**Automation Script**

User Guide

|  |  |
| --- | --- |
| **Date of Publication 5/27/2014**  **Last Updated 5/27/2014** |  |
|  |  |
| **Abstract**  This document helps the end user in automation scripts to execute, analyze results, and troubleshoot issues. | **Version**  0.1 |
| Revision History   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Version | Date | Comment | Author | Reviewer | | 0.1 | 5/21/14 | Initial draft | Navneet Sharma |  | |  |

Table of Contents

[Purpose of This Document](#h.gjdgxs)

[Objectives](#h.30j0zll)

[High-Level Requirements - ‘What’](#h.1fob9te)

[Target Platforms](#h.78d42udss5c8)

[Architecture Design](#h.3znysh7)

[Configuration Files](#h.tyjcwt)

[Object Repository](#h.3dy6vkm)

[Test Case Data](#h.1t3h5sf)

[Test Cases Repository](#h.4d34og8)

[Result Repository](#h.2s8eyo1)

[Execution Log](#h.17dp8vu)

[Project Function library](#h.3rdcrjn)

[Common Function library](#h.26in1rg)

[Deliveries](#h.lnxbz9)

[How to Use](#h.35nkun2)

[Prerequisite (One time Steps Only – for a fresh machine)](#h.1ksv4uv)

[Steps to be followed](#h.44sinio)

[Changing Configuration from UI](#h.deplh6e25bem)

[Automation Script Releases](#h.2jxsxqh)

[Release 1.0](#h.z337ya)

[Release 1.1](#h.3j2qqm3)

[Release 1.2](#h.1y810tw)

[Release 1.3](#h.4i7ojhp)

[Release 1.4](#h.2xcytpi)

[Release 1.5](#h.1ci93xb)

[Release 1.6](#h.3whwml4)

[Release 1.7](#h.3nb0kn5290el)

[Release 1.8](#h.ga9isip508bo)

[FAQ and Trouble shooting](#h.2bn6wsx)

# Purpose of This Document

The purpose of this document is to help the end user’s of Automation scripts to execute, analyze the results and troubleshoot some common issues faced during the process.

# Objectives

Automation Scripts allows the manual testers to speed up the testing and reduce the manual efforts by performing the regular jobs in an automated fashion. Automation suite provides them the flexibility of running the test cases when they are not on their desk or not even in the office.

Organization needs the Automation to achieve the higher quality levels, reduce the turnaround time of the releases and make the life easier for the testers and the developers alike.

# High-Level Requirements - ‘What’

* End Users shall have a centralized system from where they can see what all test cases are available for execution.
* Shall offer the choice of execution of test cases.
* Shall show the current status update, which test case is being executed.
* Shall generate a summary results of total test cases executed, Total pass and Total Failed test cases.
* Detailed results shall be available for analyzing the failures happening in the process of execution.
* Logs shall be available to ensure that the failures are real failures of application and NOT scripts issues.
* Sanity Test suite shall be available
* Regression test suite shall be available (User shall have a choice what all he/she needs to run in regression)
* Coverage shall be as per the Manual Sanity test suites (mobile and online) and Manual regression test suites (Mobile and online).
* Test suites shall run on Mobile devices as well.

