IRS

DevOps Practice, Automated Testing Initiative (ATI)
Automated Test Scripts Usage and Design Documentation
Version #: 1.3
Date: 03/18/2020

IRS

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

IRS

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

IRS

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.

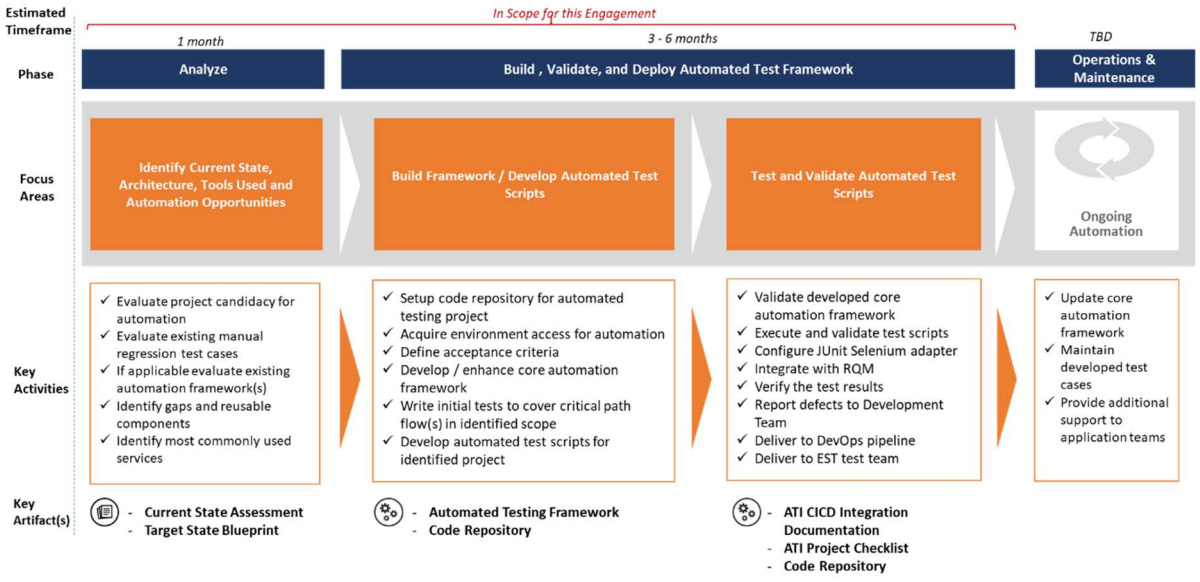


Figure 1 – Automated Testing Approach

- **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)

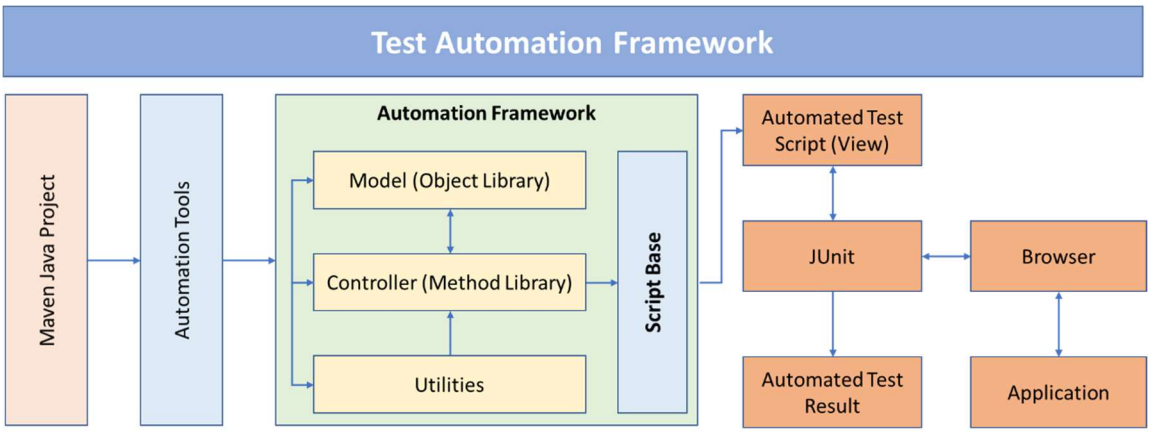


Figure 2 – Test Automation Framework

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

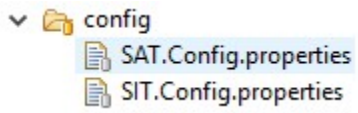


Figure 3 – Property File Location

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))
    - Connect to RTC Source control repository from Jazz Administrator prospective
    - Select <Project_Name>_SELENIUM_TEST-SC-comp

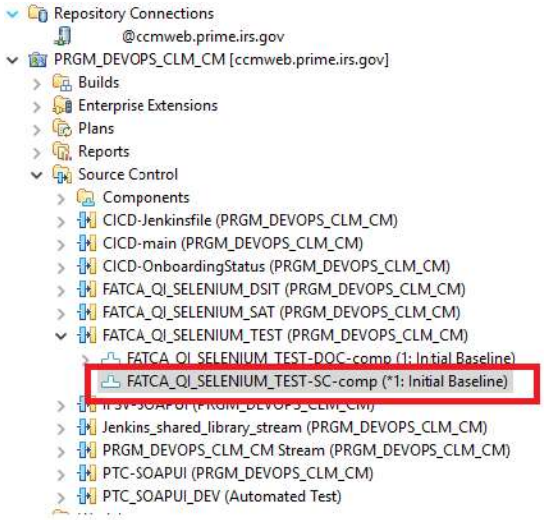


Figure 4 – RTC SCM FATCA QI

- 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

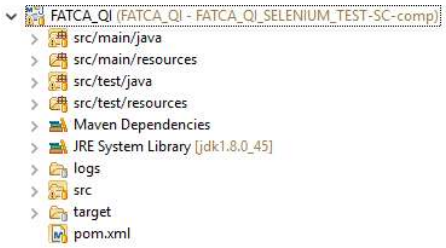


Figure 5 – RTC SCM Project Explorer

4.3 Project Layout (Folder Structure)

The image below provides the overall project hierarchy:

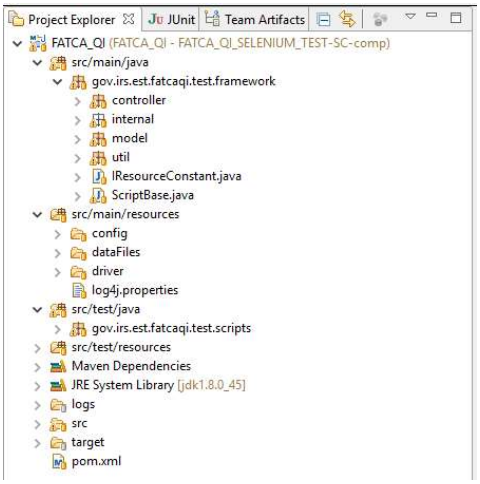


Figure 6 – Project Layout

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

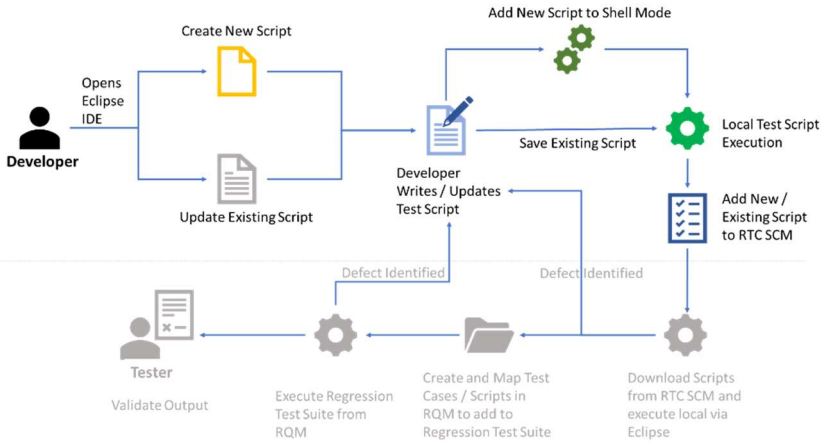


Figure 7 – Test Script Creation Workflow

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/exiting 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

