



Qualitia Software Pvt. Ltd.

Qualitia Offline - Help

Contents

1. Introduction	4
2. Qualitia Offline Package.....	5
1.1 Offline Package - Creation.....	5
1.1.1 Step 1: Packaging Execution Engine.....	5
1.1.2 Step 2: Creating Tests Package	5
1.2 Offline Package - Deployment and Execution.....	6
1.2.1 Understanding qualitia.properties.....	6
1.2.2 Offline Package - Execution.....	7
2 Offline Execution on Different Environments.....	9
2.1 Supported Environments	9
2.1.1 Offline execution on local	9
2.1.2 Offline execution on remote	9
2.1.3 Offline execution on Sauce Labs	9
2.2 Supported Platforms and Browsers	9
2.2.1 Execution - Supported Browsers and Platforms	9
2.2.2 Report Viewing - Supported Browsers and Platforms	9
2.3 Offline Execution On Local Environment	10
2.3.1 Configuration Settings.....	10
2.3.2 Offline Execution and Report.....	11
2.3.3 Execution using Browser Capabilities	11
2.4 Offline Execution on Remote/Distributed Environment	11
2.4.1 Normal Remote Execution	11
2.4.2 Remote Execution through Selenium Grid	13
2.4.3 Offline Execution on Sauce Labs	15
2.4.4 Execution using Browser's Capabilities	21
2.4.5 Reports	22
2.4.6 Limitations Of Remote/Sauce Executions.....	23
2.4.7 Support for AutoIT in Sauce execution environment	23
2.4.8 Best Practices	24

3	Cross platform execution	25
3.1	Supported Platforms and Browsers	25
3.1.1	Execution – Supported Browsers and Platforms	25
3.1.2	Report Viewing - Supported Browsers and Platforms	25
4	Cross platform executions.....	26
4.1.1	Qualitia System Variables	27
4.1.2	Building cross platformtest cases	27
5	Parallel Execution	29
6	Headless Executions	30
7	Continuous Integration	31
7.1	Jenkins Installation.....	31
7.2	Jenkins Configuration.....	31
7.3	Execute and View Report.....	35
7.4	Parallel Execution with Jenkins.....	36
7.4.1	Install Jenkins Plugins.....	36
7.4.2	Configure Build Flow plugin for Jenkins	36
7.4.3	Configure qualitia offline job	36
7.4.4	Execute Tests in Parallel.....	37
8	Team City	38
8.1	Team City Installation	38
8.2	Team City Configuration	38
8.3	Execute and View Report.....	39

1. Introduction

Qualitia can now execute tests on Linux and Mac. This means that you can now create anOffline package in Qualitia and execute the same package on multiple platforms. The offline package is a jar (QualitiaOffline.jar) which can be configured to execute on multiple platforms and browsers. Please see the platform, browser matrix given below to see what is supported.

This document explains how you can create an offline package, configure and execute it against multiple platforms and browsers.

2. Qualitia Offline Package

Qualitia Offline Package (QOP) is a portable and readily deployable package of Qualitia Tests and Qualitia Execution engine which has the ability to independently (without Qualitia Test Automation Studio) execute and report results.

Qualitia Offline Package = Qualitia Execution Engine + Tests Package

QOP can be created from the Test Automation Studio and then executed on multiple environments. These environments DO NOT need Qualitia Test Automation Studio installed as Qualitia Offline houses the Qualitia Execution Engine which can drive and report tests exactly as Qualitia Test Automation Studio does.

1.1 Offline Package - Creation

As Qualitia Offline constitutes of Qualitia tests package and Qualitia Execution Engine, the Offline package creation is a two-step process.

- Packaging Execution Engine
- Creating Tests Package

1.1.1 Step 1: Packaging Execution Engine

- i. The first step of creating an offline package is to package the Qualitia Execution Engine.
- ii. In File menu, click Offline Package.
- iii. "Browse for Folder" screen is displayed.
- iv. Select the path to create the Offline Package.
Good Practice: Create a folder for e.g. "qualitia-offline" where Qualitia Offline package will be saved
- v. Click OK.
- vi. "QualitiaOffline.zip" should be created at the given location which can then be extracted at the same location or any desired location of your choice.
- vii. Extract Qualitia Offline.zip to a desired location

We will understand the structure on the Qualitia Offline package in a later section when we configure the package

1.1.2 Step 2: Creating Tests Package

Tests Package in Qualitia can be created using the "Save As XML" feature in Qualitia

"Save As XML" on Execute tab compiles the selected tests marked in the suite and creates a package.

Here is a step by step walk through on how to use "Save As XML"

- i. Click Execute. Suite screen is displayed.
- ii. On the left pane, select the suite.
- iii. Right pane displays the test cases under the selected suite.

- iv. Mark the test cases to be packaged. Please note, only marked test cases will be packaged. If you wish to package all the tests in a suite, mark all the test cases.
- v. Click “Save As XML”.
- vi. “Browse for Folder” window is displayed. Select a folder to save the XML file. (e.g. D:\QualitiaOffline\<<My Tests Package>>
Good Practice: Use the name of the suite for My Tests Package.e.g. D:\QualitiaOffline\smoke
- vii. Click OK.
- viii. You can verify that the package is created at the given path and verify that it contains Suite.xml, Map.xml, and test case xmls that you saved.

1.2 Offline Package - Deployment and Execution

Once the package is created, we can move it to any location. Remember that deploying an offline package will require both Qualitia execution package and the tests package.

Now let’s understand the contents of the Offline package before we execute it

The key components to note in the package once you extract are

1. **qualitia-offline-3.9.1.jar**: This is qualitia execution engine that will execute the tests
2. **“lib”** and **“common”** folder: Both these folders are required by the Qualitia execution engine. Users should not alter the contents of these folders
3. **“config/qualitia.properties”** folder: qualitia.properties file in the config folder holds all the configuration settings that existed in Qualitia at the time of package creation. (We will see how to use the qualitia.properties later)

1.2.1 Understanding qualitia.properties

It’s important to understand the configurations in qualitia.properties as this is where we will configure our offline package for execution.

The default configuration values are same as those in your Qualitia configuration at the time of offline package creation. Some of the configurations, especially the ones that contain file paths and a few others will need to be changed before we execute our tests

Here is the list of properties that will most likely need change, before we execute the offline package. Please see the qualitia.properties file for the complete list of properties and explanations for each of them.

1. **xmlPath**
xmlpath –The folder path where Suite.xml and test case xmls are stored. This is where you saved the Tests Package. Please refer Section 1.1.2
2. **LogPath**
LogPath– Path where Qualitia reports will be saved
3. **SBrowserType**
SBrowserType – Browser to be used during execution. e.g. firefox, iexplore, googlechrome, safari

4. ShowReportAfterExec

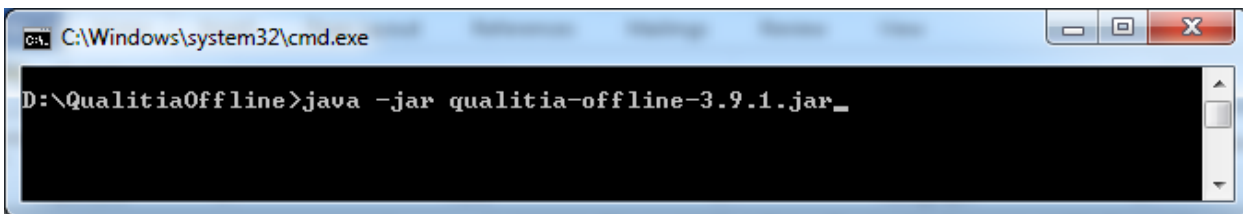
ShowReportAfterExec – true to show report after execution, false otherwise. Default is true for Qualitia Suite execution and false for Offline Package execution

You can change most properties unless indicated otherwise in the qualitia.properties file.

1.2.2 Offline Package - Execution

We can execute the offline package from the command line after navigating to the folder where we saved the Offline Package and calling the qualitia-offline jar using java. See below image for an example

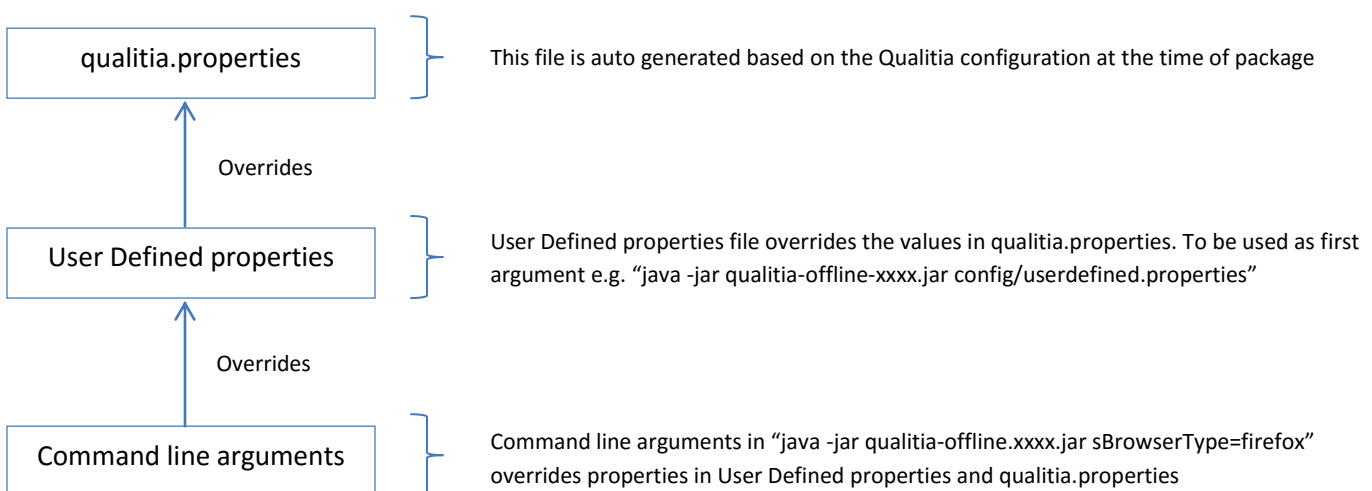
Note: This will use the java referenced in the “JAVA_HOME” system variables of your environment variables. It’s important to note this in case you have multiple versions of java installed.



This command should execute the tests using qualitia.properties under config folder.

You can also use other ways of configuring properties in Offline to better maneuver your configurations in multiple environments.

To understand this we will need to understand how Qualitia Offline package uses the properties. This can be understood well from the diagram below



**xxxx: denotes the version of qualitia-offline jar. Please check the Offline location for the exact jar file name*

As shown in the above diagram, configuration properties can be changed at three different levels

1. **Level 1: qualitia.properties**

This file is generated by Qualitia when we create the offline package

2. **Level 2: User Defined properties file**

We can create our own properties file and provide the path for it as the first argument to the qualitia-offline-xxxx.jar. This provides the flexibility to create multiple properties files based on the required configuration compositions. This allows us to create properties files for each OS, browser, their combinations, etc...

For e.g. win7.properties, mac.properties, linux.properties, linux-ff.properties, linux-headless.properties and so on.

3. **Level 3: Command line arguments**

Command line arguments allow further fine tuning of the configurations by providing arguments to the qualitia-offline-xxxx.jar. Command line arguments will override user defined properties if one is provided as the first argument or it will override the values in qualitia.properties

All Properties part of the qualitia.properties file can be provided as command line arguments using name=value format

e.g. Property Name: sBrowserType containing Property Value: firefox can be provided as follows

`qualitia-offline-xxxx.jar sBrowserType=firefox`

Similarly we can provide multiple property=value pairs by separating them with a space

`qualitia-offline-xxxx.jar sBrowserType=firefox xmlPath=D:\QualitiaOffline\Scripts SyncTimeInSeconds=40`

2 Offline Execution on Different Environments

The same Qualitia offline jar can be used to execute tests on different environments. For this, we will have to make changes to the properties file or provide arguments based on the execution environment.

2.1 Supported Environments

Offline package now can be executed on Sauce Labs and Remote environments along with local environment. Depending on the desired environment Qualitia.Properties will accommodate configurations as follows.