## Target Platforms

Automation scripts shall be able to run on

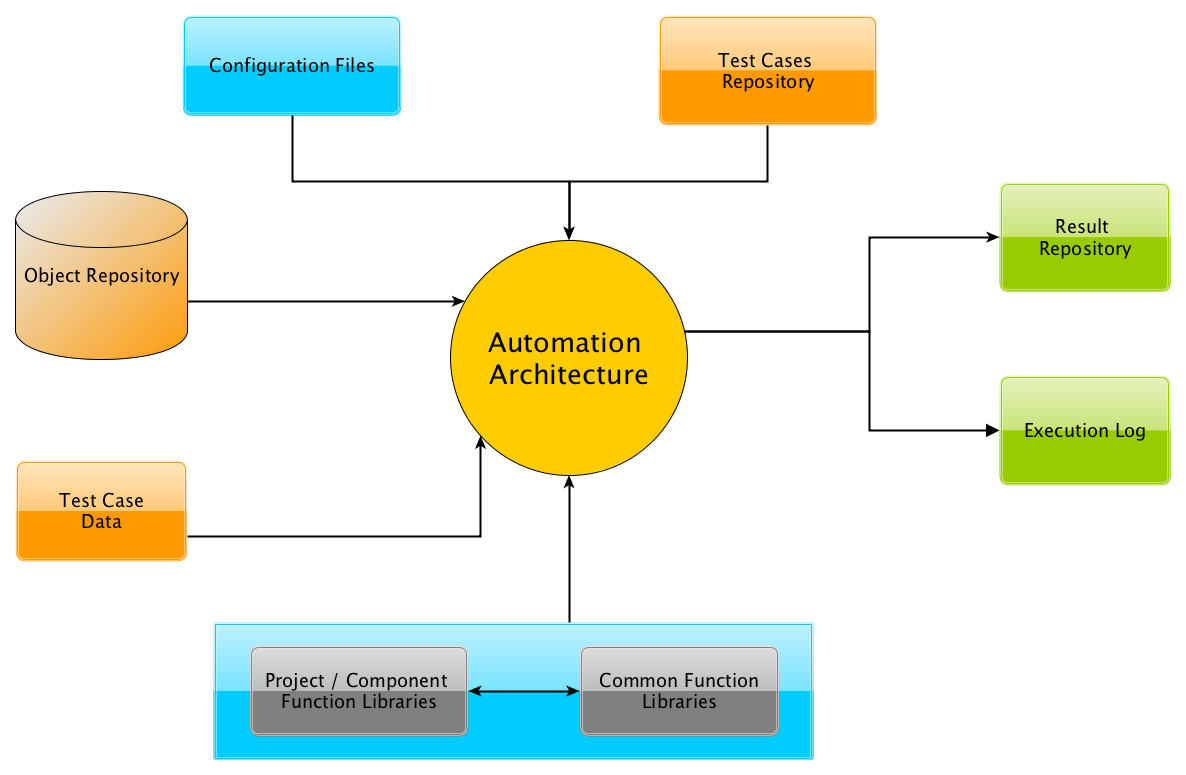
* MAC machines.
* Windows Machines

Shall support execution of scripts on

* Chrome Browser
* Firefox Browser
* Android Mobile Devices
* IOS Mobile Devices.

# Architecture Design

Automation architecture looks something like this. Various components are explained below.



**Figure 1 Automation architecture**

## Configuration Files

* This Folder contains all the configuration files needed by the system. Some of the parameters Which can be configured are like ‘URL to execute the test cases on’, ‘Account to be used to execute the campaigns from’, ‘Account in which the campaigns shall be created’ etc.

## Object Repository

* This repository stored the definition of the all the UI objects on the portal used by the automation scripts. If there is any change in the portal UI object definition than this repository needs to be changed. Currently the repository is maintained in form of Excel file.

## Test Case Data

* This Component takes care of the Data driven testing. If a test case needs to be executed for multiple data values or for some custom values than those values needs to be specified in the excel file kept in this location. This provides the scalability to the test suite in a major way.

## Test Cases Repository

* This component/folder takes care of storing all the test cases ( test scripts) in the form of .java files. Test Cases are stored based on the application components. Separate folders are maintained for each app component. E.g. Portal, Adserving, RTB etc.

## Result Repository

* This folder contains the execution results stored in the form of .xls file. Separate folders are maintained on the basis of Date and Time of execution. If in a single day multiple executions has been done than separate folders will be created. Single excel file contain separate sheets for each Suites. Result file also contains a summary sheet that will give the summary results of that execution cycle.

## Execution Log

* Execution logs are maintained on the basis of per day. One folder is created for each day and a separate file is created for each execution in the day bearing the timestamp of the execution. These logs gives the details of each and every action performed by the automation scripts and hence are the best mode of debugging what is happening at the back.

## Project Function library

* This section contains all those functions, which are common to the project, if the test scripts perform a certain action/Sub task of the application frequently than it is separated out as a method and is called from different suites. This gives the modularity and the code re-usage.