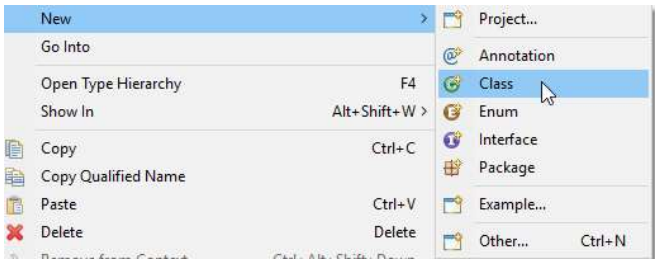


Figure 8 – New Class Creation

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

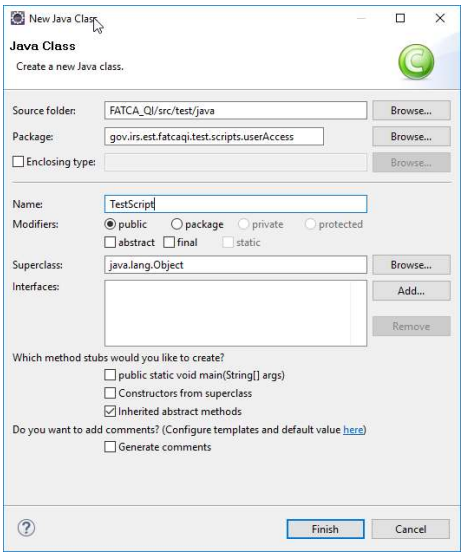


Figure 9 – Create New Class Completion

5. Ensure that the new Java class was created

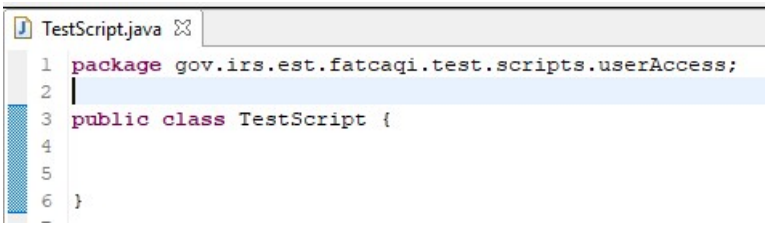


Figure 10 – Validate New Class

6. Extend the class to ScriptBase

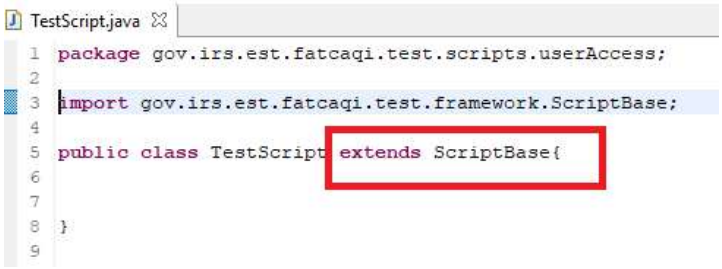


Figure 11 – Extend Class to ScriptBase

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

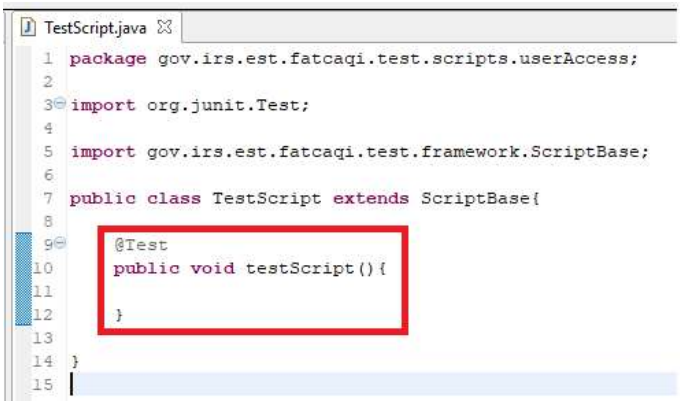


Figure 12 – Create Test Script Method

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

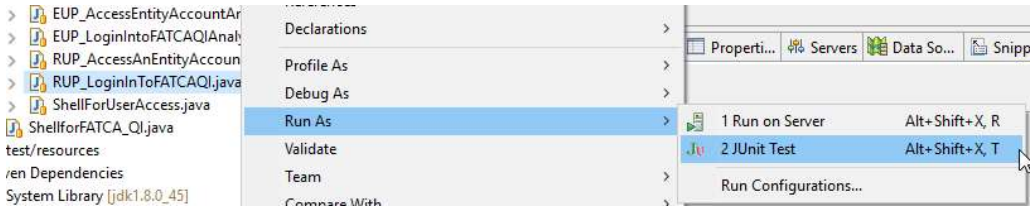


Figure 13 – Test Script Execution

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[APLICATION_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

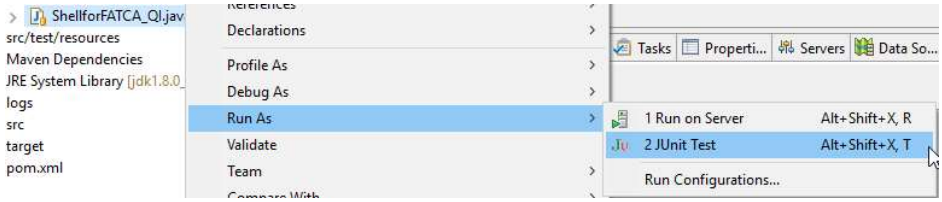


Figure 15 – Project Execution

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:\EclispseProject\FATCA_QI\src\main\resources\config**

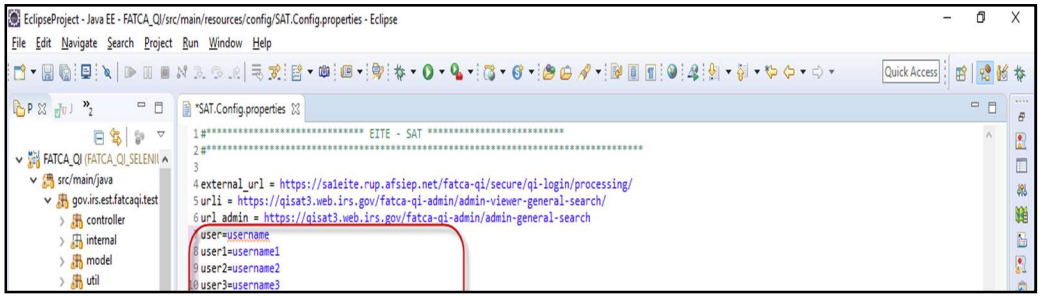


Figure 16 – SAT Config

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. urli - 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. user***X*** – 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

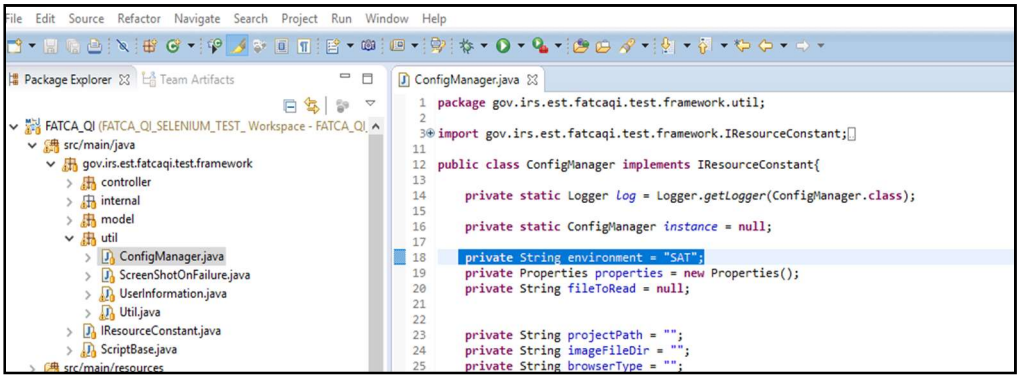


Figure 17 – ConfigManager

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

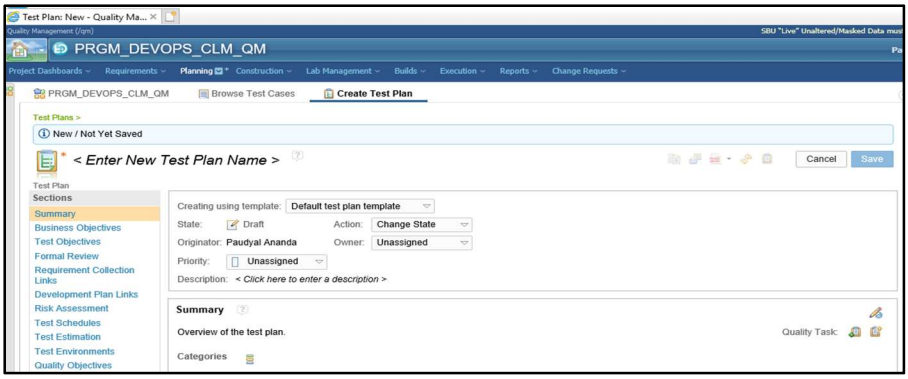


Figure 18 – Create RQM Test Plan

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

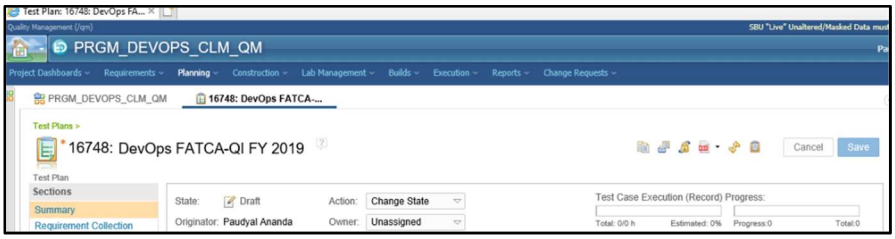


Figure 19 – RQM Test Plan Sample

4. Create RQM Test Case
    a. If browser is closed, then
        - Navigate to https://rqmweb.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