2.1.1 Offline execution on local

To execute offline package on local machine, configure Qualitia.Properties as ExecutionEnvironment=local

2.1.2 Offline execution on remote

To execute offline package on local machine, configure Qualitia.Properties as ExecutionEnvironment=remote

2.1.3 Offline execution on Sauce Labs

To execute offline package on local machine, configure Qualitia.Properties as ExecutionEnvironment=sauce

2.2 Supported Platforms and Browsers

Support for browsers and platforms can be classified into

- Support for Execution
- Support for Report Viewing

2.2.1 Execution - Supported Browsers and Platforms

Here is a quick view of the browsers and platforms that you can execute your tests on, by using the offline package.

	Local				Remote				Sauce Labs			
	IE	Firefox	Chrome	Safari	IE	Firefox	Chrome	Safari	IE	Firefox	Chrome	Safari
Windows	X	X	X		X	X	X		X	X	X	
Linux		X	X			X	X			X	X	
Mac												X

X: denotes support for the browser on a platform

2.2.2 Report Viewing - Supported Browsers and Platforms

After the execution Qualitia will show the report in IE on the local machine from where execution is triggered.

Here is a quick view of the browsers that can be used for Qualitia report viewing.

	IE	Firefox	Chrome	Safari
Windows	x	x	Not Supported	
Linux		x	Not Supported	
Mac		x	Not Supported	x

Chrome is not supported for report viewing.

2.3 Offline Execution On Local Environment

This is the traditional way of executing offline package that has been followed till the date. The only change in order to achieve it, is you need to tell qualitia-offline.jar that you want to execute the Offline Package on your local machine as explained below.

2.3.1 Configuration Settings

- Open Qualitia.Properties file present in 'Config' folder of Offline Package.

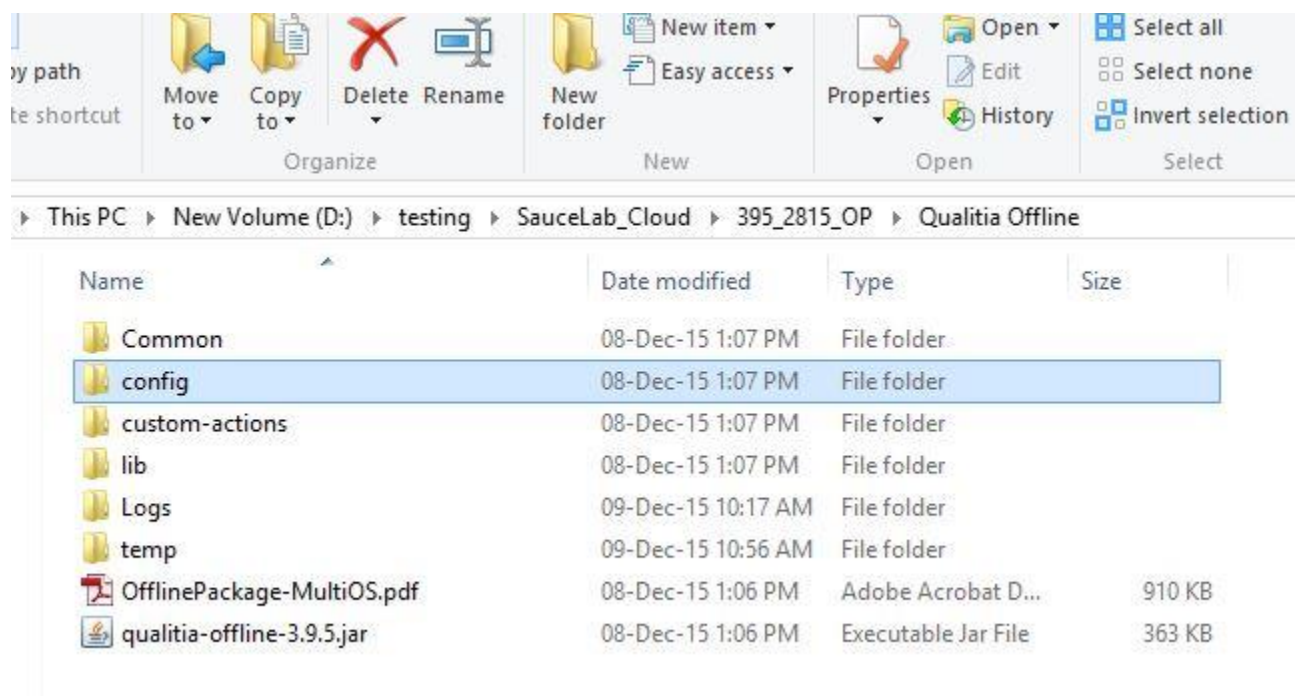


Fig: 2.3.1-a

- Now set "ExecutionEnvironment" key to 'local' which tells qualitia-offline.jar that Offline package has to be executed on local machine.

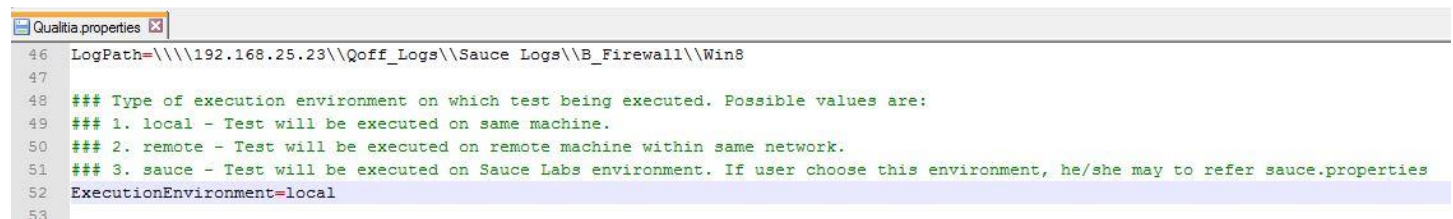


Fig:2.3.1-b

Note that all the other configuration settings in Qualitia.properties file has to be done in the traditional way.

2.3.2 Offline Execution and Report

- Once configuration settings are done, execute Offline Package. To execute it you can follow the traditional method of offline execution .For more details please refer section 1.2.2 above.
- For execution, the locally available browsers will be used. e.g. If you have chosen "Firefox" to be your execution browser and your local machine has Firefox v42, the execution will be continued on Firefox v42.
- After the execution is complete, Test Execution Report will be shown on the local machine.

2.3.3 Execution using Browser Capabilities

To execute test on remote environment using selenium supported Browser Capabilities, refer section "Execution using Browser's Capabilities".

2.4 Offline Execution on Remote/Distributed Environment

Now execution of an Offline Package can be achieved on any remote machine in the network. This gives a provision of triggering execution of a test suite from any machine in the network so that execution will happen on any other machine in the same network. Hence you don't need to move the Offline Package to all the machines in network in order to execute it there.

There are two ways to execute your offline package on Remote machine in the network.

- Normal Remote Execution
- Remote Execution through Selenium Grid

2.4.1 Normal Remote Execution

If you are aware of the remote machine details where you wish to execute your test you can choose this option. Note that here you choose to execute your Offline Package on a remote machine with specific details.

2.4.1.1 Prerequisites

A selenium- stand-alone-server jar to be used for execution should be running on the target remote machine as follows.

- Steps to Run Selenium Jar:
 - Say Remote Machine 'Test_Remote' is the target machine. Download and store the selenium jar in a directory. Refer figure below.

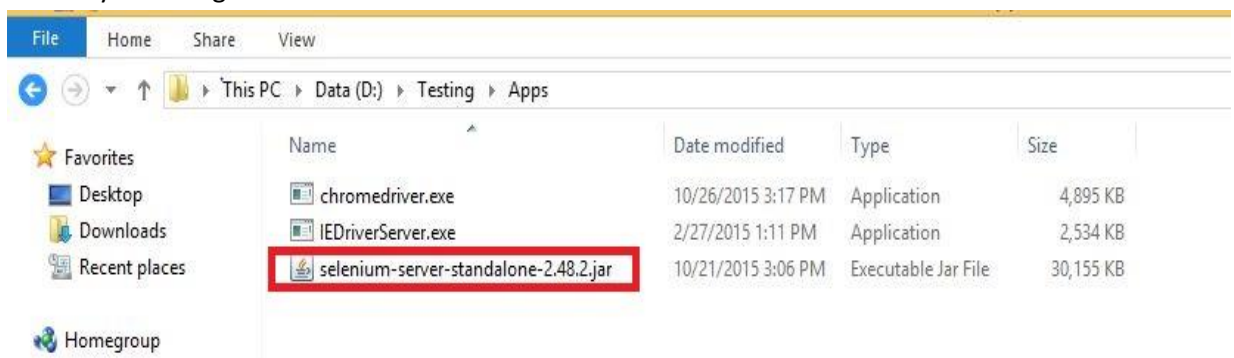


Fig: 2.4.1.1-a

- Open command prompt and navigate till directory where jar is stored. Type command '*java -jar selenium-server-standalone-2.45.0.jar*' and press enter key.
- When Offline execution browser is 'Iexplore', the corresponding driver(IEDriverServer.exe) should also be bundled with the selenium jar. Download and store IEDriverServer in a directory(refer fig 2.4.1.1-a) and execute command as '*java -jar selenium-server-standalone-2.45.0.jar -Dwebdriver.ie.driver="D:\Data\Testing\Apps\IEDriverServer.exe"*'.

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

D:\Testing\Apps>java -jar selenium-server-standalone-2.48.2.jar -Dwebdriver.ie.driver="D:\Testing\Apps\IEDriverServer.exe"

```

Fig: 2.4.1.1-b

- When Offline execution browser is 'GoogleChrome', the corresponding driver(ChromeDriver.exe) should also be bundled with the selenium jar. Download and store ChromeDriver in a directory(refer fig 2.4.1.1-a) and execute command as '*java -jar selenium-server-standalone-2.45.0.jar -Dwebdriver.chrome.driver="D:\Testing\Apps\chromedriver.exe"*'

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

D:\Testing\Apps>java -jar selenium-server-standalone-2.48.2.jar -Dwebdriver.chrome.driver="D:\Testing\Apps\chromedriver.exe"

```

Fig: 2.4.1.1-c

2.4.1.2 Configuration Settings

- Open Qualitia.Properties file present in 'Config' folder of Offline Package. Refer figure 2.3.1 above.
- Set "ExecutionEnvironment" key to 'remote' which tells qualitia-offline.jar that Offline package has to be executed on a remote machine in the network.
- Now set key "RemoteURL" key to a valid remote URL which is nothing but IP and port of remote machine e.g. *RemoteURL=<http://192.168.25.123:4445>*

```

Qualitia.properties
46 LogPath=\\\\192.168.25.23\\Qoff_Logs\\Sauce_Logs\\B_Firewall\\Win8
47
48 ### Type of execution environment on which test being executed. Possible values are:
49 ### 1. local - Test will be executed on same machine.
50 ### 2. remote - Test will be executed on remote machine within same network.
51 ### 3. sauce - Test will be executed on Sauce Labs environment. If user choose this environment, he/she may to refer sauce.properties
52 ExecutionEnvironment=remote
53
54 ###This property basically use when ExecutionEnvironment is Remote. Remote machine URL where test to be executed.
55 RemoteURL=http://testing\_vm:4444

```

Fig: 2.4.1.2

2.4.1.3 Offline Execution and Reports

- Once configuration settings are done, execute Offline Package. To execute it you can follow the traditional method of offline execution .For more details please refer section 1.2.2 above.
- For execution, the browsers available on specified remote environment will be used. e.g. If you have chosen "Firefox" to be your execution browser and your remote machine has Firefox v42, the execution will be continued on Firefox v42.
- After the execution is complete, Test Execution Report will be shown on the local machine from where execution is triggered.

2.4.1.4 Execution using Browser Capabilities

To execute test on remote environment using selenium supported Browser Capabilities, refer section "Execution using Browser's Capabilities".

2.4.2 Remote Execution through Selenium Grid

When you have distributed test environments, where you want network to select a remote machine based on the availability of machines, you can choose this option.

Before using it lets understand few terminologies.

What is Hub?

The Hub is the main command center for managing which machines your test will run on. This is where you can see list of all the available nodes currently running in your grid environment (Remote Execution Environment).

What is Node?

A Node is a machine that you register with your Hub. By registering with the Hub, the Hub now knows about the Node and any configuration information you used when you registered the node.

2.4.2.1 Prerequisites

Configure Hub and node in the network as follows.