## Common Function library

* This contains all those function, which are not specific to the application under test but are commonly performed across the various suites of the automation scripts. e.g. Validation of existence of an object on the screen.

# Deliveries

Automation Scripts are delivered as a zip file via google shared drive. The location from where the latest scripts can be downloaded is <https://drive.google.com/a/vdopia.com/folderview?id=0B5YZvytgKlAbY0NPdUpFZk9WMGc&usp=sharing>

This location contains all versions released so far, User needs to choose the latest version available and download it on the local machine from which he/she wants to run the suite from.

# How to Use

This Section will describe how an end user can use the automation scripts using the Automation   
 test deck.

## Prerequisite (One time Steps Only – for a fresh machine)

1. Install Python (if python is not installed). Recommended python version is 2.7. \*. If older version of python is installed, then remove older version and install recommended version of python. Python installer is available:

<https://drive.google.com/a/vdopia.com/?usp=folder#folders/0B3LJExJq5Juva2w2Z1Q0QkRtdkk>

2. Install wxpython. wxpython installer is available at location (Used to create Automation Deck UI) <https://drive.google.com/a/vdopia.com/?usp=folder#folders/0B3LJExJq5JuvVG8xMTZaRXZ4TjQ>

3. Download the installer of java jdk from location:

[jdk7-downloads-1880260.html](http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html)

4. Download Android SDK for Mac and Windows. Setup is available at:

<https://drive.google.com/a/vdopia.com/?tab=mo#folders/0B2dXTwsmn7VRbml2eXpQbWlnd2c>

a. Extract the downloaded zip file.

b. Change the values of AndroidSDK\_Windows and AndroidSDK\_Mac in Config file (qaconf.properies - described in below Step:4) with the root directory path (in case of mac - xxx/android\_sdk\_mac/sdk and in case windows - xxx/android\_sdk\_windows/sdk ) of extracted zip.

5. Install and configure Appium for Mac using the document available at:

[SDK\_Prerequisite.docx](https://drive.google.com/a/vdopia.com/?tab=mo#folders/0B5YZvytgKlAbY0NPdUpFZk9WMGc)

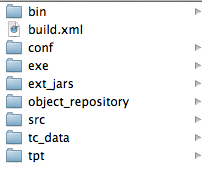
## Steps to be followed

1. Download the latest version of Vdopia\_Automation\_V\_<XXX>.zip file from:   
<https://drive.google.com/a/vdopia.com/folderview?id=0B5YZvytgKlAbY0NPdUpFZk9WMGc&usp=sharing>

2. Extract it at some location on your machine. It will create a folder by name:

**Vdopia\_Automation**

3. You will have the following folders inside it. :



**Figure 2 Folder Structure**

4. Go to Conf folder and open qaconf.properties to set values according to your need (Instructions are provided in the properties file). Save this file.



**Figure 3 Drill into 'Conf' folder**

Note: - Configuration can be changed from UI as well as explained later.

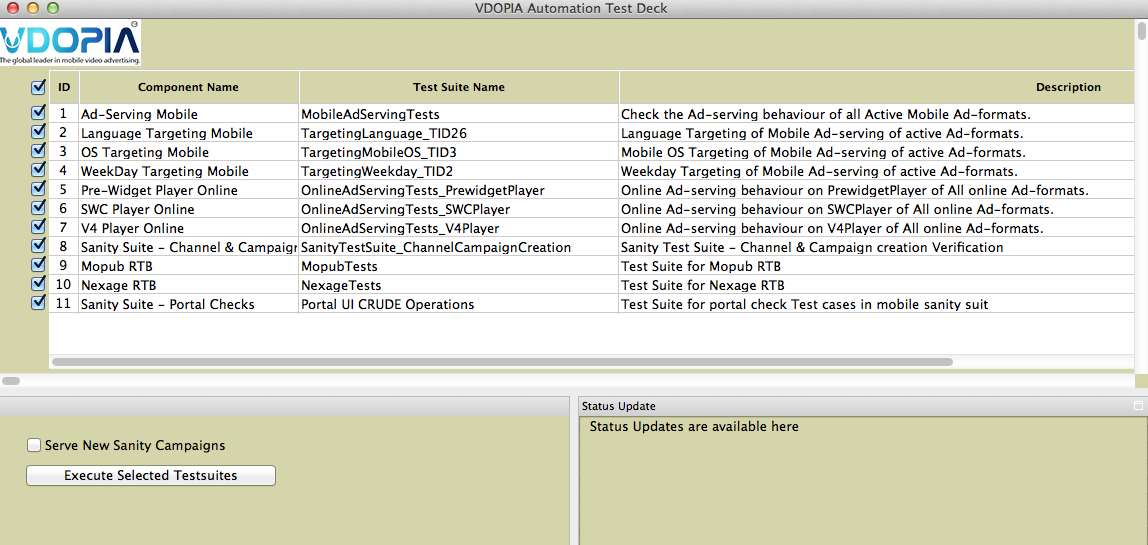
5. Go to exe folder and execute:



**Figure 4 Drill into 'exe' folder**

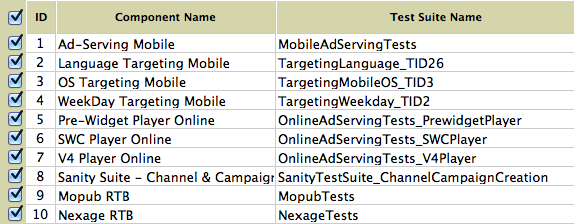
* automation\_gui.sh (For MAC)   
  + Open a terminal window and go to the path   
    <Download location>/Vdopia\_Automation/exe.
  + Type ./automation\_gui.sh and press Enter.
* automation\_gui.bat (for Windows) file. (Double click on automation\_gui.bat)

6. Automation Test Deck GUI will appear.



**Figure 5 Automation Test Deck GUI**

7. Select the list of Test Suites you want to execute.



**Figure 6 List of Test Suits**

If you want to perform ad-serving on newly created campaigns, then select the checkbox of the following screenshot



**Figure 7 Select Serve New Sanity Campaigns**

8.  After execution is complete, go to folder Results/Main\_Result\_<latest\_timestamp> folder.



**Figure 8 Drill into Results folder**

9. Open the xls file created with the latest date-time stamp and see the pass/fail results.

https://mail.google.com/mail/u/0/images/cleardot.gif10. To further debug the execution flow you can see the logfile created at location.

