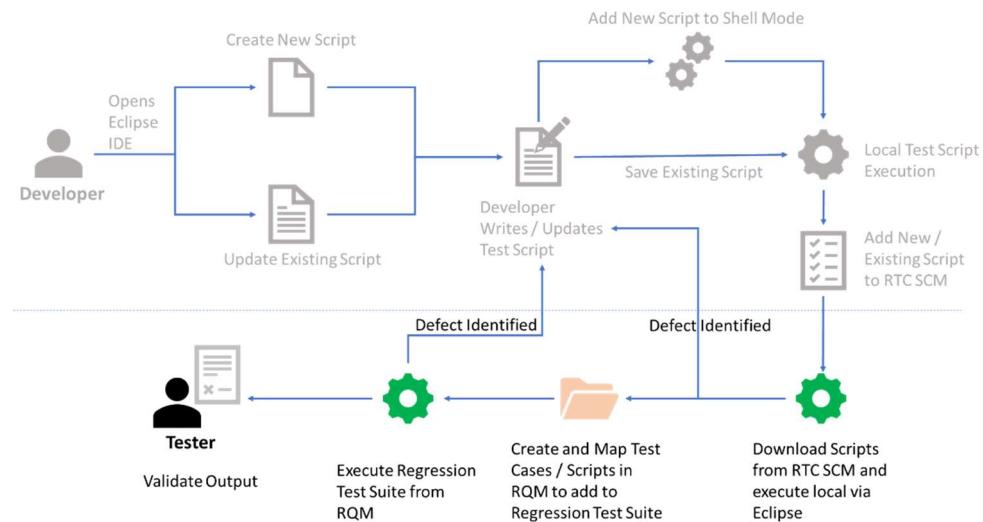


Figure 74 – Defect Reporting Process



8. CICD Handoff

This section contains the detailed steps to execute the developed automated test scripts in the CICD environment.

8.1 RQM Execution

8.1.1 Prerequisites

- User access to project area in RTC
- Application specific service accounts are created
- RQM service account with test lead role in the PRMG_DEVOPS_CLM_QM project area
- RTC service account with test lead role in the PRMG_DEVOPS_CLM_CM project area
- Compiled Source Code from Eclipse

8.1.2 Configuring Selenium JUnit Adapter

Section 7.5.2 Configuring Selenium JUnit Adapter provides an example of how to configure a Selenium JUnit Adapter. Please note that CICD has a process in place to configure a Selenium JUnit Adapter.

8.2 CICD Project Checklist

Once validation is complete the AT Team provides the CICD Team with a project specific checklist. This checklist serves as a validation milestone in the transition process. Content such as test accounts, source code location and number of test scripts are included in the checklist. For the full list of questions and sample content please reference Project Checklist **Section 11. References**

8.3 Service Account Creation Process

The service account creation process is considered out of scope for the ATI Team. A list of service accounts necessary to execute the automated test suite will be provided to the CICD Team on an application by application basis. For additional information on the service account creation process, please contact the DevOps CICD Team.

9. CICD Integration

This section outlines the steps on a high level and is not project specific. Depending on how each project is developed, additional steps might be needed.

9.1 Configuration Instructions

Configure the Jenkins Master - Windows workers setup to execute the Automation Test Scripts. They are set up in Jenkins master already in both CICD SBX and DEV environments.



MEM_BAD40	(offline)
MEM_BAD41	(offline)
MEM_BAD42	(offline)
MTB_BAD34	
1	Idle
2	Idle
3	Idle
4	Idle
MTB_BAD35	(offline)
MTB_BAD36	(offline)
MTB_BAD5	

Figure 75 – Jenkins Master - Worker Configuration

9.2 Installation Instructions

This section contains the instructions for setting up the framework before it is used to execute the test scripts

Software (KISAM CR)

- JDK 1.8.0.45
- Google Chrome
- Firefox
- RTC Versioning Tool
- RQMCommandLineAdapter-6.0.6
- RQMEexecutionTool_6.0.6
- RQMUrlUtility
- Junit Selenium Adapter

This section defines the instructions for configuring a Jenkins worker environment to execute the developed test scripts.