- **Configure Hub:**
Decide a machine(with ip say 192.168.25.6) in the network that will be registered as Hub.
Download and store Selenium-server-standalone jar file on Hub machine.
Start the command prompt and navigate to the location at which Selenium server jar file is present.

Execute command "*java -jar selenium-server-standalone-2.45.0.jar -role hub -port 4445*"

```

C:\Windows\System32\cmd.exe - java -jar selenium-server-standalone-2.48.2.jar...
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

D:\Testing\Apps>java -jar selenium-server-standalone-2.48.2.jar -role hub -port
4445
17:28:27.889 INFO - Launching Selenium Grid hub
2015-12-10 17:28:29.296:INFO::main: Logging initialized @1847ms
17:28:29.329 INFO - Will listen on 4445
17:28:29.432 INFO - Will listen on 4445
2015-12-10 17:28:29.438:INFO:osjs.Server:main: jetty-9.2.z-SNAPSHOT
2015-12-10 17:28:29.498:INFO:osjs.ContextHandler:main: Started o.s.j.s.ServletC
ontextHandler@1d0c81b</,null,AVAILABLE>
2015-12-10 17:28:29.741:INFO:osjs.ServerConnector:main: Started ServerConnector@
ad1f1f<HTTP/1.1><0.0.0.0:4445>
2015-12-10 17:28:29.744:INFO:osjs.Server:main: Started @2295ms
17:28:29.744 INFO - Nodes should register to http://192.168.25.23:4445/grid/regi
ster/
17:28:29.744 INFO - Selenium Grid hub is up and running
  
```

Fig:2.4.2.1-a

- **Create Node and Register to Hub**

There can be many machines which may act as nodes for you. Select them and declare them to be node by registering to the configured hub as follows.

Download and store Selenium-server-standalone jar file on Hub machine.

Start the command prompt and navigate to the location in which Selenium server jar file is present.

Execute command "*java -jar selenium-server-standalone-2.45.0.jar -role node -hub*

<http://hubIP:4445/grid/register>"

```

C:\Windows\System32\cmd.exe

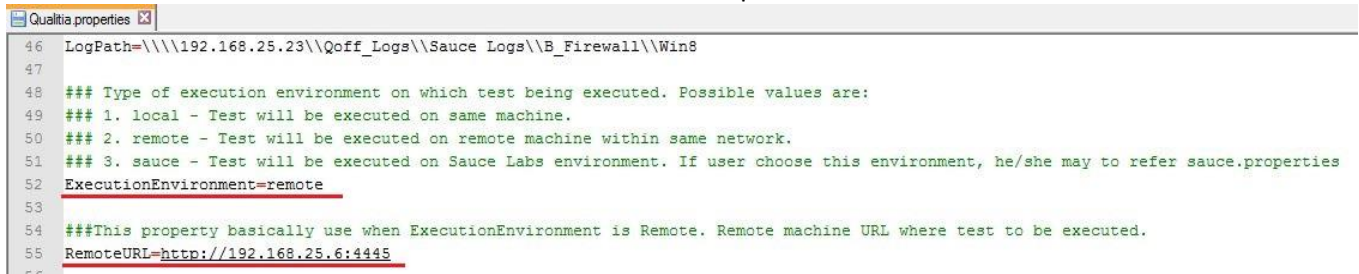
D:\Testing\Apps>java -jar selenium-server-standalone-2.48.2.jar -role node -hub
http://192.168.25.6:4445/grid/register -Dwebdriver.chrome.driver="D:\Testing\app
s\chromedriver.exe"
  
```

Fig:2.4.2.1-b

Note that you can bundle your IEServerDriver.exe or ChromeDriver.exe running while registering to Hub as shown in fig 2.4.2.1-b.

2.4.2.2 Configuration Settings

- Open Qualitia.Properties file present in 'Config' folder of Offline Package. Refer figure 2.3.1 above.
- Set "ExecutionEnvironment" key to 'remote' which tells qualitia-offline.jar that Offline package has to be executed on a remote machine in the network.
- Now set key "RemoteURL" key to a valid remote URL which is nothing but IP and port of machine registered as Hub. e.g. RemoteURL=<http://192.168.25.6:4445>



```

46 LogPath=\\\\192.168.25.23\\Qoff_Logs\\Sauce_Logs\\B_Firewall\\Win8
47
48 ### Type of execution environment on which test being executed. Possible values are:
49 ### 1. local - Test will be executed on same machine.
50 ### 2. remote - Test will be executed on remote machine within same network.
51 ### 3. sauce - Test will be executed on Sauce Labs environment. If user choose this environment, he/she may to refer sauce.properties
52 ExecutionEnvironment=remote
53
54 ###This property basically use when ExecutionEnvironment is Remote. Remote machine URL where test to be executed.
55 RemoteURL=http://192.168.25.6:4445

```

Fig:2. 4.2.2

2.4.2.3 Offline Execution and Report

To execute test on remote environment using selenium supported Browser Capabilities, refer section "Execution Using Browser's Capabilities".

2.4.2.4 Notes

- Selenium server jar version used in Qualitia offline package and that used for remote machines including Hub and Nodes should be same all over.
- Java need to be installed and configured in computer's environment path.
- Port number used for Hub is changeable and default is 4444.
- To know more about how to setup Selenium grid Hub and Node we recommend to use the selenium grid links given below.

<https://github.com/SeleniumHQ/selenium/wiki/Grid2>

<http://www.joecolantonio.com/2014/10/07/selenium-grid-how-to-setup-a-hub-and-node/>

2.4.2.5 Limitations of using Selenium grid

- As Hub selects a Node for execution at its own at run time, You cannot provide details for specific OS/Platform.

2.4.3 Offline Execution on Sauce Labs

Qualitia Offline now can be executed in Sauce Labs.

What is Sauce Labs?

Sauce Labs is a cloud environment which allows users to run tests in the cloud on different operating systems, browsers, browser version, etc. In this environment all tests are executed on Sauce Labs VM. You can see the ongoing suite execution on Sauce labs user dashboard.

To know more about Sauce Labs visit: <https://saucelabs.com/>

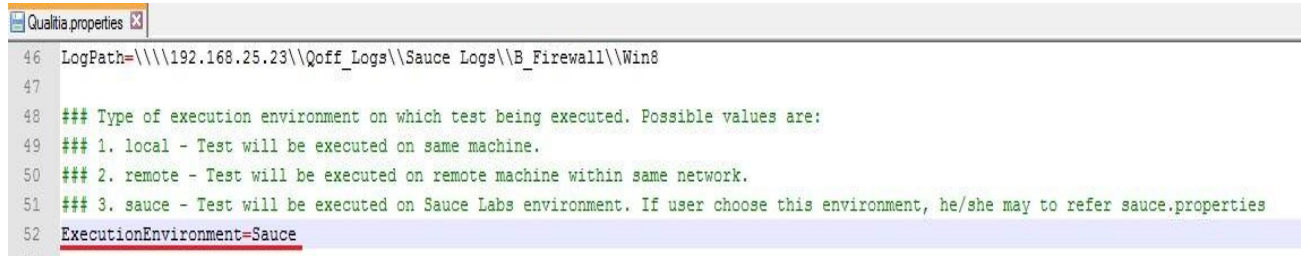
2.4.3.1 Prerequisites

- To execute Offline Package on Sauce Labs you should be a registered user on Sauce Labs. Make sure that you have an active account on Sauce Labs with correct credentials. e.g. Username: omakrd, AccessKey: e602943f-f776-4db9-99b0-a4a61f8ddf10.
- Try login to Sauce Labs and validate if your account is active with sufficient(depends on your test suite execution time) balance of execution hours.

2.4.3.2 Configurations Settings

- Open Qualitia.Properties file present in 'Config' folder of Offline Package. Refer figure 1.1.2 above.

- Set "ExecutionEnvironment" key to 'sauce' which tells qualitia-offline.jar that Offline package has to be executed on a Sauce Labs.



```

46 LogPath=\\\\192.168.25.23\\Qoff_Logs\\Sauce_Logs\\B_Firewall\\Win8
47
48 ### Type of execution environment on which test being executed. Possible values are:
49 ### 1. local - Test will be executed on same machine.
50 ### 2. remote - Test will be executed on remote machine within same network.
51 ### 3. sauce - Test will be executed on Sauce Labs environment. If user choose this environment, he/she may to refer sauce.properties
52 ExecutionEnvironment=Sauce

```

Fig:2.4.3.2-a

- As you have selected 'Sauce' to be your execution environment, before starting execution, Sauce will need Sauce specific details and some additional information. e.g. Platform, Browser, Browser Version etc. Sauce.Properties file is the place where you can provide these details as below.
- Open Sauce.Properties file present in 'Config' folder of Offline Package.
- Set properties "custom_username", "custom_accesskey" to valid values w.r.t. an Sauce-Lab's active account user.
- Set "platform", "browser", "version" to corresponding OS, browser and browser version you want your offline package to execute on.
 - e.g. If you wish your offline package to execute on Iexplore 11 on OS Windows 8.1 then set keys as follows:
 - platform=Windows 8.1
 - browser=Iexplore
 - version=11
 - Note that if you are providing these values through command line, each value should be provided in single quotes else any space in values will raise some inconsistency.
 - e.g. java -jar Qualitia-Offline-3.9.5.jar platform='Windows 8.1'
 - For platforms, browsers and browser versions supported on sauce check link:
 - <https://saucelabs.com/platforms/>
- Note that its sauce who decides default values for the properties from Sauce.Properties. Hence, if you do not provide anything for 'Platform' it will assume 'Linux'. Also if you do not mention anything against 'version' it will execute your tests on latest version of mentioned browser.


```

3 #Property Name:custom_username
4 #Sauce Labs user name. It is used to form Sauce Labs URL and using REST API.
5 #e.g. custom_username=dshivarkar
6
7 custom_username=omakrd
8
9 #Property Name:custom_accesskey
10 #Sauce Labs user access key. It is used to form Sauce Labs URL and using REST API.
11 #e.g. custom_accesskey=3969ad81-8556-4510-a122-d077728f2e15
12
13 custom_accesskey=e602943f-f776-4db9-99b0-a4a61f8ddf10
14
15 #Property Name: platform
16 #A key specifying which platform the browser should be running on. Default is Linux.
17 #e.g. platform=Windows 8
18
19 platform=Windows 8.1
20
21 #Property Name: browser
22 #Browser to be used during execution. e.g. firefox, iexplore, googlechrome, safari
23 #e.g. browser=firefox
24
25 browser=Iexplore
26
27 #Property Name: version
28 #The browser version, or the empty string if unknown. If empty default version will be used.
29 #e.g. version=40.0.1
30
31 version=11
32

```

Fig:2.4.3.2-b

2.4.3.3 Offline Execution and Report

2.4.3.3.1 Execute Offline

- Once configuration settings are done, execute Offline Package. To execute it you can follow the traditional method of offline execution .For more details please refer section 1.2.2 above.
- Note that values for settings keys of Sauce.properties can also be provided from command line. To avoid inconsistencies with special characters(like <space>), provide key values in quotes as mentioned above.
- Overriding Sauce Properties:

There are 3 sources from where user can set the Sauce properties.

- Sauce.properties - This is default Sauce properties file comes with offline package in config folder. Properties values would be set to defaults. User can always use this file to set his/her own values.
- userdefinedsauce.properties - Qualita allows users to create their own custom properties file for Sauce which include sauce properties. To use this user has to specify its file path as command line argument while running offline package.
e.g. java -jar qualitia-offline-xxxx.jar saucedfilename=<<user defined sauce properties file path>>
- command line arguments - Qualitia allows users to pass Sauce properties as command line arguments like
e.g. java -jar qualitia-offline-xxxx.jar custom_username=<<sauce user name>> platform=MAC recordVideo=false

2.4.3.3.2 View Current Execution in Sauce

- As soon as you trigger Offline execution, you can validate the ongoing execution on Sauce UI. **Each test case in your suite will appear to be a single job in Sauce UI. Execution status of each test case can be seen in the Sauce UI.**
- Once execution of a single test case from suite(Sauce job) gets completed, Dashboard of Sauce UI will get updated with its status(fail/pass) and the next test case in sequence from the suite will be picked for execution e.g. If your suite contains 3 test cases 'Alert_Prompt', 'AllObjects', 'DownloadFile' in sequence, the very first test case 'Alert_Prompt' will be seen running in Sauce labs first. Once the execution is complete for the test case, its status will be updated to be pass or failed. 'AllObjects' can be seen running next and so on. Refer fig below.