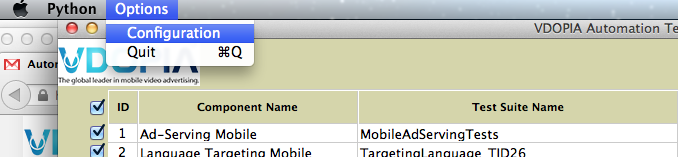


**Figure 9 Drill into logs folder**

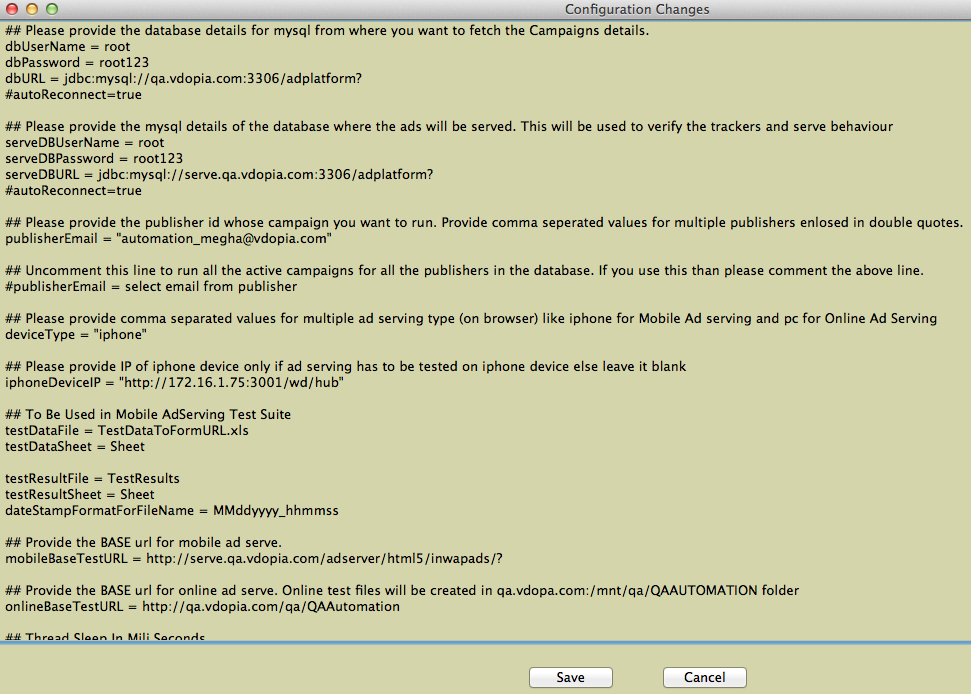
## Changing Configuration from UI

User can change the configuration of the Automation suite from UI as well.

Click on Options -> Configuration.



Configuration screen will open as shown



Perform the changes and click on Save button.

# Automation Script Releases

There has been number of releases done for Automation scripts till now. Below are the details of what all has been released in those versions.

## Release 1.0

1. Enable you to run all the active campaigns present on a specific server.

2. Enable you to execute campaigns of a single publisher. Multiple publishers or all publishers  
 present in the database.

3. Verify all the trackers fired by a campaign when it runs.

4. Give you a summarized result in XLS file along with the TAGs used to run the ad do that you can   
 easily debug the filed ones.

5. Highly portable and can be executed from any machine.

6. No Code knowledge needed (Batch file/Shell script execution).

## Release 1.1

1. Support of Targeting Verification

a. Language Targeting

b. Weekday Targeting

2. Optimizing test data generation

Normal Adserving behavior will not check for the targeted ads, it will test the universal targeted   
 ads only.

3. Log File for executions

Automation Scripts provide the log files in [Automation home]/log folder. The file will be created for each suite. Currently three suites are running.

## Release 1.2

1. Support of Online Ad Serving following players

a. V4

b. SWC

c. Prewidget

2. Campaign and Channel Side Tracker validation for following online ad formats:

a. Preroll  
b. Midroll  
c. Postroll  
d. Branded  
e. Overlay

3. Configuration Changes:

By default, test suite will run for both online and mobile ad formats.

## Release 1.3

1. Channel creation verification in Mobile portal

This release of automation creates and verifies the channel creation on mobile portal.

2. Campaign creation verification in Mobile portal

These automation scripts can be used for verification of campaign creation on mobile portal.   
Currently automation script supports the following as formats of mobile portal

* Animatedbanner
* Vast feed
* Interstitial (without campaign-ion banner)
* Preroll-Vast.

3. GUI deck for automation test suite execution

Automation scripts provide a GUI to user to control the list of test suites user wants to execute. After the execution is complete, result will be available in one single excel file.

## Release 1.4

1. Campaign creation verification in Mobile portal

These automation scripts can be used for verification of campaign creation on mobile portal. Automation script supports all ad-formats of mobile portal like:

* Animated banner Banner
* Vast feed
* Interstitial - without campaign-ion banner
* Interstitial - with campaign-ion banner
* Video - without campanion banner
* Video- with campanion banner Video Banner
* Leader Video
* Preroll –Vast
* Preroll- Video.

**2. RTB Automation for Mopub Ad-exchange**

Automation team has automated tests for Mopub RTB ad-exchange. Now we can perform end-to-end automation validation for Mopub ad-exchange. Areas covered in automation are Mopub ad-exchange:

* Response validation of Mopub RTB requests.
* Validation of channel creation of Mopub exchange.
* Campaign Ad-Serving on RTB Response.
* Tracker Validation of Campaign serving.

3. Enhanced GUI Deck for test suits execution

Automation team has updated the look and feel of GUI for automated test execution. Now we can control the configuration file using GUI deck.