1. Retrieve Latest Source Code (FATCA_QI_SELENIUM_TEST (PRGM_DEVOPS_CLM_CM))



```
42
43
44 //node("windows" && "LIVE")
45 node(RUNNING_HOST)
46 {
47     def sAdapterID
48
49
50     stage("Pull data files")
51 {
52     RTC_source_pull(env.RTC_BUILD_TOOL_KIT, "CICD-FATCA_QI_HL2019_BD", "RTC_Shared_Library_SVC_ID",
53 }
54
55
56 stage("get RQM Adapter ID")
57 {
58     def sHostName
59
60     if(Jenkins.instance.getNode(NODE_NAME).toComputer().isUnix())
61 {
62         sHostName = sh(returnStdout: true, script: """ hostname """).replaceAll("\\..*", "").trim()
63     }
64     else
65 {
66         setPath()
67         sHostName = bat(returnStdout: true, script: """ hostname """).replaceAll("\\..*", "").trim()
68         sHostName = sHostName.toLowerCase()
69     }
70
71 }
```

Figure 76 – Fetching Source Code

2. Project folder should display under the build workspace Jenkins worker

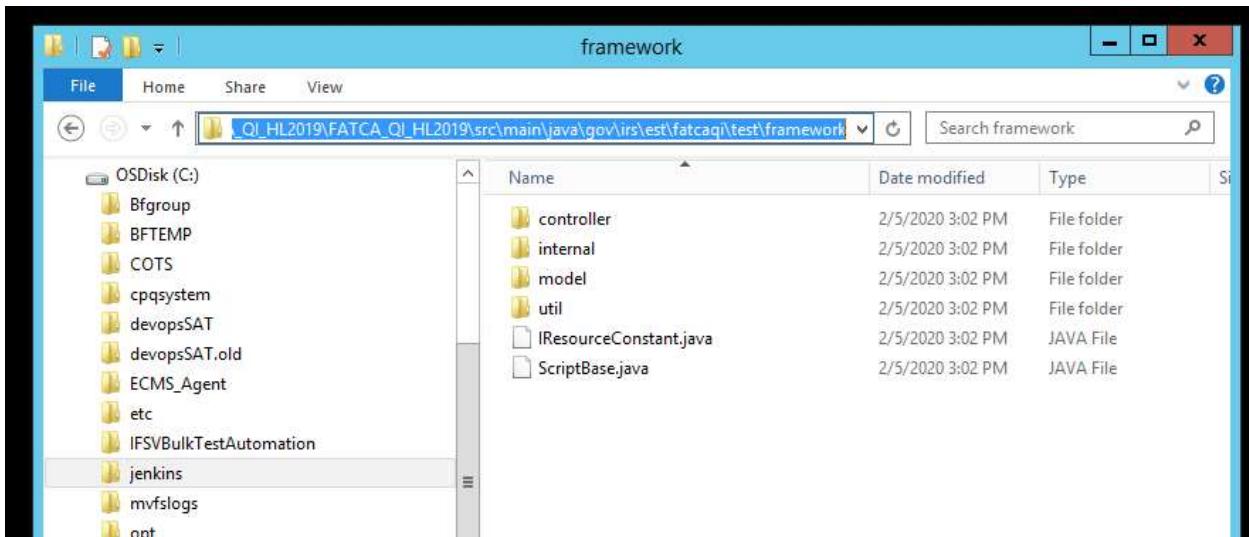


Figure 77 – Project Folder Structure



9.3 RQM Integration Process

For all of the below items, refer Section 7.5.1 -7.5.5

- Setting Up RQM
- Configuring Junit Selenium Adapter
- Starting Junit Selenium Adapter
- Executing Test Cases from RQM
- Executing Test Suite from RQM