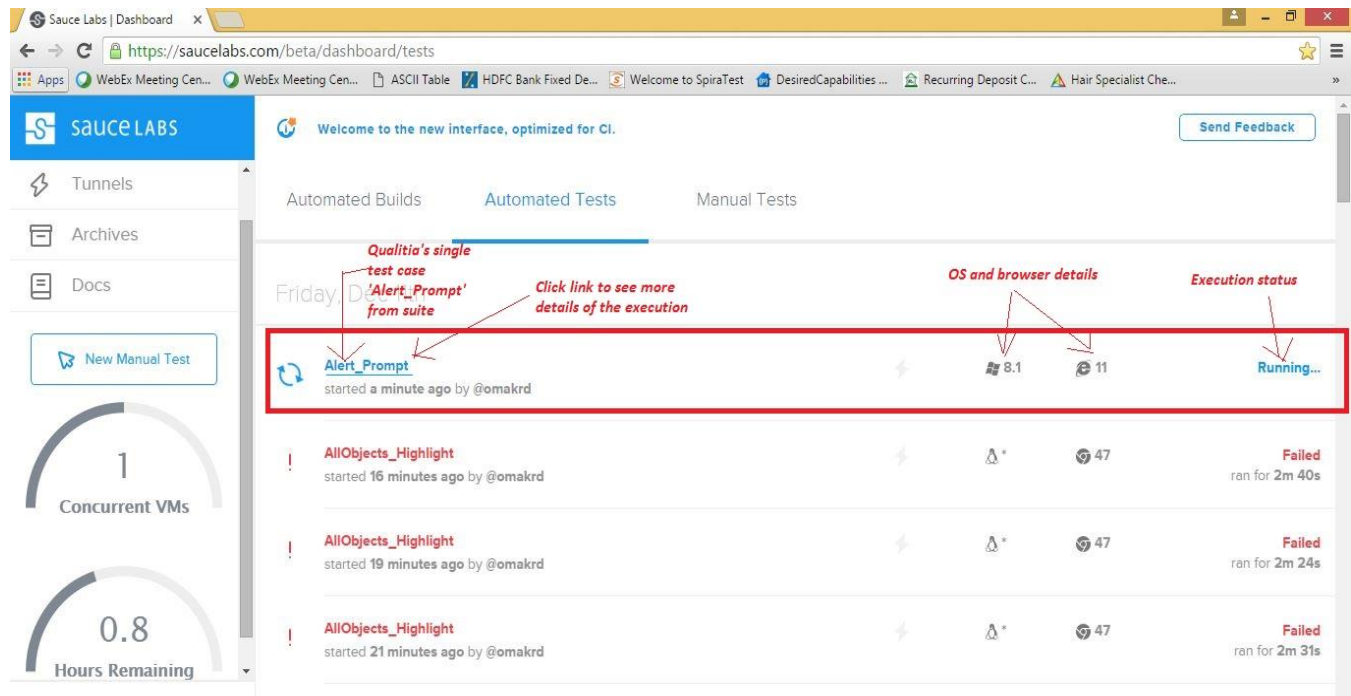


Fig:2.4.3.3.2 -a

- Clicking on the current job on dashboard of Sauce UI, details of the test case execution will be shown.
- If you wish Sauce to capture a video for your test case execution, set key recordVideo=true in Sauce.Properties file. Clicking on corresponding job link in Sauce UI, an ongoing video can be seen. To download video for the executed test case, go to 'Metadata' tab for the job once execution for the test case is complete. Refer fig below.
- Note that if 'QualitiaExecMode' property in Qualitia.Properties file is set to 'ps'(i.e. per suite), all the test cases will be executed under a single job in Sauce. Also the recorded video will be a single corresponding video for the whole suite execution against corresponding job.**

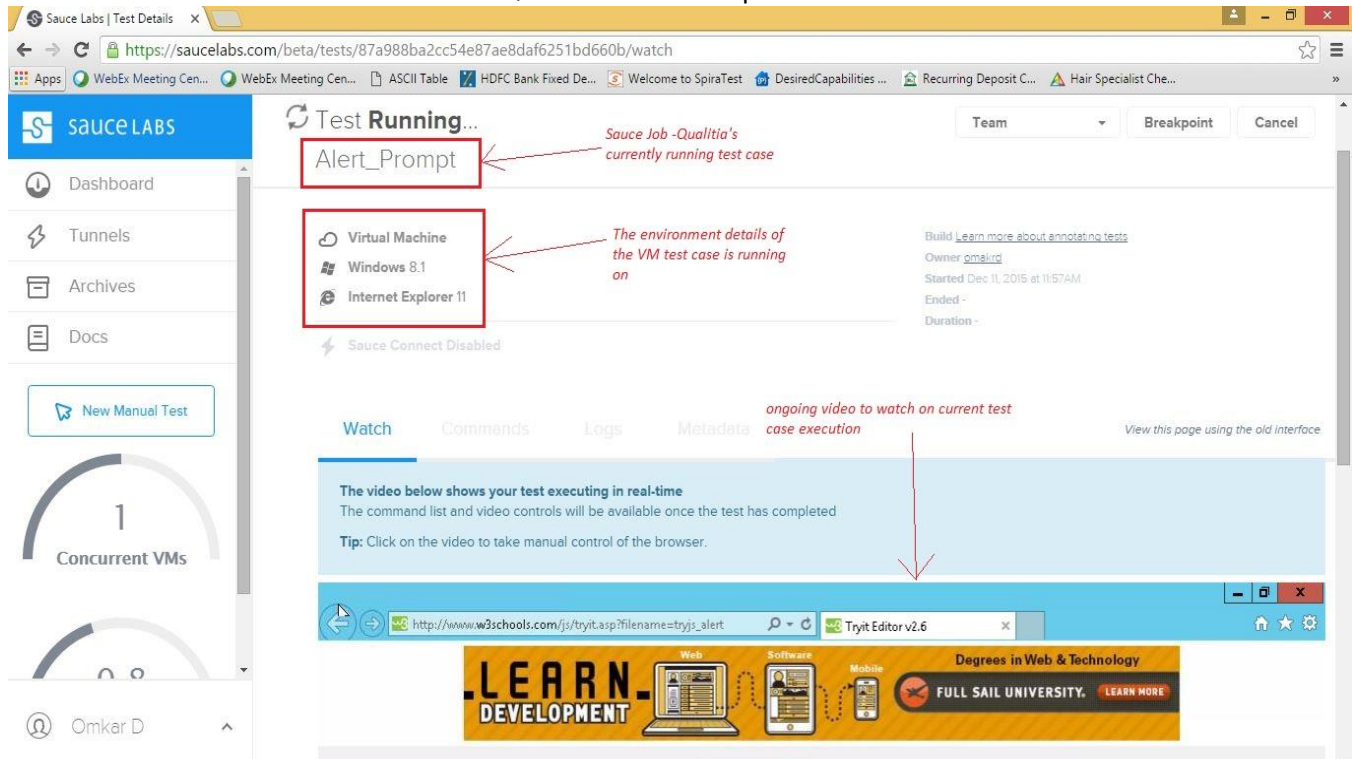


Fig:2.4.3.3.2 -b

2.4.3.3.3 Execution Reports

- Once the utter suite execution gets completed in Sauce, Test Execution Report will be launched on the machine from where the execution was triggered.
- To download video for the executed test case, go to 'Metadata' tab for a job and click on link 'Download' against captured video. Refer figure below.

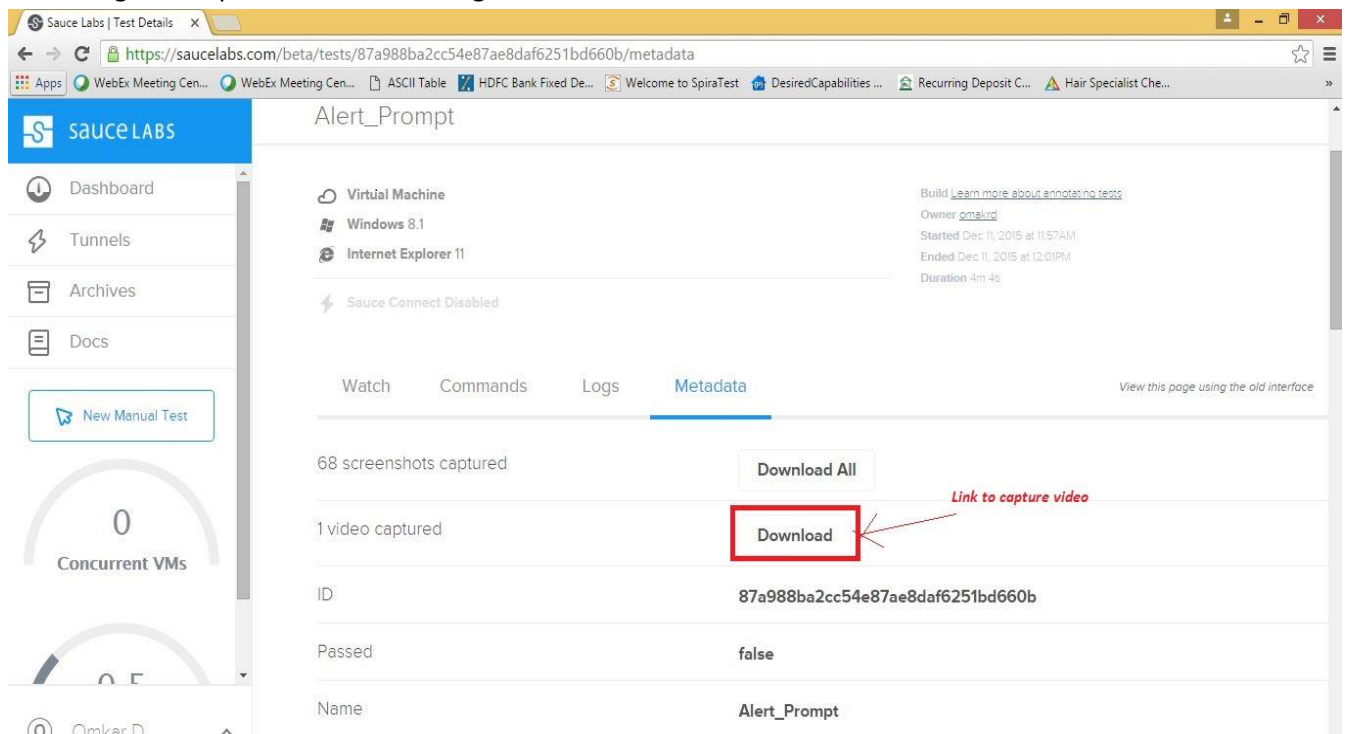


Fig: 2.4.3.3.3

2.4.3.3.4 Execution of tests Running on Behind Firewall Applications

Few applications may have been configured such that they are accessible only through private networks. Such applications deny access when are being accessed from other than local networks. Executing tests on such applications will definitely fail on Sauce. Refer fig below.