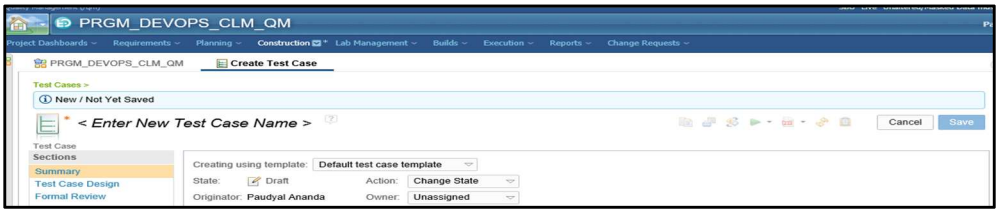


Figure 20 – Create RQM Test Case

    c. Click Save

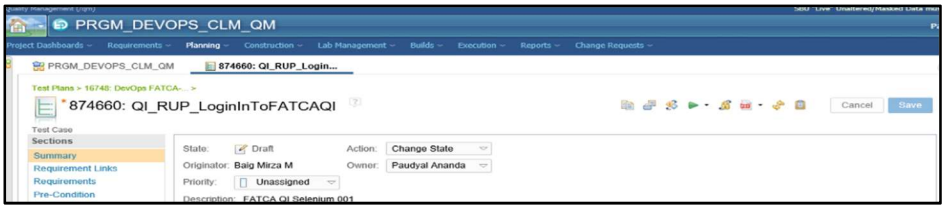


Figure 21 – RQM Test Case Sample

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

Figure 22 – Create RQM Test Script

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

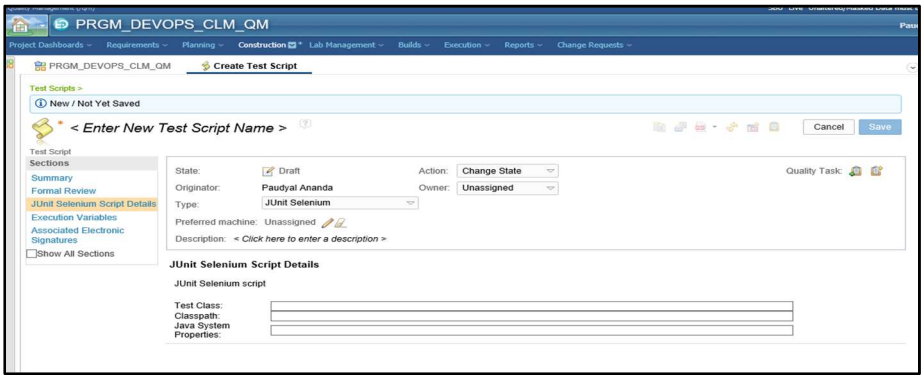


Figure 23 – Selecting JUnit Adapter for Test Script

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

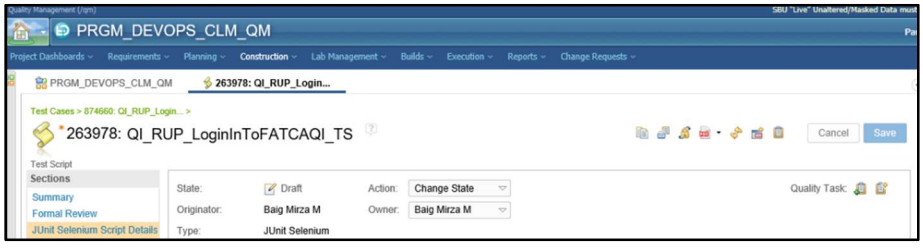


Figure 24 – RQM Test Script Sample

6. Link test script to test case
    a. If Browser is closed, then
        - Navigate to https://rqmweb.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

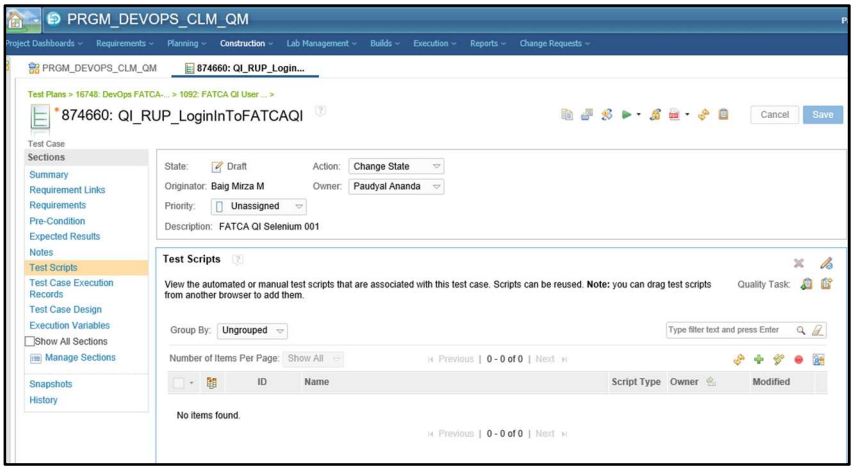


Figure 25 – RQM Test Case Sample

b. Click **+** to identify the test scripts to be added to the test case
c. Select the test scripts to be added
d. Click “Add and Close” button

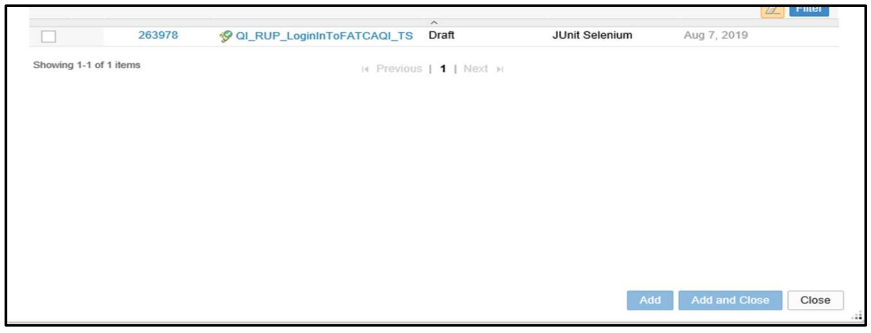


Figure 26 – Selection RQM Test Script

e. Click Save

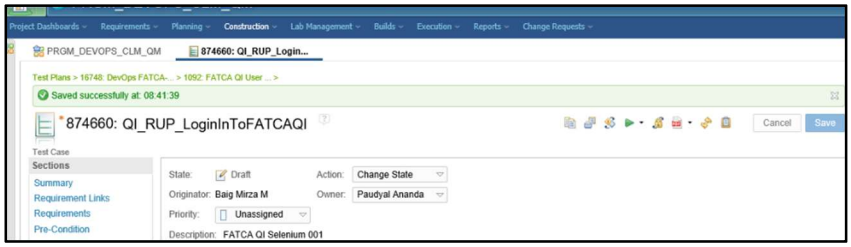


Figure 27 – Linking to RQM Test Script to RQM Test Case Sample

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

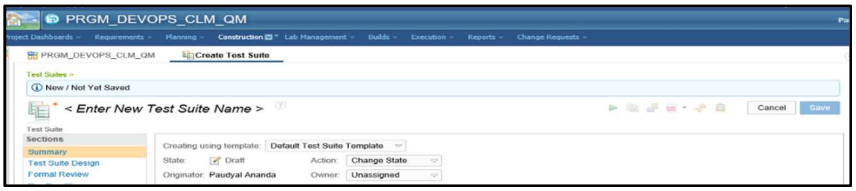


Figure 28 – Create RQM Test Suite

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

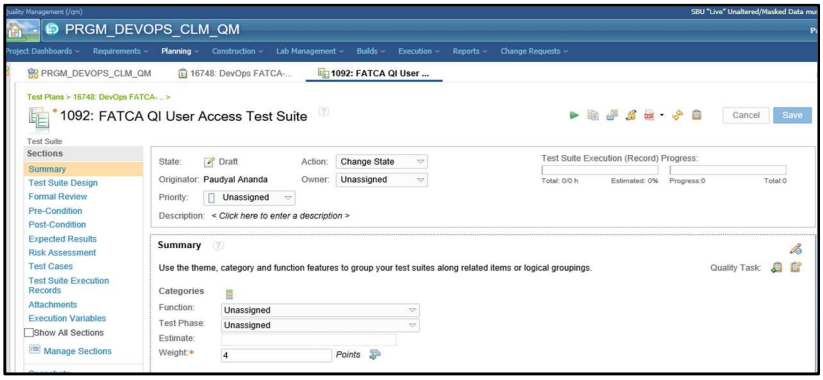


Figure 30 – RQM Test Suite Sample

b. Click Test Cases in the left navigation pane

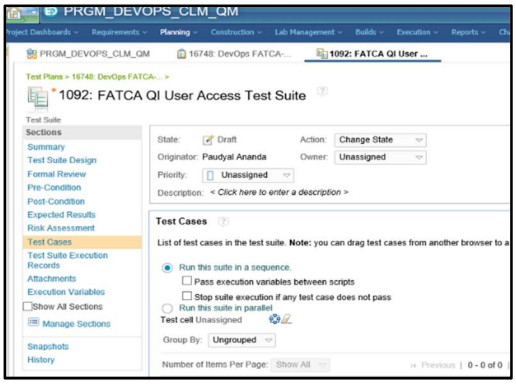


Figure 31 – Add RQM Test Case to RQM Test Suite

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”

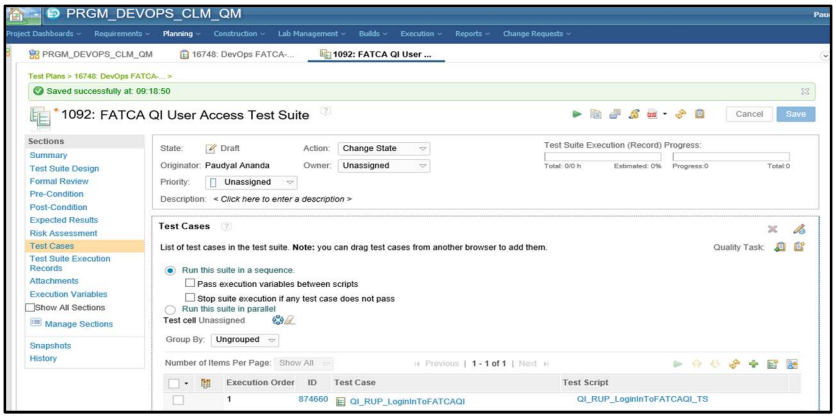


Figure 32 – Linking RQM Test Case to RQM Test Suite Sample

9. Link Test Suite with Test Plan

a. If Browser is closed, then
   - Navigate to https://rqmweb.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).