## Release 1.5

1. Automation of Sanity Suite - Portal Check Test Cases  :

              Support for verification of portal UI test cases like :

* + - Verification of Publisher & Advertiser Profile tests.
    - Credit / Revenue update of Publisher & Advertiser Tests.
    - Priority Campaign Ad-serving test.
    - Frequency capping test cases.
    - Verification of Channel side tag generations.
    - Verification of Campaign test page serving.

2. RTB Automation for Mopub & Nexage Ad-exchange

Now we can perform end-to-end automation validation for ad-exchange. Areas covered in automation are Mopub ad-exchange:

* + - Response validation of Nexage & Mopub RTB requests.
    - Validation of channel creation of Mopub exchange.
    - Campaign Ad-Serving on RTB Response.
    - Tracker Validation of Campaign serving.
    - Verification of Win logs.

3. Mobile Ad-serving on Multiple Android Devices:

Now we can test ad-serving trackers (like cl, mu, umu, sk, pa etc…) on multiple Android devices. We just need to connect mobile devices.

4**. Automation of Portal UI – Targeting**

Targeting support while campaign creation for all mobile ad-formats. Now we can verify all the following targetings using automation scripts

* + - Operating System Targeting.
    - Weekday Targeting.‬‬
    - Connection Type Targeting.
    - Language Targeting.‬‬
    - OS Version Targeting.
    - Adult Content Targeting.
    - Device Capabilities Targeting.‬‬
    - Keyword Targeting.‬‬
    - Mobile Device Type Targeting.‬‬
    - Time Range Targeting.‬‬
    - Postal Targeting.‬‬
    - Location Targeting (lat./long).
    - Publisher Targeting.
    - Carrier Targeting.
    - Geographical targeting.
    - DMA Code Targeting.
    - Category Targeting.

## Release 1.6

1.    Automation of Sanity Suite - Portal Checks Test cases

Support for verification of portal UI test cases like

* Verification of UI of new Publisher reports (like Daily, Weekly, monthly etc...) flow.
* Changed the result generation procedure of portal check test cases.
* Automation verification of sanity tests of Ad-feeb & Global ad-feed.
* Automation verification of custom fields (like "ads-by" text) on Mobile Ad-serve (Video).
* Automation of Sanity suite encoding test cases.
* Automation verification of canvas url ad-serving.

2. Automation of Sanity Suite - Mobile Portal

Support of the following functionality in mobile portal UI verification.

* Creation of Preroll & Tag ad-formats on Mobile portal.
* Ad-serving automation of Tag ad-format.
* Ad-serving of Newly created Campaigns (created by automation scripts).

**3.** Bug fixes & Feedback implementation of Mobile Ad-serve

Support added for verification of click tracker in different ad-formats).

* + - Fixed ad-serving support for Vast Ad-format.

## Release 1.7

**1. SDK Media Player Ad-serving ( Only Mac ):**

The automation scripts can be used to verify ad-serving of SDK media player campaigns. This suite is performing the following actions :

* Support devices (like iPhone & iPad) & Simulators.
* Tracker validation of PreRoll ad-format campaigns.
* This suite compile & build SDK iOS app.

**2. Summary Report of Automation Execution :**

Automation team has added the functionality to create a consolidated report of Automation execution. Now a summary\_report sheet is added in main result which contains details like suites executed, total test cases, total pass, total failed test cases etc.

**3. YT Pre-Roll Support for Online :**

The Automation Scripts are now supporting YT Pre-Roll online Ad-format on online ad-serving.

**4. Mopub Version 1.0 :**

Automation team has added a new test suite for testing Mopub version 1.0 functionality. Now we can test bothversions of Mopub (2.1 & 1.0) as per the requirement.

**5. Re-Run of failed test cases:**

Automation Team has provided a new option to re-run all the failed test cases. On selecting this option, all the failed test cases of the automation execution, will be queued for rerun and updated result information will be available in main result sheet.