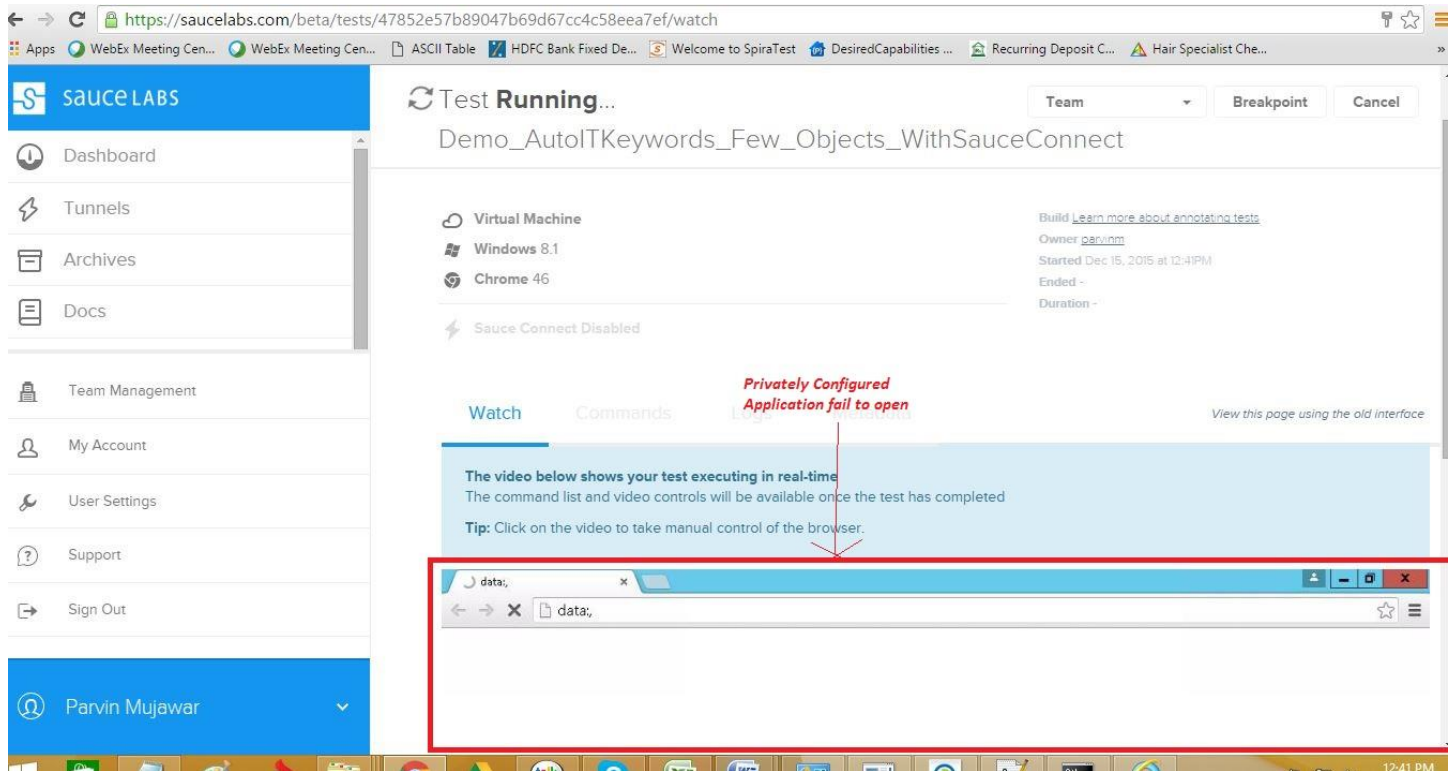


Fig: 2.4.3.3.4-a

Solution:

To execute tests on such applications, Sauce provides you a provision. Executing 'Sauce-Connect.jar' file on any machine in the private network(as follows) allows Sauce to execute your tests flawlessly.

Steps:

1. On any machine in the network(not necessarily machine where website is hosted) download the Sauce connect zip from following url.
<https://saucelabs.com/downloads/Sauce-Connect-3.1-r32.zip>
2. Unzip the folder at desired location. You will find Sauce-Connect.jar file.
3. Execute the downloaded Sauce-connect.jar through command prompt by passing Sauce Username and Access key as shown in figure below.
e.g. Desired location>java -jar Sauce-Connect.jar parvinm ab8fa4d8-8c1f-4904-8353-b6074fc744fb
4. Once the Sauce-Connect.jar is executed and successfully connected, you can execute your Offline tests on Sauce environment same as mentioned above. Execution will happen in Sauce as expected.

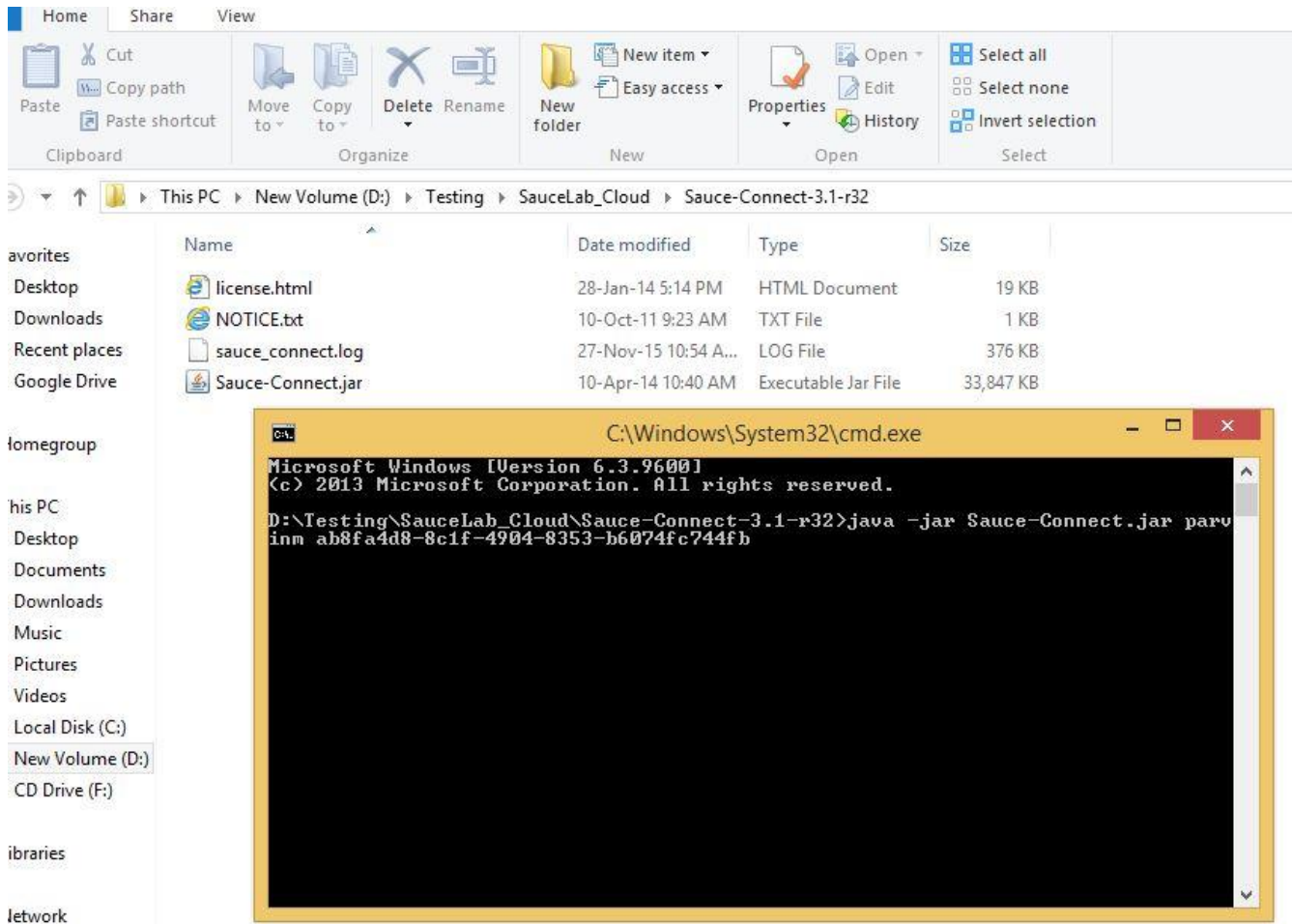


Fig: 2.4.3.3.4-b

2.4.3.3.5 Execution using Browser Capabilities

To execute test on Sauce Labs using selenium supported Browser Capabilities, refer section "Execution Using Browser's Capabilities".

2.4.4 Execution using Browser's Capabilities

Selenium allows you to set your desired browser specific properties for your execution browser. Qualitia also now allows you to set these properties for your Offline Execution. You can provide these browser specific details through browser-capability files. For each supported browser(Firefox, Iexplore, GoogleChrome, Safari) there is a separate property file present in the 'Config' folder of Qualitia's Offline package. v. z.

- firefoxdc.properties
- googlechromedc.properties
- iexploredc.properties
- safaridc.properties

Each above property file is a collection of all the browser specific properties that Selenium allows you to change.

If you do not change/provide settings for these properties, selenium will use its own default values for each property.

How to change Browser Capability?

- All the properties from all the Browser Capability files are initially commented except 'unexpectedAlertBehaviour'. It means Selenium uses its own settings for each of the key which has been commented in file.
- When you change the property value, it will override corresponding value (if any) provided in Qualitia.properties/Sauce.properties or then selenium's default value.
- To change a property/capability go to corresponding Browserdc.properties file and uncomment corresponding key by removing preceding #. You can do this for multiple keys in property file.
e.g. If you want to change 'version' property from capability file, remove preceding # and set it to desired version(say 41) shown in figure below.
- **Note:** All capability keys do not accept values from command line. Hence it is recommended to provide these values through property files.

```

1 ##### This file contains Firefox browser specific properties information #####
2 ##### To use properties uncomment the property by removing preceding # symbol from property name. Commented properties will
3 ##### Property keys are case-sensitive. It is highly recommended to do not change the case of keys, if changed then those
4 ##### There is no value validation from Qualitia. It is highly recommended that to set valid values to properties. If any
5
6 #Property Name: version
7 #The browser version, or the empty string if unknown. If empty default version will be used.
8 #e.g. version=40.0.1
9
10 version=41
11

```

Fig:2.4.4

Note that Qualitia Offline gives you just a provision to pass these properties to Selenium. Selenium has its own control on what value to use and how to reflect them in Execution Browser. For more details, you can visit:

<https://code.google.com/p/selenium/wiki/DesiredCapabilities>

<https://sites.google.com/a/chromium.org/chromedriver/capabilities>

2.4.5 Reports

Now Qualitia's Test Execution Reports will show OS and Browser with their corresponding versions on its summary page along with the Execution Environment used.

e.g. If you have chosen execution environment to be 'Sauce' and expecting execution to happen on Firefox 42 on Windows 8.1, Summary Report will show these details as follows.

QUALITIA

Test Suite : Regs_itrns_Conditional_EnvVar_OnError
 Execution Time : 0:3:53
 Start Time : 2015-12-11 14:02:22
 End Time : 2015-12-11 14:06:15

OS - Browser : Windows 8.1 42.0
Environment : sauce

Total Test Cases : 5
Passed Cases: 2 Failed Cases: 2
Not Executed: 1 Defects: 0

Test Case	Scenario Name	Manual TC ID	Status	Execution Time
Reg9_DataEncryption	Regression	Reg9_DataEncryption	✓	00:01:06.028
Reg2_ConditionalExpression	Regression	Reg2_ConditionalExpression	✓	00:00:46.417
Reg3_OnError_Continue_ExitTask_ExitTCIteration	Regression	Reg3_OnError_Continue_ExitTask_ExitTCIteration	✗	00:01:52.429
Reg4_OnError_ExitSuite	Regression	Reg4_OnError_ExitSuite	✗	00:00:16.874
Testing	Regression	Testing	✓	00:00:00.000

Fig: 2.4.5

Notes:

- For Execution through remote environment, OS version will not be shown.
- For execution through any environment on Linux, OS version will not be shown.
- For execution on local on Windows 8.1, the OS details will be shown as 'Windows NT(UNKNOWN)'.

2.4.6 Limitations Of Remote/Sauce Executions

1. Qualitia cannot support Java Robot based actions in Sauce execution environment as this is not supported by Sauce. Hence following actions will not get executed in Sauce/Remote environment.
 - AttachFile
 - DownloadFile
 - WindowsAuthentication
 - PressKeys

Problem:

Java Robot is not supported from Sauce side itself, due to this Qualitia also does not support this. Executing tests with such actions on Sauce/Remote environment will fail the action.