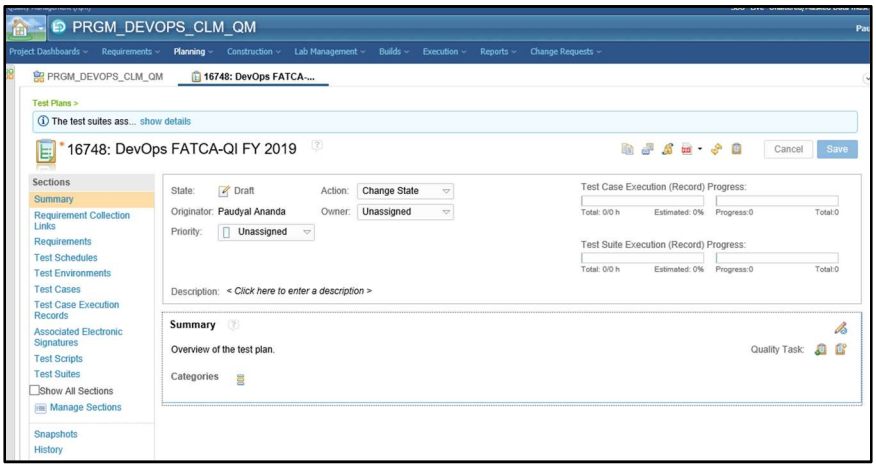


Figure 33 – RQM Test Plan Sample

c. Click Test Suite

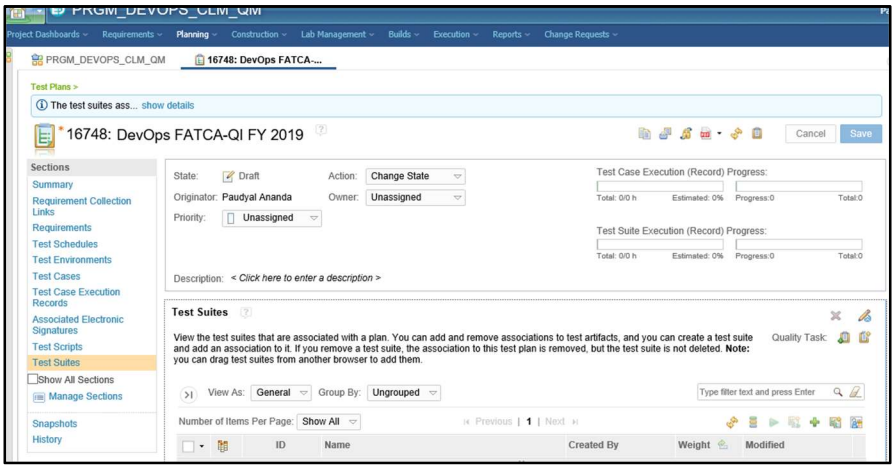


Figure 34 – RQM Test Suite View

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"

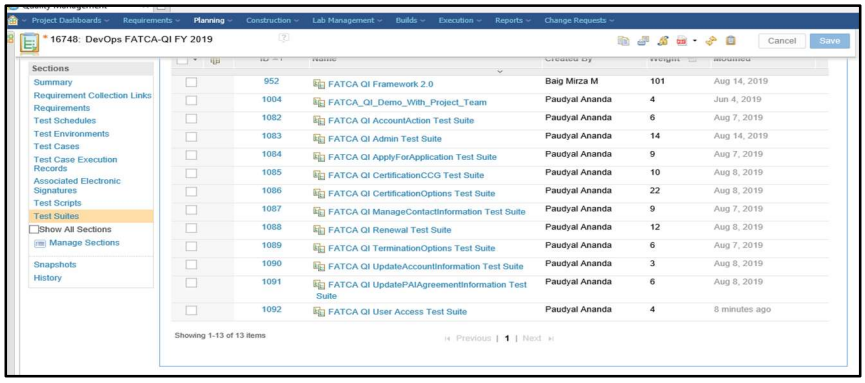


Figure 35 – Linking RQM Test Suite within RQM Test Plan

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

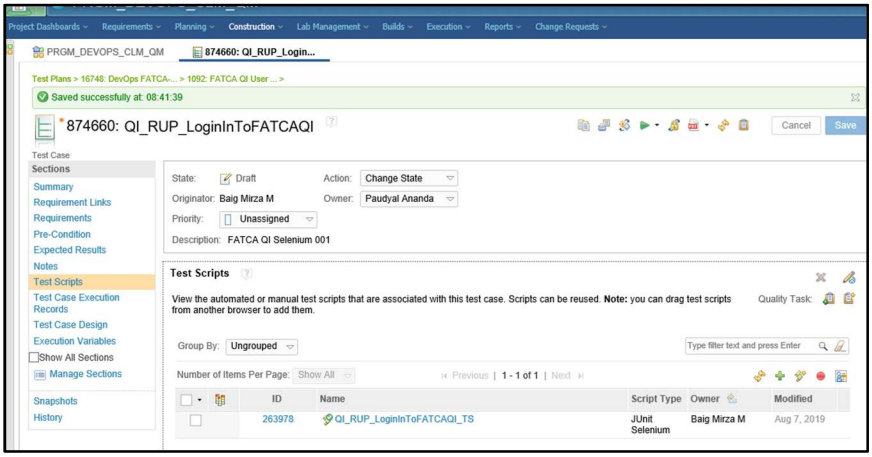


Figure 36 – Mapping RQM Test Script to RQM Test Case

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\RQMSeleniumAdapter***

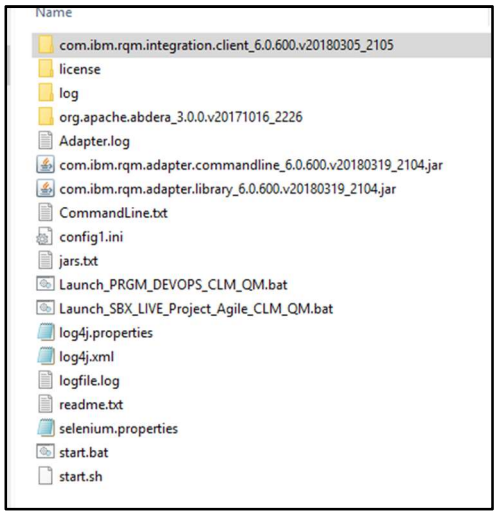
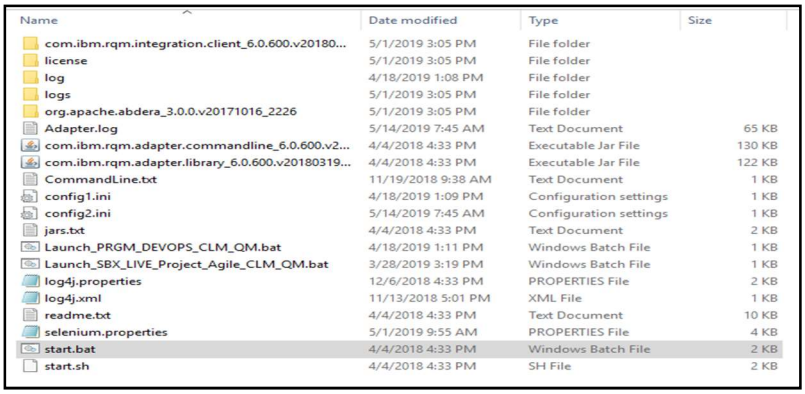


Figure 38 – JUnit Selenium Adapter Folder

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\RQMSeleniumAdapter***)
    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_PRGM_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

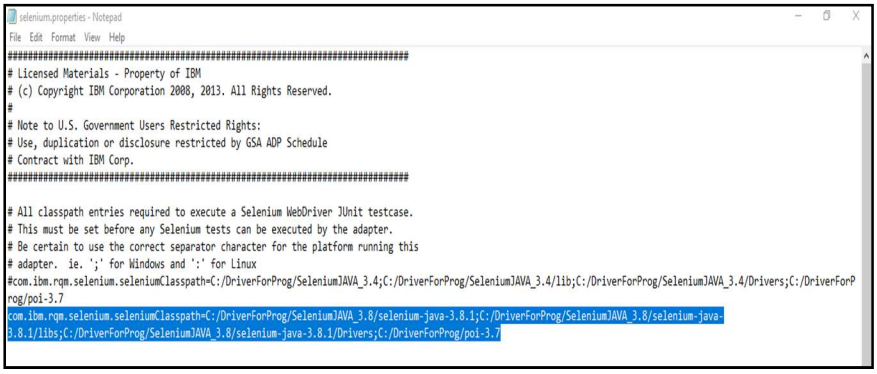
c. Open the text file and paste the following:

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\RQMSeleniumAdapter *** Selenium.properties with the following content:

com.ibm.rqm.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



```
selenium.properties - Notepad
File  Edit  Format  View  Help
################################################################################
# Licensed Materials - Property of IBM
# (c) Copyright IBM Corporation 2008, 2013. All Rights Reserved.
#
# Note 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.rqm.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.rqm.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:

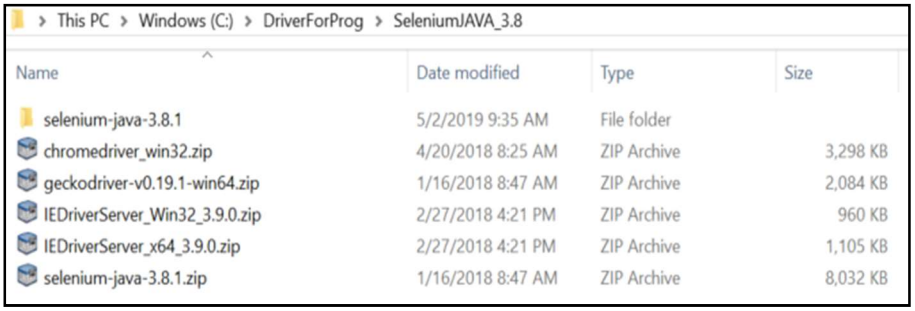


Figure 41 – Selenium 3.8 Folder Structure

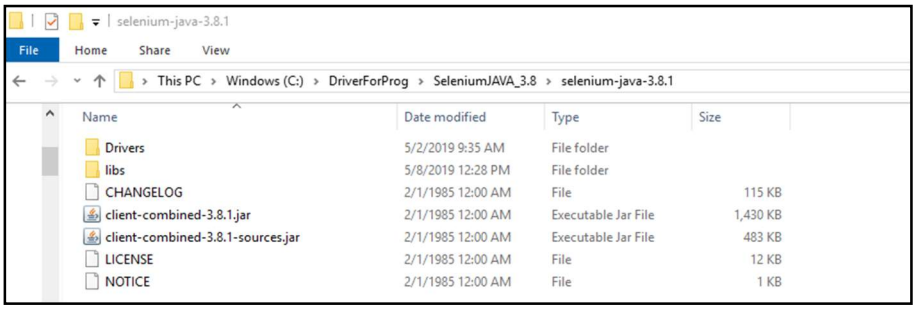


Figure 42 – Selenium 3.8.1 Java Folder Structure

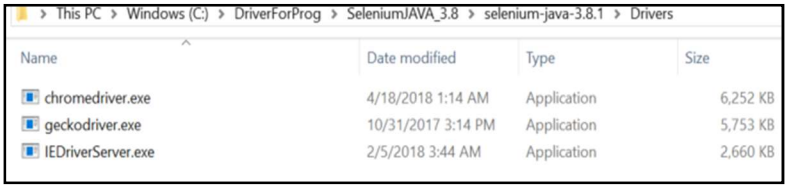


Figure 43 – Selenium 3.8.1 Drivers Folder Structure

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.

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

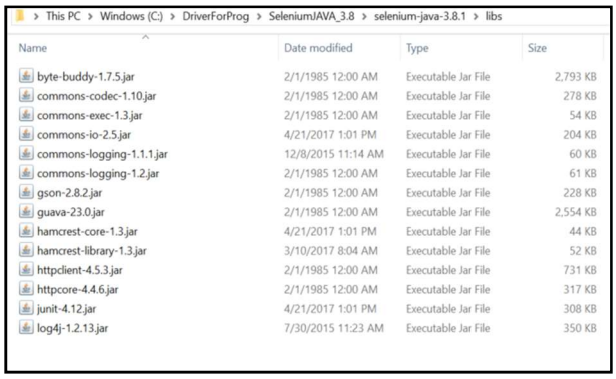


Figure 45 – Complete Set of Selenium 3.8 Library Files

7.5.3 Starting JUnit Selenium Adapter

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

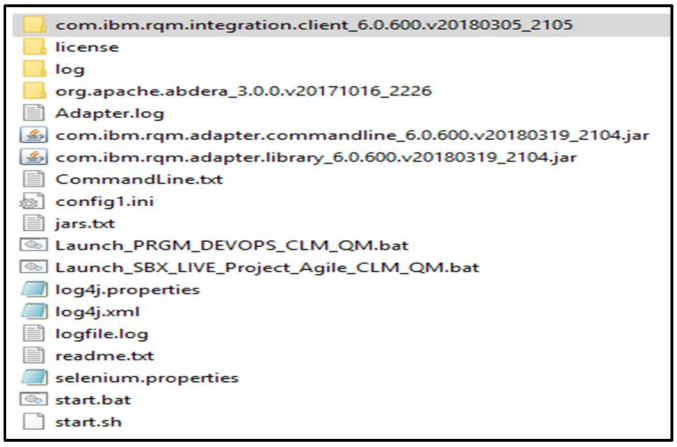


Figure 46 – JUnit Selenium Adapter Folder

2. Double-click on Launch_PRGM_DEVOPS_CLM_QM.bat

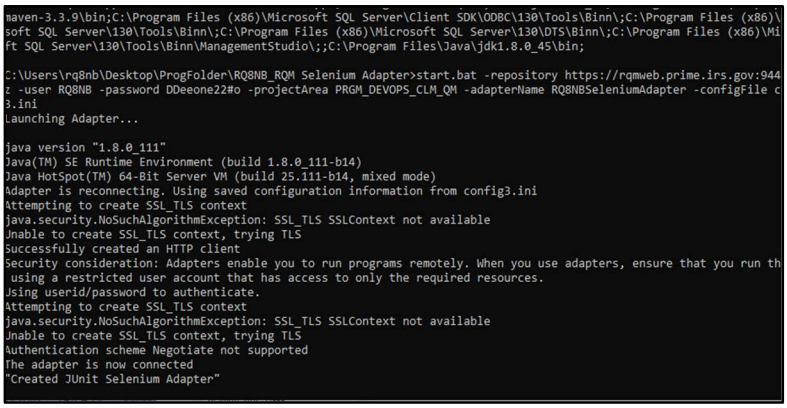


Figure 47 – Starting Adapter in CMD window

3. Verify Selenium adapter is connected to RQM