9.4 RQM Command Line Execution Tool

```
> Version / location  
C:\opt\app\CICD\tools\RQM\RQMExecutionTool_6.0.6  
> Usage in Jenkins  
iRet = bat(returnStatus: true, script: """ java -jar  
"C:\\opt\\app\\CICD\\tools\\RQM\\RQMExecutionTool_6.0.6\\RQMExecutionTool.jar" -  
projectName="PRGM_DEVOPS_CLM_QM" -publicURI="https://rwmweb.prime.irs.gov:9443/jazz" -  
user="buildsrdstestsvc" -password=${sPassword} -tcerId=${TCERID} -scriptId="XXXX" -  
adapterId="${sAdapterID}" -exitOnComplete=true """)  
if(iRet != 20)  
> Usage in command line  
[buildsrdstestsv@vl2smtbappbada2 ~]$ java -jar  
C:\opt\app\CICD\tools\RQM\RQMExecutionTool_6.0.6/RQMExecutionTool.jar -  
projectName='<Project-Area>' -publicURI='https://rwmweb.prime.irs.gov:9443/jazz' -  
user=buildsrdstestsvc -password='<password>' -tserid=<Test-Suite-Execution-Record-ID> -  
adapterIdForAllSuiteStep=<Adapter-ID> -exitOnComplete=true  
=> <Project-Area>: RQM Project Area. "PRGM_DEVOPS_CLM_QM" is the EST team project area.  
=> <Test-Suite-Execution-Records-ID>: Should be defined in the RQM server. Artifacts (test results)  
will be attached under the TSERID. ** Please see RQM server section.  
=> <adapterIdForAllSuiteStep>: The Jenkins job gets the adapter ID from the RQM server using host  
name automatically so no need to find it manually. Only for the manual test, need to get the  
adapter ID from the RQM server. ** Please see RQM server section.
```

9.5 RTC Configuration

- Build Engine
 - 2 build engines are created for SBX and DEV Jenkins
 - CICD-SBX-_BE points SBX Jenkins and CICD-DEV_BE points DEV Jenkins
- Build Definition
 - 1 build definition is created for FATCA_QI_SELENIUM_TEST_BD
 - FATCA_QI_SELENIUM_TEST_BD points to FATCA QI repository workspace
- Stream
 - One stream is created ATCA_QI_SELENIUM_TEST, it has 2 components:
 - FATCA_QI_SELENIUM_TEST-DOC-comp
 - FATCA_QI_SELENIUM_TEST-SC-comp

Build domain user: CICD service domain account
- Workspace



1 Repository workspace is created
FATCA_QI_SELENIUM_TEST Workspace

10. Project Team Delivery and Acceptance

The ATI Team works closely with each application specific EST Test Team to improve knowledge transfer, reduce communication barriers and decrease the risks associated with transition once development and validation are complete. The process implemented by the ATI Team is a four (4) step approach:

1. **Automation Suite Demo** –The ATI Team provides a detailed walkthrough of the developed automated test suite. The following topics are discussed during transition:
 - Framework Documentation
 - Framework Capabilities
 - Configuration Management
 - Environment Configuration
 - Test Execution
2. **Test Case/Script Delivery** –The ATI Team deliver the completed test cases and scripts to the CICD Team and EST Test Team
3. **Application Team Acceptance** –The EST Test Team reviews the developed automated test suite and provides acceptance to the ATI Team
4. **Operation and Maintenance** –The ATI and the EST Teams discuss further enhancements and updates to the automated test suite

11. References

Click to access Document	Click to access Document	Click to access Document	Click to access Document
Maven IMR User Guide	RTC SCM User Guide	RTC SCM Access	Project Checklist (Sample)
The purpose of this document is to provide all Internal Maven Repository (IMR) users with reference instructions and documentation for the daily use of the Internal Maven Repository	The purpose of this document is to provide users with reference instructions and documentation for using the Rational Team Concert (RTC) Source Code Management (SCM) Tool	The purpose of this document is to provide users the steps necessary for accessing RTC SCM v6.0.4 repositories using the RTC clients	The purpose of this document is to provide users information on the application specific automation test scripts, RQM Test Cases and Scripts, Application account information and RTC tool

Click to access Document	Click to access Document	Click to access Document
Text File (Sample)	HTML File (Sample)	Application Initiative Tracker (Sample)
This file is a console output of detailed steps that are performed during execution from a script level	This file is a console output of detailed steps that are performed	The purpose of this document is to track the scope and development



	during execution from a script level in HTML format	status of the application currently in development
--	--	---



12. Version History

Version	Date	Author	Change Description
1.0	12/20/2019	AT Development and AT Test Teams	Initial Version
1.1	02/05/2020	DevOps CICD Team	Completed Integration section 9
1.2	03/05/2020	Jason Cleary (Communications Team)	Draft Review
1.3	03/18/2020	Ronald Shepherd/Ben Wright	Final Review