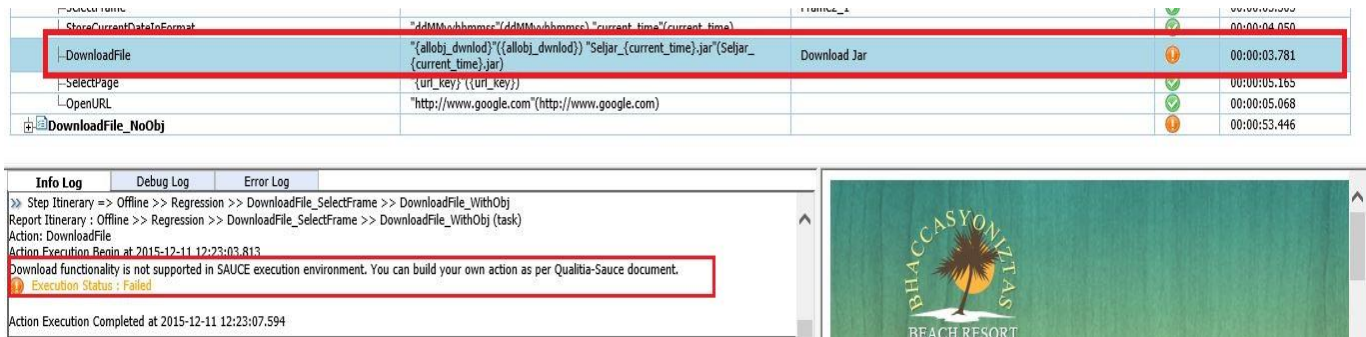


Fig:2.4.6

Solution

- It is highly recommended to user to validate tests that use the above action in local environment. In local environment user can continue to use the above actions without any issues.
 - User can create his own AutoIT scripts that can be executed on Sauce. Please refer section " Support for AutoIT in Sauce execution environment".
2. Qualitia cannot support Headless mode in Sauce execution environment as this is not supported by Sauce. If you want to do parallel execution in Sauce, you can do it by executing tests in multiple VMs. For each offline package execution one VM is allocated, so you can run multiple offline packages to get multiple VMs.
 3. Executing test case in Sauce environment on firefox browser by using user defined Firefox Profile is experimental.

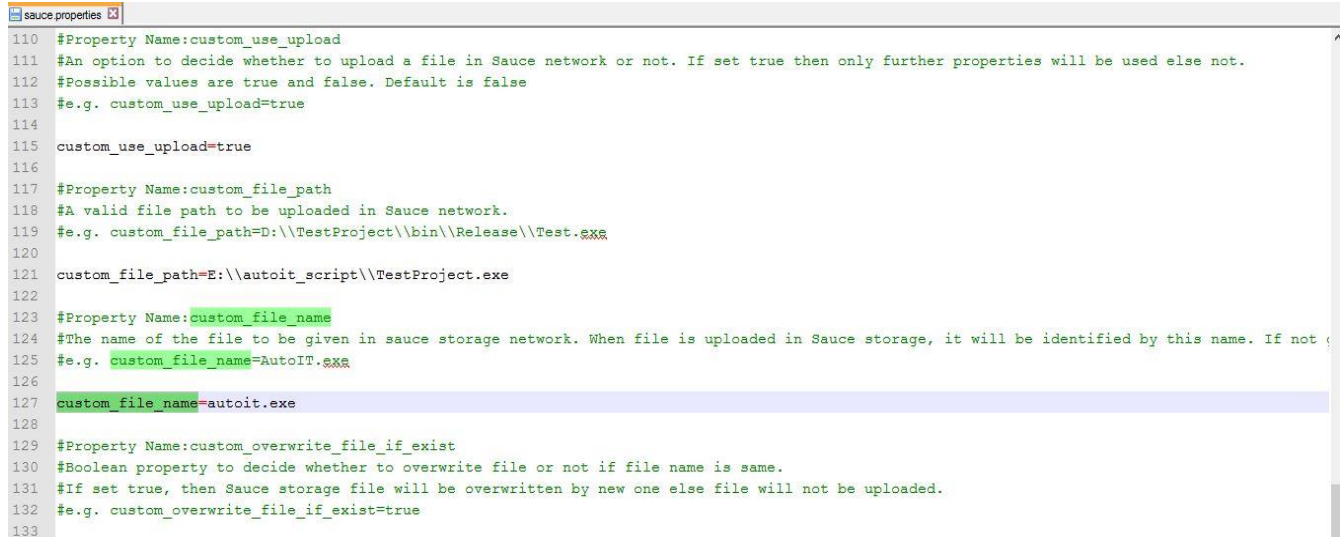
2.4.7 Support for AutoIT in Sauce execution environment

To accomplish Sauce execution of Windows based actions like DownloadFile, Press Keys, WindowsAuthentication, Qualitia supports "Sauce's pre-run" feature by which user is allowed to upload an AutoIT.exe (or any exe) and run before starting test execution. This exe would perform above actions.

2.4.7.1 Prerequisites and Configurations

To achieve this you have to set some properties in sauce.properties

- custom_use_prerun - An option to decide whether to use pre-run feature or not. If set true then only pre-run related properties will be used else not.
- custom_exe_name - The exe file name to be executed in Sauce environment. To run exe in Sauce this exe must be uploaded in Sauce network. To know how to upload an exe see how to upload file in Sauce network below.
- args- A list of the command line parameters that you want the executable to receive.
- custom_use_upload - An option to decide whether to upload a file in Sauce network or not. If set true then only upload related properties will be used else not.
- custom_file_path - A valid file path to be uploaded in Sauce network.
- On same line set all other properties for pre-run and uploading script file in sauce.properties file.



```

110 #Property Name:custom_use_upload
111 #An option to decide whether to upload a file in Sauce network or not. If set true then only further properties will be used else not.
112 #Possible values are true and false. Default is false
113 #e.g. custom_use_upload=true
114
115 custom_use_upload=true
116
117 #Property Name:custom_file_path
118 #A valid file path to be uploaded in Sauce network.
119 #e.g. custom_file_path=D:\\TestProject\\bin\\Release\\Test.exe
120
121 custom_file_path=E:\\autoit_script\\TestProject.exe
122
123 #Property Name:custom_file_name
124 #The name of the file to be given in sauce storage network. When file is uploaded in Sauce storage, it will be identified by this name. If not
125 #e.g. custom_file_name=AutoIT.exe
126
127 custom_file_name=autoit.exe
128
129 #Property Name:custom_overwrite_file_if_exist
130 #Boolean property to decide whether to overwrite file or not if file name is same.
131 #If set true, then Sauce storage file will be overwritten by new one else file will not be uploaded.
132 #e.g. custom_overwrite_file_if_exist=true
133

```

Fig:2.4.7

- To know more about details of how Sauce pre-run feature works visit:

Pre-run-

<https://wiki.saucelabs.com/display/DOCS/Test+Configuration+Options#TestConfigurationOptions-Pre-runExecutables>

Upload-

<https://wiki.saucelabs.com/display/DOCS/Temporary+Storage+Methods>

- **Notes:**
 - AutoIT only works on Windows platform. There is no support for other platforms like Linux, Mac.
 - User is responsible for writing AutoIT script and generating respective '.exe' file. Qualitia provides an example AutoIT script in "Help Doc" folder i.e. AutoITExampleScriptForDownload.au3 file. User can use this script as reference and create own script according to test.

2.4.8 Best Practices

1. When your execution environment is "Sauce" or "Remote", use ScreenCaptureMode to be "Browser".
2. If your application under test (on which Qualitia test cases are running) generates any runtime Alert/Windows, alter "FailStepOnBrowserScreenshotFailure" key value from Qualitia.Properties file to true or false based on test case need.

3 Cross platform execution

The same Qualitia offline jar can be used to execute tests on Windows, Linux and MAC against multiple browsers. For this, we will have to make changes to the properties file or provide arguments based on the execution platform.

3.1 Supported Platforms and Browsers

Support for browsers and platforms can be classified into

- Support for Execution
- Support for Report Viewing

3.1.1 Execution – Supported Browsers and Platforms

Here is a quick view of the browsers and platforms that you can execute your tests on, by using the offline package

	Headful (HeadlessMode = False)				Headless (HeadlessMode = True)			
	IE	Firefox	Chrome	Safari	IE	Firefox	Chrome	Safari
Windows	X	x	x					
Linux		x	x			x	Experimental	
Mac		x	x	X				

x: denotes support for the browser on a platform

Headful: This is the normal mode of execution where we have a box or a laptop with Display monitor, Keyboard and Mouse

Headless: In this mode executions are possible on Linux where you can have a box without the display, Keyboard and Mouse. Headless mode of execution is supported on Linux only. Such executions are known to be faster as compared to headful executions. There are multiple configurations and allied software components that are required to support executions in headless mode. We will see how to configure this later.

3.1.2 Report Viewing - Supported Browsers and Platforms

After the execution Qualitia will now use the “**Default Browser**” on the execution system to show the report, instead of always showing it on IE.

Here is a quick view of the browsers that can be used for Qualitia report viewing.

	IE	Firefox	Chrome	Safari
Windows	x	x	Not Supported	
Linux		x	Not Supported	
Mac		x	Not Supported	x

Chrome is not supported for report viewing. If Chrome is the default browser then you will need to copy the report URL and open in any of the supported browsers on the system.

4 Cross platform executions

Executing QOP on Mac is not much different from what we do on Windows, except for a making a few changes to the properties file or providing the correct arguments.

We should note the following things before we make changes to our properties file in QOP, especially if we are moving from Windows → Mac or Linux

1. All file and folder paths in Linux and MAC use “/” (forward slash) as a file separator
2. IE is not supported on Linux and MAC. Safari (MAC only), FF and Chrome are supported
3. Data provided to some of the Qualitia Actions will need changes to work in Linux and Mac

4.1.1 Qualitia System Variables

Qualitia system variables can be used to identify the execution environment using readily available System Variables like OS and Browser.

System Variable	Description
\$\$_ISMAC	Returns "TRUE" if execution OS is MAC, "FALSE" otherwise
\$\$_ISWINDOWS	Returns "TRUE" if execution OS is Windows, "FALSE" otherwise
\$\$_ISLINUX	Returns "TRUE" if execution OS is Linux, "FALSE" otherwise
\$\$_ISFIREFOX	Returns "TRUE" if execution Browser is Firefox, "FALSE" otherwise
\$\$_ISCHROME	Returns "TRUE" if execution Browser is Chrome, "FALSE" otherwise
\$\$_ISIE	Returns "TRUE" if execution Browser is IE, "FALSE" otherwise
\$\$_ISSAFARI	Returns "TRUE" if execution Browser is Safari, "FALSE" otherwise

Note: TRUE and FALSE are in Upper Case

4.1.2 Building cross platformtest cases

As we develop tests in Qualitia Test Automation Studio, our test executions are bound to happen on Windows. Though most tests will be ready for on Linux and MAC execution, there are some that might need changes.

In cases where steps need to be executed based on the OS, we can use "If Else" conditions along with the Qualitia System Variables.

Most of the changes are related to the actions that need file paths as parameters. If the following actions are used then we will need to make changes to the test case to make them compatible across platforms

ExecuteFile

For e.g. the **ExecuteFile** step can be changed in the following manner to work on Windows, Linux and Mac

```

If "$$_ISMAC"
    ExecuteFile "/Users/admin/scripts/myscript.sh"
Else If "$$_ISWINDOWS"
    ExecuteFile "C:\users\user1\myapp.exe"
Else If "$$_ISLINUX"
    ExecuteFile "/Users/admin/scripts/myscript.sh"

```

Similarly steps using the below actions can also be accommodated to work across multiple platforms

- DownloadFile
- ExecuteQueryAndStoreInFile
- CompareTabularResults
- AttachFile

Shown below is a sample of how this can be used in Qualitia

Add product [K:XML]									
Task		Step							
New Import Detach		+ TC Step Add Remove If-Else Up Down Save Save As Run Test Data							
Steps	Task	Condition	Object	Action	Parameter	Parameter	Parameter	Parameter	Parameter
		IF "\$\$_ISMAC==TRUE"							
				ExecuteFile	"Users/admin/Downloads/viewdata.sh"				
				ExecuteQueryAndStoreInFile	"com.mysql.jdbc.Driver"	"jdbc:mysql://dbServer:3306/QualitiaMaster"	"root"	"root"	"Select * from employee"
				AttachFile	"attach"	"link"	"Users/admin/Downloads/download1.bt"		"Users/admin/Downloads/download2.bt"
				DownloadFile	"download"	"link"	"Users/admin/Downloads"	"download1.bt"	
				CompareTabularResults	"Users/admin/Downloads/download1.bt"	"Users/admin/Downloads/download2.bt"	"BROWSER"		
		ELSE IF "\$\$_ISLINUX==TRUE"							
				ExecuteFile	"home/Qualitia/viewdata.sh"				
				ExecuteQueryAndStoreInFile	"com.mysql.jdbc.Driver"	"jdbc:mysql://dbServer:3306/QualitiaMaster"	"root"	"root"	"Select * from employee"
				AttachFile	"attach"	"link"	"home/Qualitia/Results/result1.bt"		"home/Qualitia/Results/result1.bt"
				DownloadFile	"download"	"link"	"home/Qualitia/Results"	"result1.bt"	
				CompareTabularResults	"home/Qualitia/Results/result1.bt"	"home/Qualitia/Results/result2.bt"	"BROWSER"		
		ELSE							
				ExecuteFile	"d:\ExecuteBatFile\viewdata.bat"				
				ExecuteQueryAndStoreInFile	"com.mysql.jdbc.Driver"	"jdbc:mysql://dbServer:3306/QualitiaMaster"	"root"	"root"	"Select * from employee"
				AttachFile	"attach"	"link"	"d:\results\result1.xls"		"d:\results\result1.xls"
				DownloadFile	"download"	"link"	"d:\results"	"result2.xls"	
				CompareTabularResults	"d:\results\result1.xls"	"d:\results\result2.xls"	"MICROSOFT"		
		END IF							

5 Parallel Execution

From Qualitia 3.9 QOP tests can be executed in parallel on the same system. Parallel executions can reduce the total execution time by using the utmost processing power of your system.