4. 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 RQM Adapter Console under the execution menu

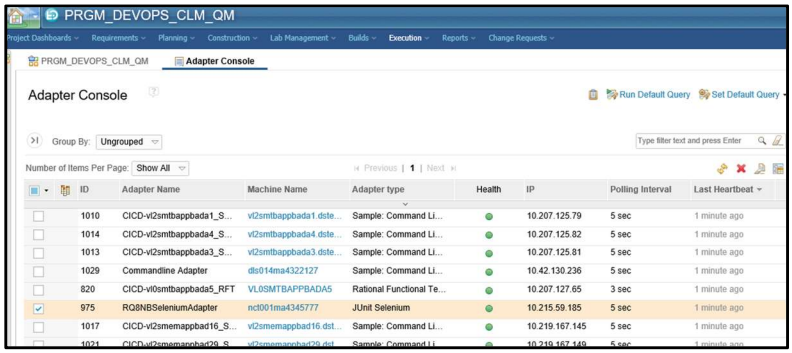


Figure 48 – Adapter Console

7.5.4 Executing Test Cases from RQM

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 case created in previous steps

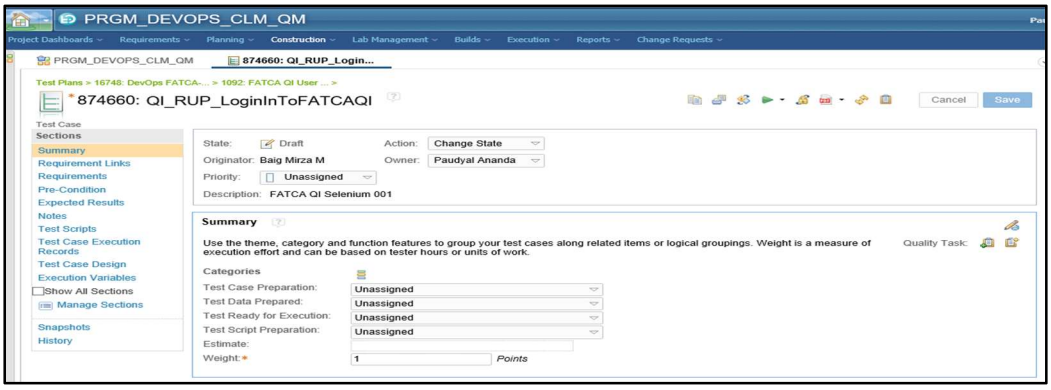


Figure 49 – RQM Test Case Sample

2. Click on 

Figure 50 – RQM Test Case Execution Mode

3. Click on Run

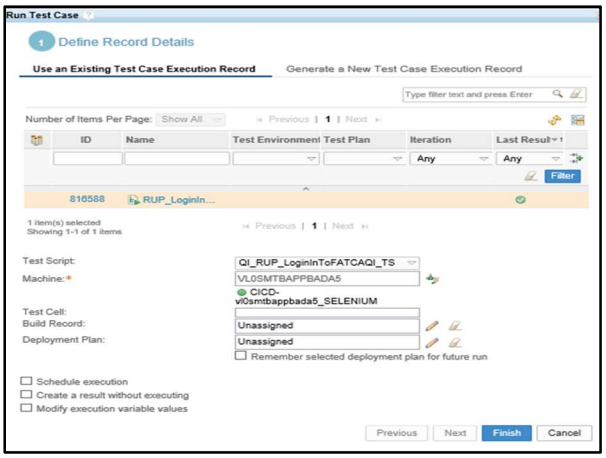


Figure 51 – Run Test Case

4. Click on Machine to select your adapter

5. Select your adapter then click OK button

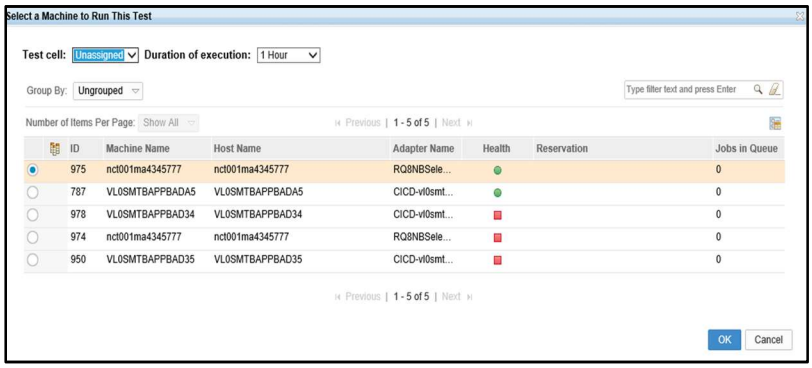


Figure 52 – Select a Machine to Run This Test

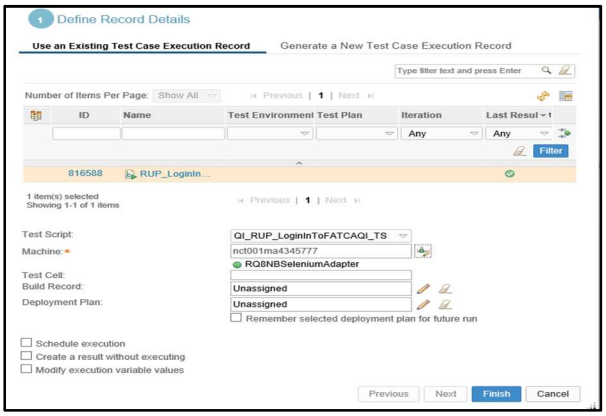


Figure 53 – Define Execution Record Detail

6. Click Finish

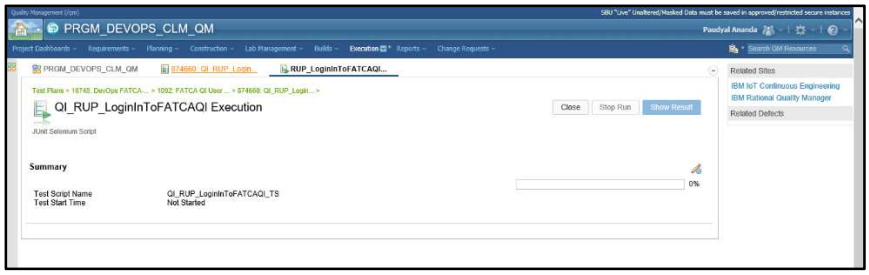


Figure 54 – Starting the Test

7. Automation will begin executing

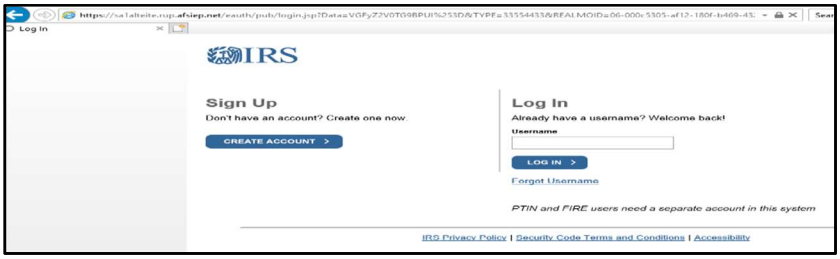


Figure 55 – Navigating FATCA QI Application

8. Wait for test execution to complete

9. Showing error message during execution

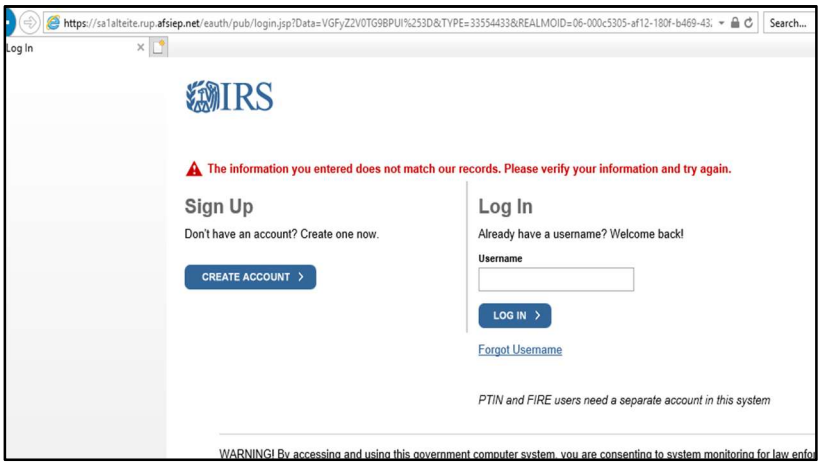


Figure 56 – Showing Error Message

10. Test Execution is complete and click on Show Result

Figure 57 – Test Case Execution Status

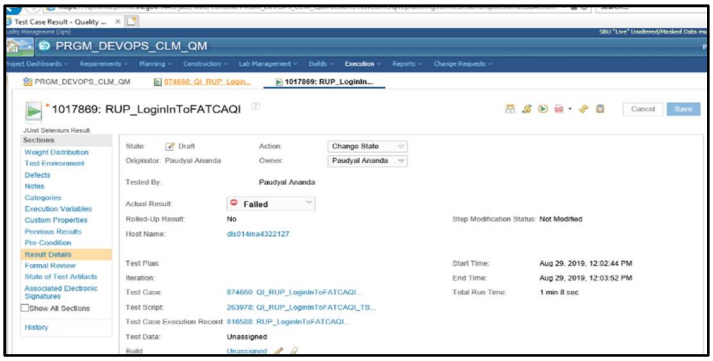


Figure 58 – Test Case Status