## Release 1.8

**1. Media Player SDK PreRoll Ad-serving (Only Mac):**

The automation scripts can be used to verify ad-serving of SDK media player campaigns. Key features of media player SDK automation are:

* Supported devices are iPad, iPhone, Android devices (4.X -version).
* Compile & build apk files (for android) & app files (for iOS) on runtime.
* Tracker validation of Pre-Roll ad-format campaigns.
* Support concurrent execution on multiple devices.
* Automation covers 70 percent of media player sanity tests cases.

**2. RTB Parameters validation tests :**

The Automation team has worked on RTB request and response parameter verification. By this automation, **90 percent of RTB regression test cases** of Mopub and Nexage are covered. Key features of this automation are :

* RTB parameter verification support for RTB 1.0, RTB 2.1 & Nexage exchanges.
* Can validate mandatory & optional parameters of RTB requests.
* Test suite is highly customizable using test data & configuration files.
* Response validation include mandatory keys existence in RTB response & adm data verification.

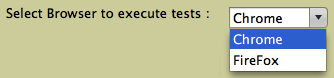
**3. LW-SDK for Mobile Ad-serving (Only Mac):**

The automation scripts can be used to verify ad-serving of all applicable Mobile ad-formats on iPhone & iPad :

* Supported devices are iPad, iPhone .
* Compile & build app files (for iOS) on runtime.
* Tracker validation of all applicable ad-format campaigns.
* Support concurrent execution on multiple devices.

**4. Support for browser compatibility in automation execution:**

Now automation GUI provides an option to execute automation suites on different browsers. Before execution, user can select any browser in the support browser list, to execute the automation test suites on the selected browser. By default, the testsuites will be executed on Chrome browser. Automation team will add other browsers in the drop down list in future.



# FAQ and Trouble shooting

**Question**: What is the preferred version of JAVA jdk for Automation scripts?

**Answer**: jdk 1.7. But Automation script also works well on other versions as well.

**Question**: How can I install Java Jdk in my system?

**Answer:** Download the installer of java jdk from location <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html> and follow the process of installation.

**Question**: Even after installing Java jdk, Automation script is giving message “javac   
 command not found”

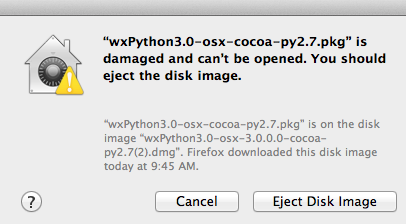
**Answer**: Javac executable need to be added in PATH variable. More detail is available at   
 location <http://www.java.com/en/download/help/path.xml>

**Question**: How can I give suggestion / feedback about automation scripts execution?

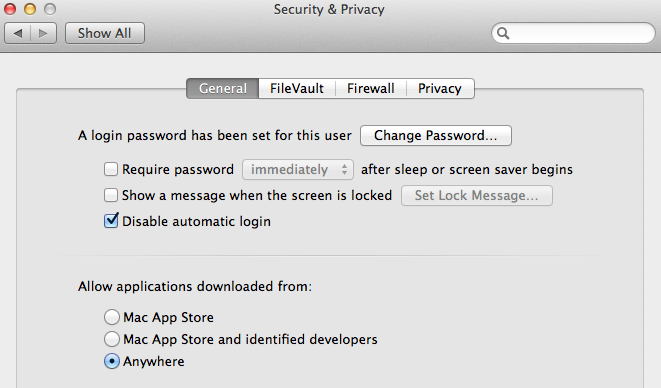
**Answer**: An excel file is created for collecting suggestions / feedback on Automation Release. Excel is available at location:

<https://docs.google.com/a/vdopia.com/spreadsheet/ccc?key=0AnLJExJq5JuvdE5ZcVlXSjYwWUpkNlpZeDFmaW5LZnc&usp=drive_web#gid=0>

**Question** : I am getting an error on MAC, What shall I do?



**Answer**: You need to change the Security settings in the system preferences.



Choose anywhere option as shown above.

**Question:** Scripts are unable to upload the media (video, banner etc.) and nothing seems to be happening on screen.

**Answer**: Probably Sikuli version has expired, you need to download (sikuli-setup.jar) from   
 <https://launchpad.net/sikuli/+download>. Follow this video link <Link of the video> to install this.