Executing multiple offline packages on the same system (Win/Linux/Mac) is as simple as initiating offline packages at the same time

e.g.

```
C:\users\abc>qualitia-Offline.xxxx.jar smoke.properties
```

```
C:\users\abc>qualitia-Offline.xxxx.jar smoke.properties BrowserType=Firefox ScreenCaptureMode=BROWSER
```

```
C:\users\abc>qualitia-Offline.xxxx.jar smoke.properties BrowserType=googlechrome ScreenCaptureMode=BROWSER
```

```
C:\users\abc>qualitia-Offline.xxxx.jar regression.propertiesScreenCaptureMode=BROWSER
```

All the above tests can be executed at the same time.

ScreenCaptureMode=BROWSER will ensure that snapshots captured during execution are independently done for each browser session separately.

Actions Not Supported:

Window sensitive actions like the following don't support parallel execution on the same system

- i. Press Keys
- ii. AttachFile
- iii. DownloadFile
- iv. WindowsAuthentication

Test cases using the above actions will fail when simultaneous executions are initiated on the same system.

6 Headless Executions

Qualitia supports headless executions on Linux. To execute tests we will need to configure the headless box to run against Qualitia.

1. Install Xvfb – X Virtual FrameBuffer

This software help perform all graphical operations in memory without having to show any screen output. In Qualitia this will allow our tests without displaying the browser on the screen. But because of Xvfb, we will be able to perform all UI operations and also capture snapshots of the execution for each step. For this reason, in headless mode its recommended to set “ScreenCaptureMode” property to BROWSER

To install xvfb run the following command,

```
sudo apt-get install xvfb
```

To run Xvfb we will also need a display (any unique number) which Xvfb can use. We will also have to turn the access control off. This is achieved by the following command

```
sudoXvfb :99 -ac
```

Once this is done we need to change the properties in QOP to execute tests heedlessly.

```
HeadlessMode=True  
DISPLAY=99  
ScreenCaptureMode=BROWSER
```

Here is the complete list of commands that we need to use to run QOP in headless mode

```
Xvfb :99 &  
  
export DISPLAY=:99  
  
java -jar qualitia-offline-xxxx.jar
```

Do not forget to include the arguments like properties file and any other arguments when executing qualitia-offline-xxxx.jar

7 Continuous Integration

7.1 Jenkins Installation

1. Download Jenkins and JRE6/JDK 1.6(if not already installed).You can also use JRE 7/ JDK 1.7
2. Install JRE6/JDK 1.6.
3. Go to *command prompt*.
4. Go till bin folder of JRE6/JDK 1.6.
5. Execute the command `java -jar "<Jenkins installation location>\jenkins.war" httpport=<port number>`.

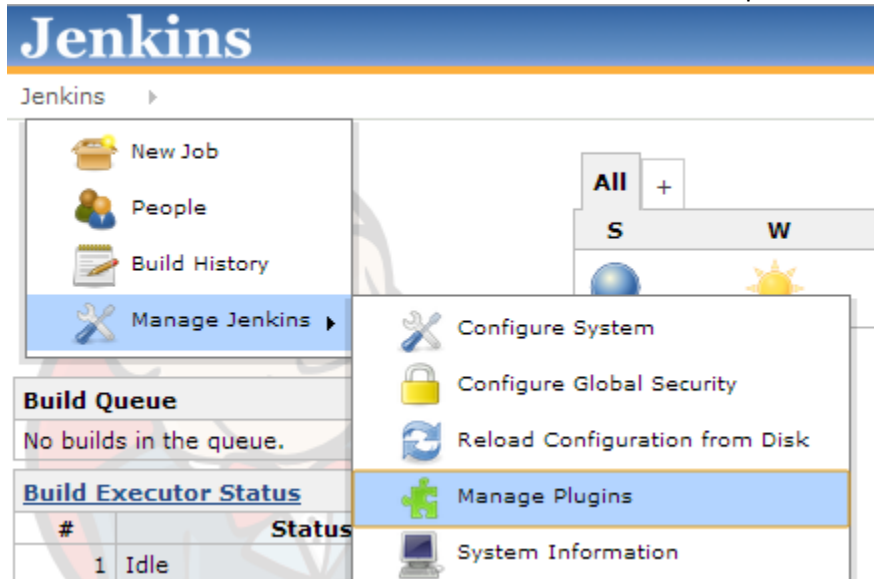
Default port is 8080.

```
10 Oct, 2013 11:41:38 AM winstone.Logger logInternal
INFO: HTTP Listener started: port=8080
10 Oct, 2013 11:41:38 AM winstone.Logger logInternal
INFO: Winstone Servlet Engine v0.9.10 running: controlPort=disabled
10 Oct, 2013 11:41:38 AM jenkins.InitReactorRunner$1 onAttained
INFO: Started initialization
10 Oct, 2013 11:41:40 AM jenkins.InitReactorRunner$1 onAttained
INFO: Listed all plugins
10 Oct, 2013 11:41:41 AM jenkins.InitReactorRunner$1 onAttained
INFO: Prepared all plugins
10 Oct, 2013 11:41:43 AM jenkins.InitReactorRunner$1 onAttained
INFO: Started all plugins
10 Oct, 2013 11:41:43 AM jenkins.InitReactorRunner$1 onAttained
INFO: Augmented all extensions
10 Oct, 2013 11:41:54 AM jenkins.InitReactorRunner$1 onAttained
INFO: Loaded all jobs
10 Oct, 2013 11:42:01 AM org.jenkinsci.main.modules.sshd.SSHD start
INFO: Started SSHD at port 58358
10 Oct, 2013 11:42:01 AM jenkins.InitReactorRunner$1 onAttained
INFO: Completed initialization
10 Oct, 2013 11:42:01 AM hudson.TcpSlaveAgentListener <init>
INFO: JNLP slave agent listener started on TCP port 58359
10 Oct, 2013 11:42:01 AM hudson.WebAppMain$2 run
INFO: Jenkins is fully up and running
```

6. Post successful execution of the command, you would see message *Jenkins is fully up and running*, open [URL:-http://localhost:8080](http://localhost:8080). This should open the home page for Jenkins.

7.2 Jenkins Configuration

1. The above URL launches Jenkins homepage.
2. Go to Jenkins → Manage Jenkins → Manage Plugins

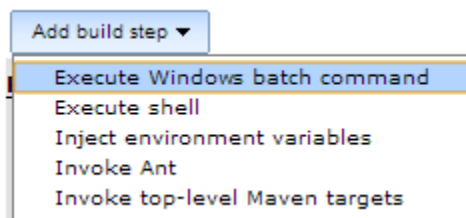


3. Install *HTML Publisher Plugin*, *Email-Ext Plugin* and *EnvInject Plugin*.
4. On Jenkins home page, go to *New Job*.
5. Enter the job name as per user understanding.
6. Select the first option

Build a free-style software project

This is the central feature of Jenkins. Jenkins will build your project, i build.

7. Click → Ok.
8. User will be directed to New Job configuration page.
9. Scroll down to *Build* section and select the first option.
10. Click on the button *Add Build Step* and select *Execute Windows batch command*.



11. The below screen appears.



12. Update the commands that lead to the location and execution of Qualitia Offline exe.
 Eg:-<drive name where offline package is saved>:\
 cd<location where offline package is saved>\Offline package
 qualitia-offline.xxxx.jar


```
d:
cd D:\Qualitia\offline package
qualitia-offline.xxxx.jar
```

13. From **Add Build Step** button, select *Inject Environment Variables*.

Note: This step has to be followed by step 12, otherwise Qualitia Reports will not be visible through Jenkins.

14. Update the path for the file *qer.properties* along with the filename. This file is located in the Offline Package.e.g:- <Offline Package location>\qer.properties

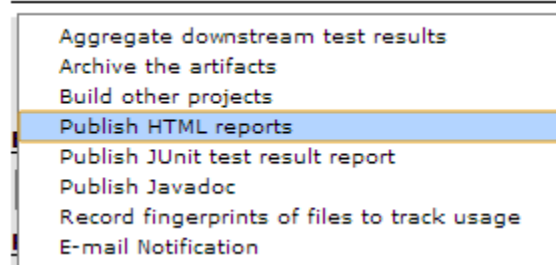
Inject environment variables

Properties File Path

15. Go to *Post build Actions* section.

16. Click **Add Post build action** and select *Publish HTML Report*.

Build Environment



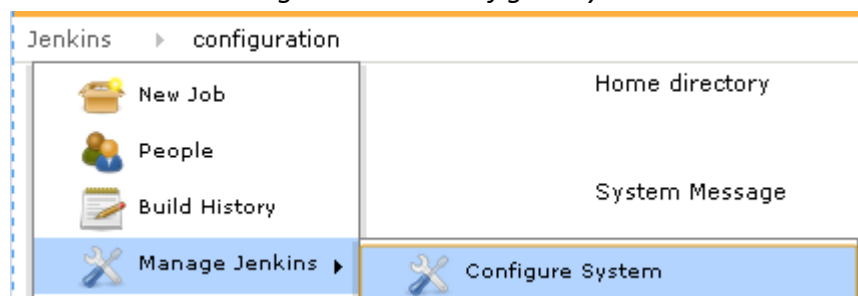
17. Below screen will appear. Update the values under respective columns as shown in the below screenshot.

Post-build Actions

Publish HTML reports

HTML directory to archive	Index page[s]	Report title	Keep past HTML reports	Allow missing report
\$qualitiareportpath	\$reportfilename	Qualitia Report	<input checked="" type="checkbox"/>	<input type="checkbox"/>

18. Select *Jenkins* → *Manage Jenkins* → *Configure System*.



19. Scroll down to *E-mail Notification* section.

20. Click → *Advanced* tab.

21. Enter valid values for the fields. Below screenshot is an example.

E-mail Notification

SMTP server	smtp.gmail.com
Default user e-mail suffix	@gmail.com
<input checked="" type="checkbox"/> Use SMTP Authentication	
User Name	amit.wadekar@qualitiasoft.com
Password
Use SSL	<input checked="" type="checkbox"/>
SMTP Port	100
Reply-To Address	noreply@qualitiasoft.com
Charset	UTF-8

22. Check the box for option *Test Configuration by sending test e-mail*, and update the email address.

23. Click → *Test configuration* button.

<input checked="" type="checkbox"/> Test configuration by sending test e-mail	
Test e-mail recipient	amit.wadekar@qualitiasoft.com
	<input type="button" value="Test configuration"/>

24. You would be notified on successful execution of step 23.

25. If there is any error message, recheck the setting values.

26. Scroll up to *Extended E-mail Notification* section.

27. Update correct setting values. Below screenshot is an example of what values are to be filled in the field.

Extended E-mail Notification

<input type="checkbox"/> Override Global Settings	
Default Content Type	HTML (text/html)
<input type="checkbox"/> Use List-ID Email Header	
<input type="checkbox"/> Add 'Precedence: bulk' Email Header	
Default Recipients	amit.wadekar@qualitiasoft.com
Reply To List	noreply@qualitiasoft.com
Emergency reroute	
Excluded Committers	
Default Subject	Qualitia Report Project - \$PROJECT_NAME Build # \$BUILD_NUMBER with Status - \$BUILD_STATUS has been completed
Maximum Attachment Size	
Default Content	<p>Hi All,</p> <p>Qualitia Report Project - \$PROJECT_NAME , Build # \$BUILD_NUMBER with Status - \$BUILD_STATUS has been completed</p> <p>To view Qualitia Report, access the below link:- \${PROJECT_URL}\${BUILD_NUMBER}/Qualitia_Report/?</p> <p>Thanks, Qualitia</p>

Note:- Multiple email addresses can be added by separating them using a ','. Subject and content of the email can be changes as per your understanding.