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.

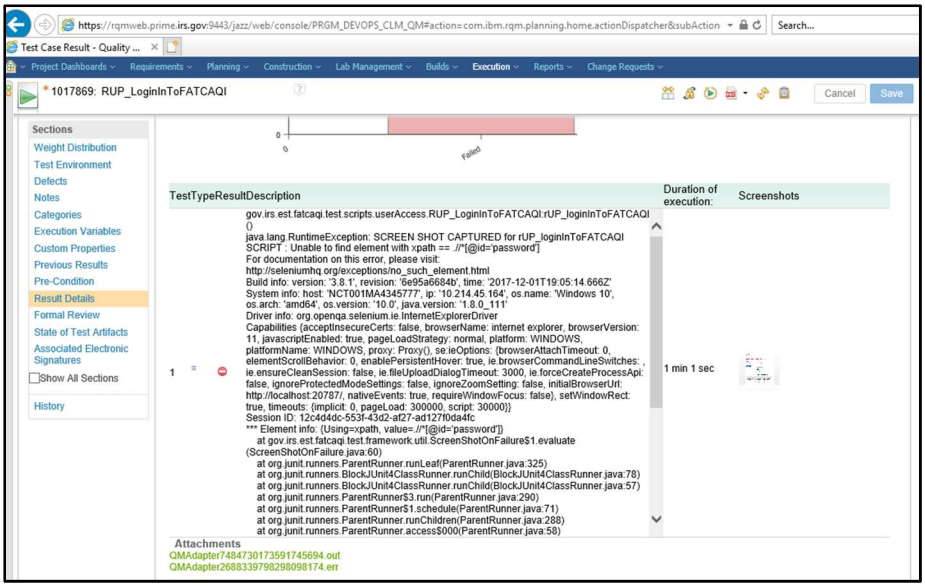


Figure 59 – Error Message for Fail Test Case

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.

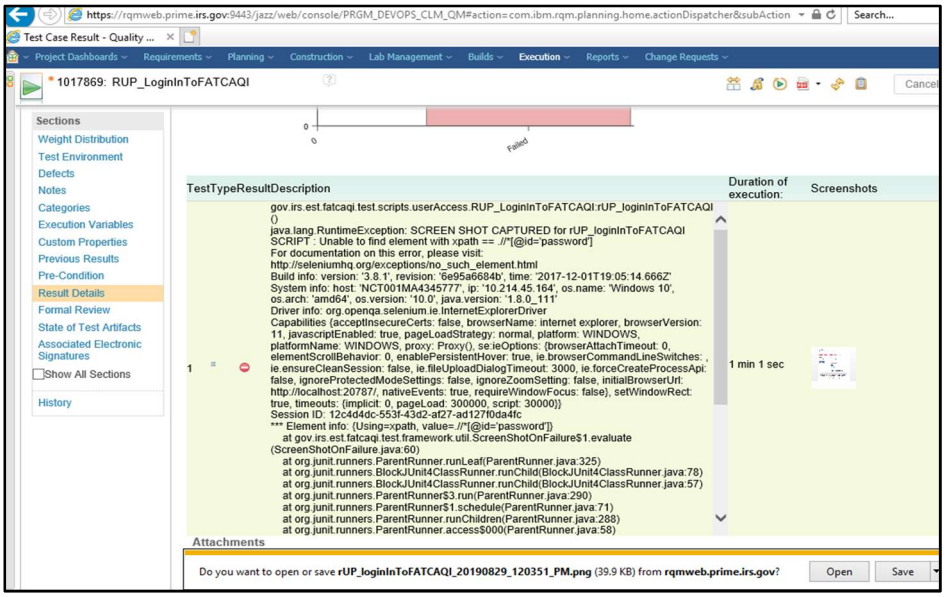


Figure 60 – Downloading PNG File

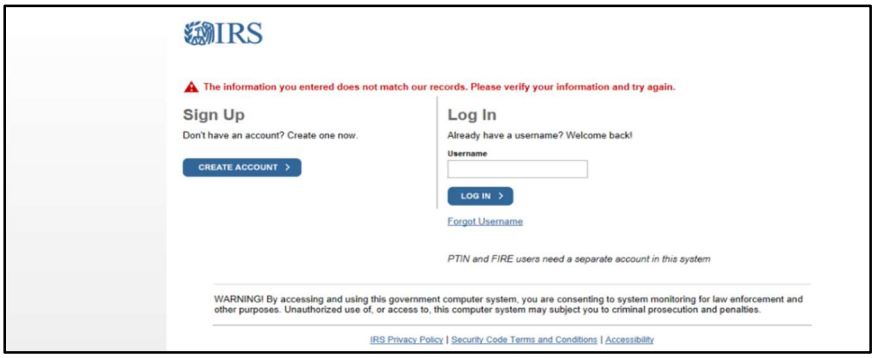


Figure 61 – Opening Error Image

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

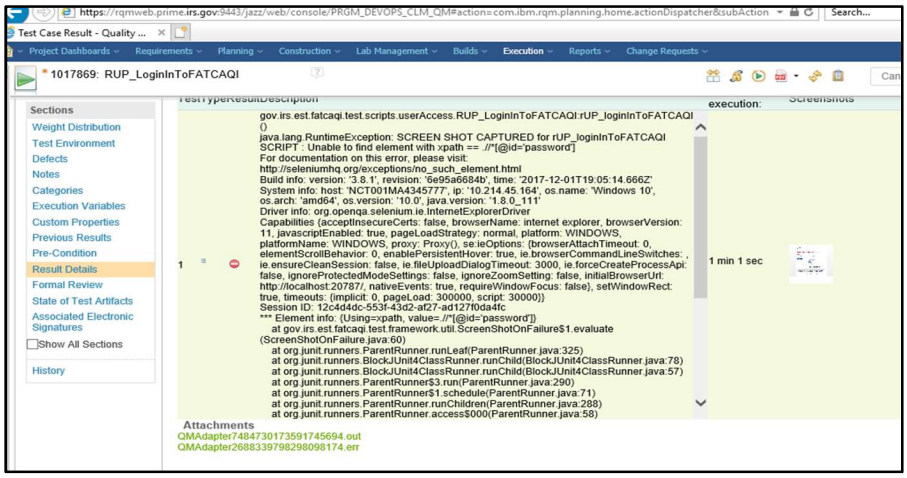


Figure 62 – Opening Out File

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

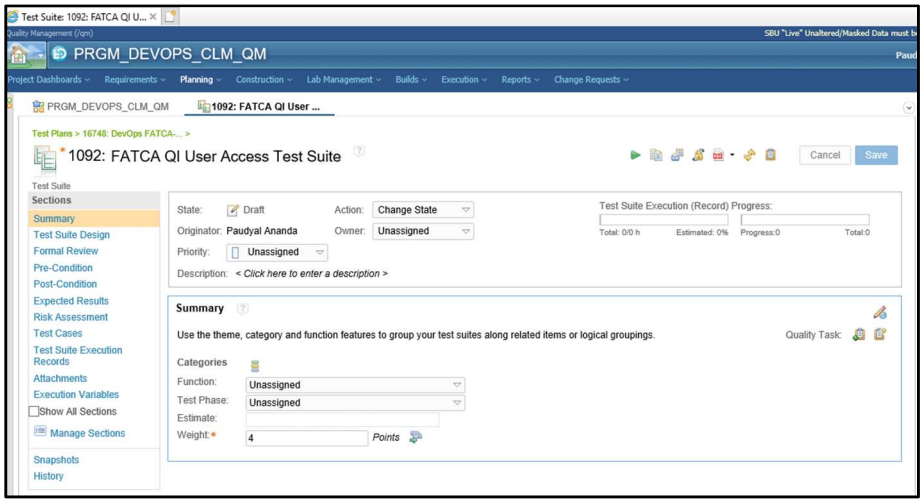


Figure 64 – RQM Test Suite Sample

2. Click on Test Case

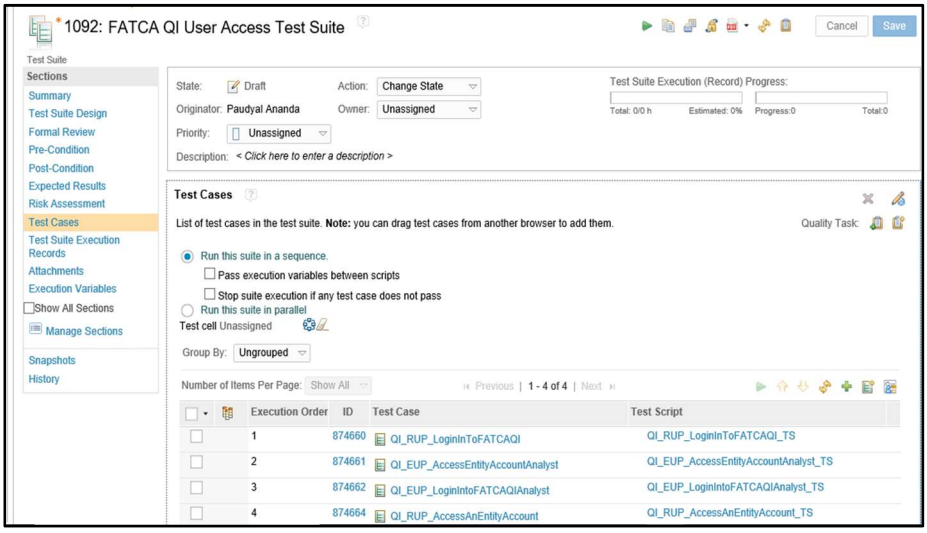


Figure 65 – RQM Test Suite Showing Test Cases

3. Click on 

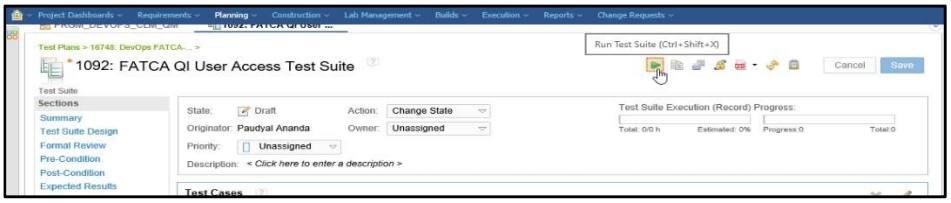


Figure 66 – Executing Test Suite

4. Select test cases for execution

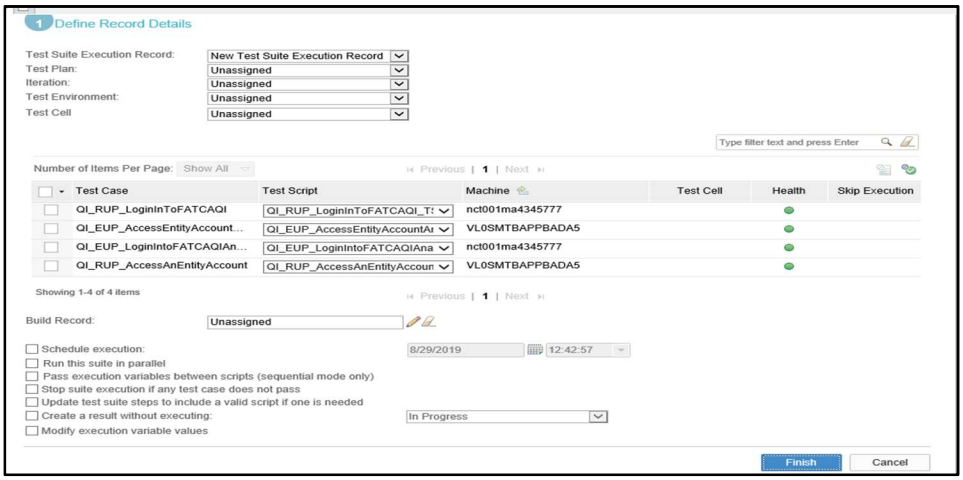


Figure 67 – Define Record Detail for Test Suite

5. Click Machine

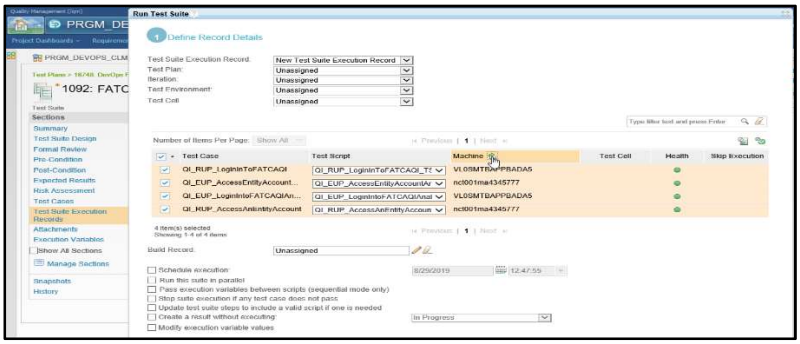


Figure 68 – Identifying an Adapter

6. Select JUnit Selenium Adapter

Figure 69 – Selecting an Adapter

7. Click OK

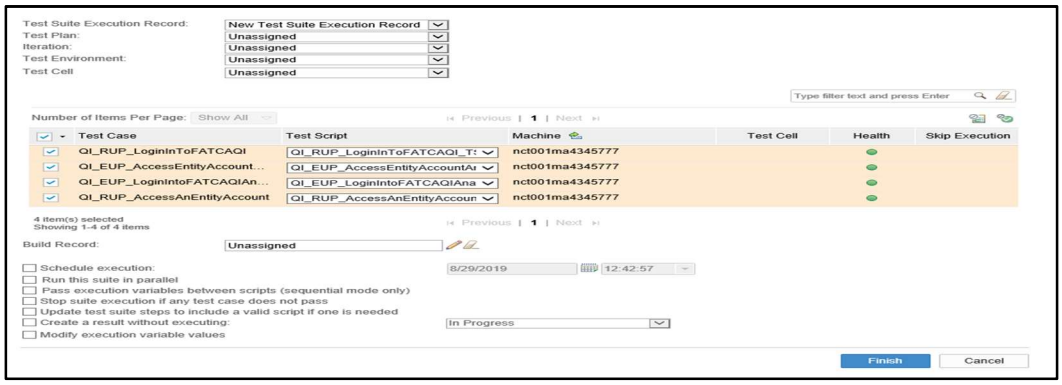


Figure 70 – Adapter Selected

8. Click Finish

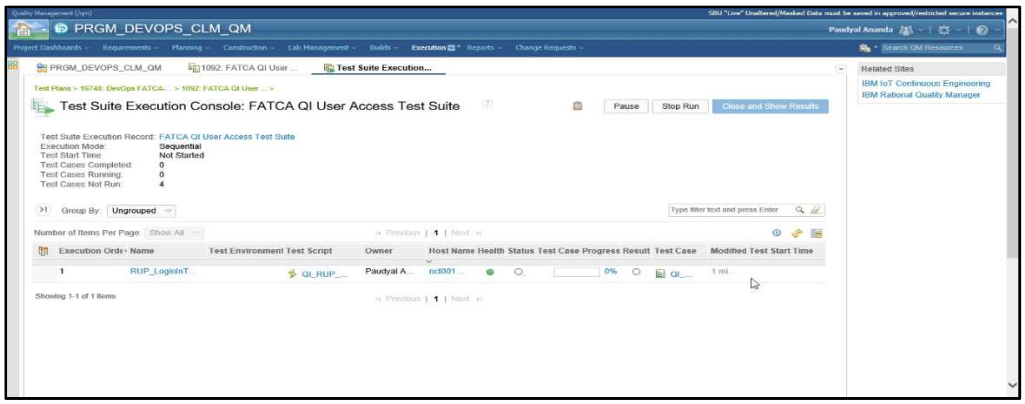


Figure 71 – Executing Test Cases with Test Suite

9. Verify Test Suite status in RQM

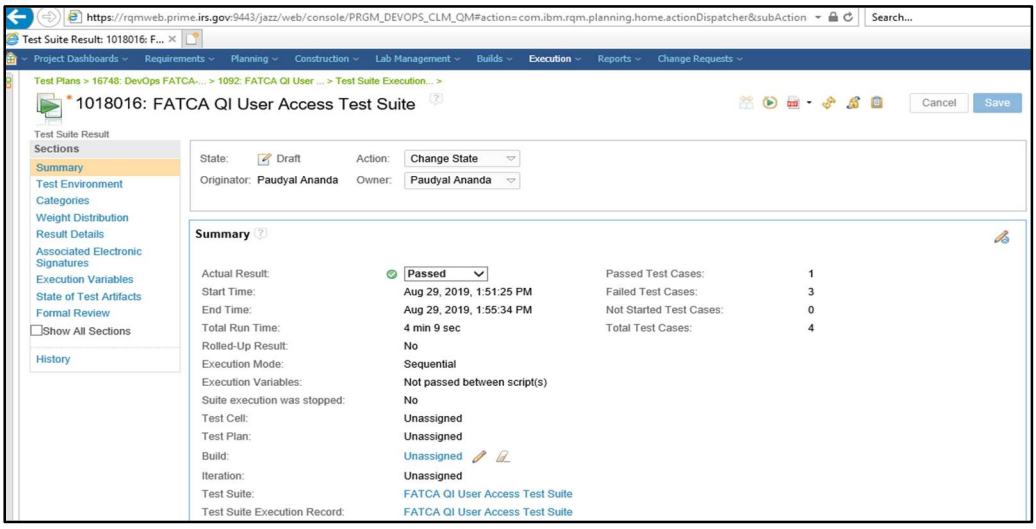


Figure 72 – Test Suite Execution Status

10. Verify test case status within Test Suite

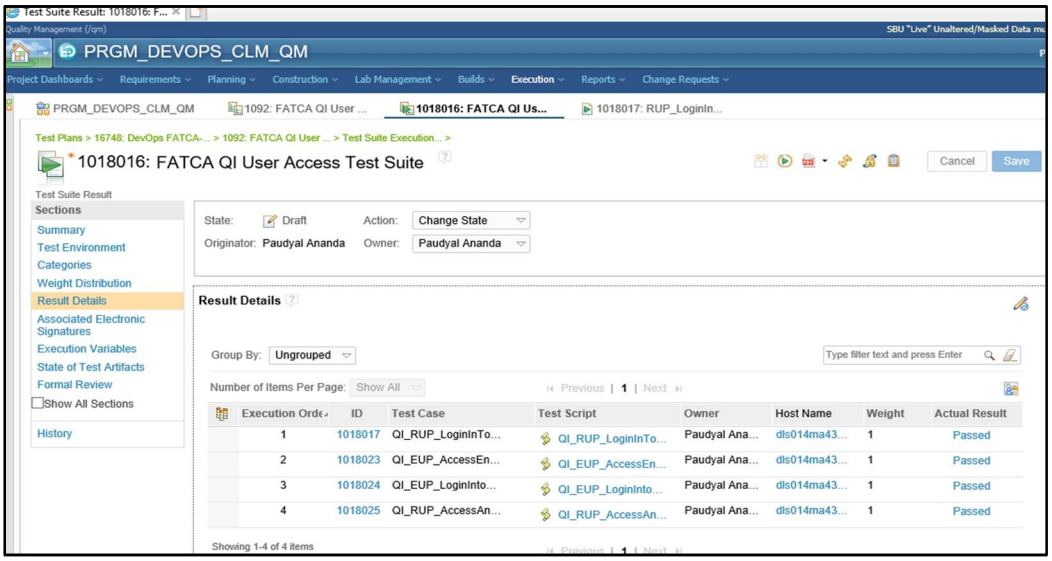


Figure 73 – Showing Status of Each Test Case

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.

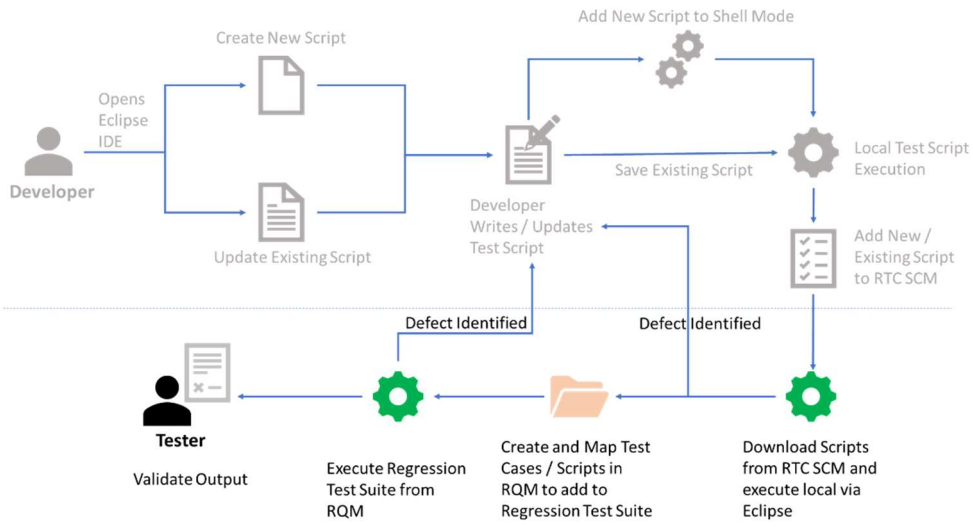


Figure 74 – Defect Reporting Process

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 leveland is not project specific. Depending on how each project is developed, addition 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.

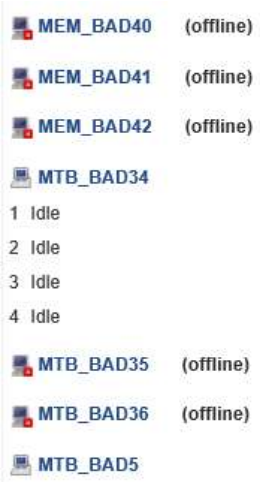


Figure 75 – Jenkins Master - Worker Configuration

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
- RQMExecutionTool_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)

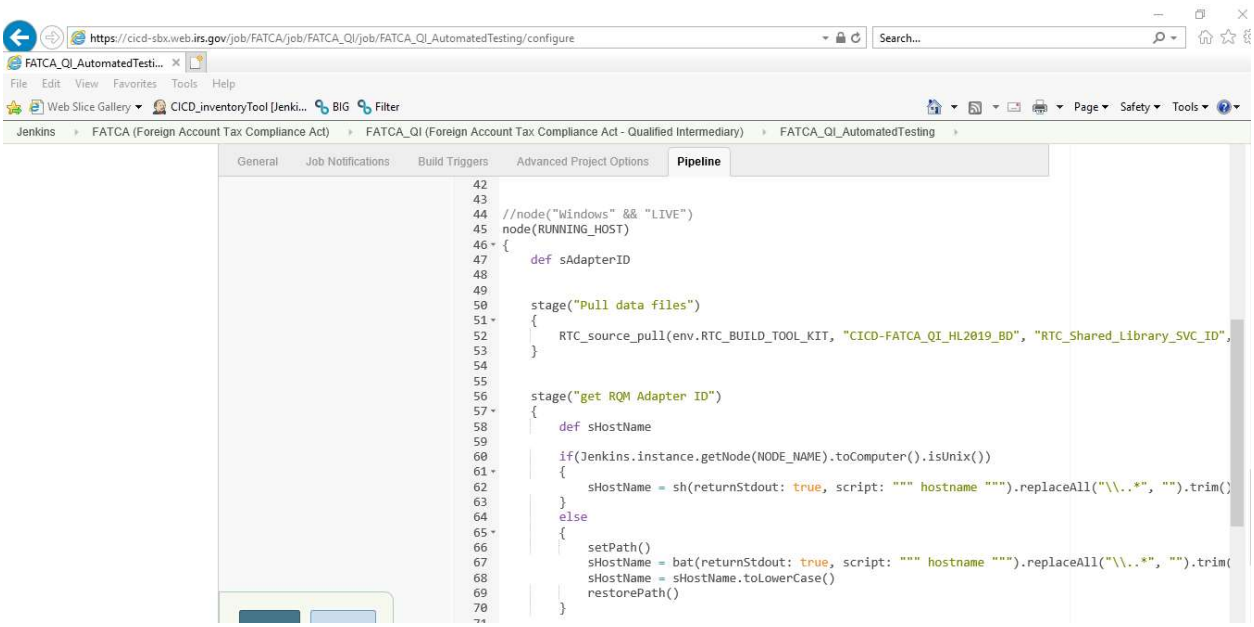


Figure 76 – Fetching Source Code

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

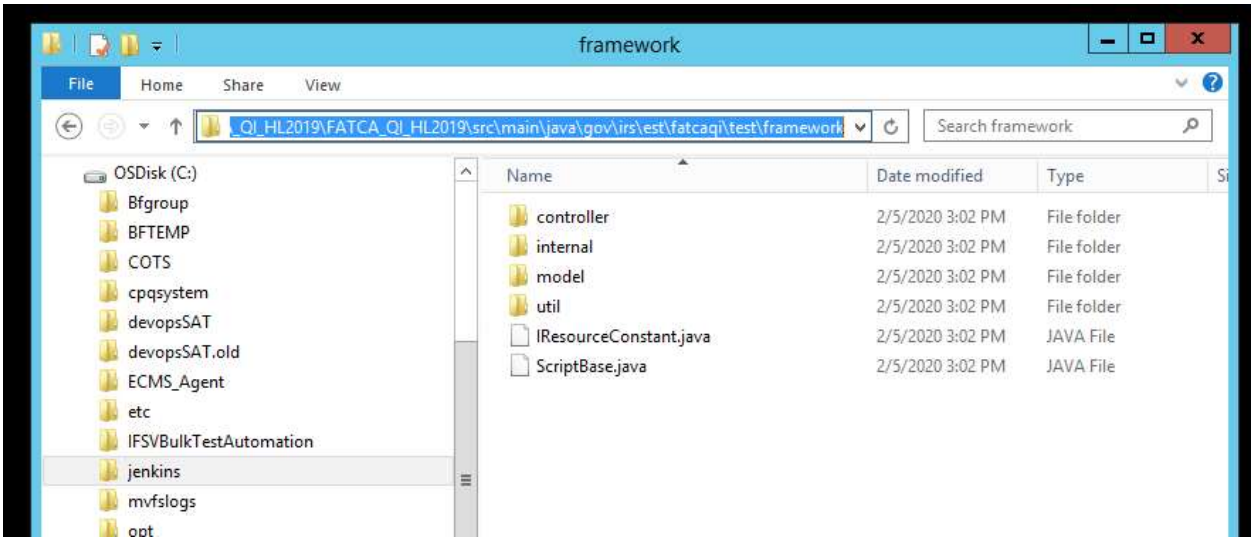


Figure 77 – Project Folder Structure

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://rqmweb.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://rqmweb.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, ithas 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

IRS

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 Reveiw
1.3	03/18/2020	Ronald Shepherd/Ben Wright	Final Review