28. To enable this email notification for a Project/Job. Go to Project Configuration.

29. Scroll down to *Post Build Actions* section.
30. Click → *Add post-build action* and select *Editable Email Notification*.
31. Click → *Advanced Settings* button.
32. Click → *Add Trigger*.
33. Select the status as per your understanding, for which you wish to receive the email notification.
34. Click → *Save*.

7.3 Execute and View Report

1. Go to home page.
2. Click on the *Job Name*.
3. Click on *Build Now*.



4. Under *Build History* section user can see the progress of the build.

Build History (trend)	
#3	Oct 8, 2013 5:15:03 PM
#2	Oct 8, 2013 5:12:51 PM
#1	Oct 8, 2013 5:04:02 PM

5. User will also see execution of the TestCases involved in the *Offline Package*.
6. Once the build is successfully generated, click → *Qualitia Report*.
7. Qualitia Offline Package execution report will show up in the browser.

[Back to Qualitia report](#) [SummaryReport](#) [Zip](#)

Test Suite : suite1
 Execution Time: 0:02:1.243
 Run Start Time: 2013-10-8 17:12:59.966
 Run End Time: 2013-10-8 17:13:43.101

Total Test Cases : 2
 Passed Cases: 0 Failed Cases: 2
 Not Executed: 0 Defects: 0

Test Case	Scenario Name	Manual TC ID	Status	Execution Time
ObjSpy	s	ObjSpy	!	00:00:19.057
ObjSpy	s	ObjSpy	!	00:00:19.057
OnError	s	OnError	!	00:00:02.186
OnError	s	OnError	!	00:00:02.120
OnError	s	OnError	!	00:00:02.066

8. User can view earlier reports by selecting the respective Build and going to Qualitia Report of that build.
9. User can click on the zip link to download the zip of the Qualitia Report folder/files.

7.4 Parallel Execution with Jenkins

This approach works on Linux and will need the following plugins installed and configured

7.4.1 Install Jenkins Plugins

We will need to install the below plugins to help us execute tests in parallel using Jenkins

1. xvfb plugin for Jenkins (<https://wiki.jenkins-ci.org/display/JENKINS/Xvfb+Plugin>)
2. Build Flow Plugin for Jenkins (<https://wiki.jenkins-ci.org/display/JENKINS/Build+Flow+Plugin>)

7.4.2 Configure Build Flow plugin for Jenkins

Once both the plugins are installed

- a. Create a job with a suitable name. e.g. Smoke_Suite
- b. Select "Build Flow" from the options and click Ok. This should take you to the Configure section of the Build Flow plugin

Job name

☒ **Build a free-style software project**
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

☐ **MultiJob Project**
MultiJob Project, suitable for running other jobs

☐ **Build a maven2/3 project**
Build a maven 2/3 project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

☒ **Build Flow**
A Build Flow can manage job orchestration as a dedicated entity, in a centralized way with complex orchestration and without polluting the jobs with various plugins to handle the upstream-downstream chain.

☐ **Build multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

☐ **Monitor an external job**
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system. See [the documentation for more details](#).

☐ **Copy existing Job**
Copy from

- c. Now go to Build Triggers and add your script like the one below.

```
parallel (
  {build("Smoke_Firefox")},
  {build("Smoke_chrome")}
)
```

"Smoke_Firefox" and "Smoke_chrome" are qualitia offline jobs that are configured to execute on Firefox and Chrome respectively.

You can build multiple complex flows using this plugin. Please go through the Build Flow plugin URL to explore further.

7.4.3 Configure qualitia offline job

Now create two (or more) Offline Package jobs as shown in section 5.1 or use ones that already exist. Let's call them Smoke_Firefox, Smoke_Chrome. To execute these jobs in parallel you can do the following.

- i. Build Environment section in job "configure" and check the check box "Start Xvfb before the build, and shut it down after"

Build Environment

☐ Inject environment variables to the build process

☐ Inject passwords to the build as environment variables

☐ Run Xvnc during build

☒ Start Xvfb before the build, and shut it down after.

Xvfb installation Xvfb Plugin

Xvfb specific displayname

Timeout in seconds 0

Xvfb screen 1024x768x24

Xvfb display name offset 1

Xvfb additional options

Log Xvfb output ☐

Shutdown Xvfb with whole job, not just with the main build action ☐

Build

Execute shell

Command `cd /home/qualitia/Desktop/TestParallelExecution/OfflinePack/QualitiaOffline
java -jar qualitia-offline-3.9.1.jar sbrowsertype=googlechrome chromedriverpath=/home/qualitia/Desktop/chromedriver logpath=chromeResults ScreenCaptureMode=desktop`

[See the list of available environment variables](#)

Delete

Inject environment variables

Properties File Path /home/qualitia/Desktop/TestParallelExecution/OfflinePack/QualitiaOffline/qr.properties

Properties Content

Delete

- ii. In Build under Execute shell run the qualitia offline jar. Sample command is shown below

e.g.

```
cd /home/qualitia/QualitiaOffline

java -jar qualitia-offline-3.9.1.jar smoke.propertyssbrowsertype=googlechrome
ScreenCaptureMode=BROWSER
```

Note: ScreenCaptureMode=BROWSER is the recommended mode for parallel executions on the same system.

7.4.4 Execute Tests in Parallel

To execute tests in parallel we just need to execute the Build Flow ("Smoke_Suite") job that we created in 7.4.2

This will execute both Smoke_Firefox and Smoke_Chrome concurrently.

Once execution is done you can go to reports on the build flow job and check the results.

8 Team City

8.1 Team City Installation

1. Download Team City and JRE 6,7/JDK 1.6,1.7.
2. Install JRE 6,7/JDK 1.6,1.7.
3. Right click on Team City setup exe and click *Run as Administrator*.
4. Follow installation steps as per instructions in respective windows.
5. Post Team City installation, install Team City agent as well.
6. Go to *services.msc* and stop Team City agent service.
7. Open *command prompt* as administrator and browse through the path to *agent.bat* file.
8. Once the setup is complete, access Team City URL (<http://localhost:<port defined by user while installation>>).

8.2 Team City Configuration

1. Access Team City URL.
2. Create a new project.
3. Under *Build Configurations* section click → *Create build configuration* tab.
4. Update value for *Name* field as per user understanding.
5. Update field *Artifact Path* as `:-QualitiaReport/**/* => Report`.
Note:- DO NOT UPDATE ANY OTHER FIELD IN THIS SECTION.
6. Click → *VCS Settings*.
7. Click → *Create and attach new VCS root*.
8. Select *Type of VCS* as *Subversion*.
9. In the URL field, update SVN URL till parent folder name where Offline Package is uploaded (Offline.jar, all the other required files for Offline execution and XML folder are present in this folder).
10. Enter SVN credentials in the username and password field.
11. Click → *Test Connection*, to test SVN connection.
Note:- DO NOT UPDATE ANY OTHER FIELD IN THIS SECTION.
12. Click → *Add Build Step*.
13. Select *Runner Type* as *Command Line*.
14. Name step as you want. E.g. Qualitia_Offline
15. Click on *Run* dropdown, select *Custom Script*.
16. Update *Custom Script* field as `java -jar <qualitiaofflineJarFileName>.jar`
17. Click → *Save*.
18. Click → *Add Build Step*.
19. Select *Runner Type* as *Command Line*.
20. Select *Execute Step*, as per user understanding.
21. Click on *Run* dropdown, select *Executable with parameters*.
22. Update *Command executable* field as `%system.teamcity.build.workingDir%<followed by Offline Package path on SVN>`.
23. Click → *Save*.

Build Steps
In this section you can configure the sequence of build steps to be executed. Each build step is represented by a build runner and provides integration with a specific build or test tool. ⓘ

[+ Add build step](#)
[Reorder build steps](#)
[🔍 Auto-detect build steps](#)

Build Step	Parameters Description		
Qualitia_Offline	Command Line Custom script: java -jar qualitiaoffline-3.11.0.jar Execute: If all previous steps finished successfully	Edit	More ▾
Command Line	Custom script: xcopy /I /Y /E %env.qualitiareportpath%\... Execute: If all previous steps finished successfully	Edit	More ▾

24. Click → *Build Parameters*.



25. Click → *Add new Parameter*.

26. Select *Kind* as *Environment variable* from the dropdown list.

27. Update *Name* field as *env.qualitiareportpath*.

28. Click → *Build Step* (step 3).

29. Click → *Add Build Step*.

30. Select *Command Line* from the dropdown list.

31. Name the build step as per user understanding.

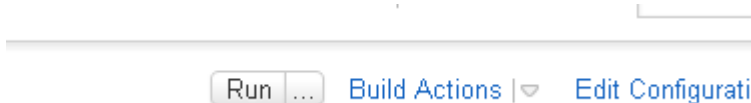
32. Click on *Run* dropdown and select *Custom script*.

33. Update *Custom Script* field as *xcopy /I /Y /E %env.qualitiareportpath%*.*\QualitiaReport*.

34. Save the settings.

8.3 Execute and View Report

1. Click → *Run* icon, present towards the right top corner.



2. Click on the build in progress, and go to build logs section to check the logs.

3. Once the build is complete, user would see *Build Finished* towards the end of the logs.

4. Click on *Artifacts* link to see the report in the browser.

5. Click on *SummaryReport.html* to vie

Administration > <Root project> > Qualitia > Build_Config

Build Configuration Settings

General Settings

Version Control Settings 1

Build Steps 2

Triggers

Failure Conditions

Build Features

Dependencies

Parameters 1

Agent Requirements

Last edited moments ago by admin (view history)

Build step settings updated.

Build Steps

In this section you can configure the sequence of build steps to be executed. Each build step is represented by a build runner and provides integration with a specific build or test tool. ⓘ

+ Add build step Reorder build steps Auto-detect build steps

Build Step	Parameters Description	Edit	More
Qualitia_Offline	Command Line Custom script: java -jar qualitiaoffline-3.11.0.jar Execute: If all previous steps finished successfully	Edit	More
Command Line	Custom script: xcopy /I /Y %env.qualitiareportpath%\... Execute: If all previous steps finished successfully	Edit	More

w the report.

Overview Changes Tests Build Log Build Parameters Artifacts

Reports

- Capture_Images
- Images
- Jquery
- js
- TC2_05fbc9b45c24417b97249618923be296
 - capturebitmap.html (292 B)
 - LogFrame.html (6.57 KB)
 - MergeXMLAndTxt.exe (9.0 KB)
 - Report.txt (165 B)
 - style.css (5.56 KB)
 - SummaryReport.html (4.82 KB)
 - SummaryReport.xml (1.16 KB)
 - SummaryReport.xslt (9.08 KB)
 - TC2_05fbc9b45c24417b97249618923be296.xml (937 B)
 - TestCaseDetail.html (10.64 KB)
 - TestCaseDetail.xslt (20.71 KB)
 - TestCaseReport.html (564 B)

- User can even pass arguments and execute same TC with different parameters on different agents simultaneously.
- To adjust the settings, user would need to go *Build Step* section.
- Edit the *Execute Parameter* step.
- Following the updated value in field, add the argument with respective data followed by **SPACE**.
- Below is an example of how to update the field value along with parameters and the syntax to be used, if the arguments value itself has *space*.

Ex:-<path updated in the field>CaptureFailBitMap=False CapturePassBitMap=False
seCreateErrorLog=False sbrowsertype=firefox "xmlPath=f:\\TeamcityOffline\\NewExec XMLs" ReleaseNum=2.01 ProjectName=offlinepack LogPath=argreport123

Note:- argument with value consisting space is updated within ' ' (double quotes).

