

# Eggplant Automation

Amit Kandoi

**Abstract**— This paper gives overview of Eggplant performance tool which is widely being used in the world of QE for evaluating performance of different mobile based products.

**Index Terms**— Introduction, Eggplant suite feature, Target market/feature, Stage of readiness, Useful feedback, Annexure



## INTRODUCTION

As technology is evolving, we were receiving different request for performance testing for different technology like mobile, cloud. Challenges for device-based testing in current engagement were

- Mobile Device automation was a known gap in the current project.
- Needed to capture Signature on device to complete the journey
- Needed to agree generated PDF files

We did proof of concept for Multiple tools, and later finalized eggplant suite (eggplant performance and functional) to conduct mobile / Ipad based flows.

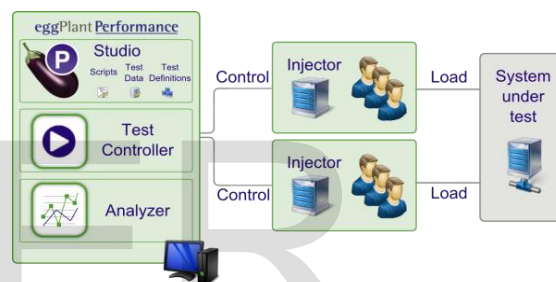
### Solution Overview

- Eggplant performance test solution is a fully automatable solution that addresses the unique challenges for Device (IPAD) Testing
- Eggplant solution provided key features which helped to complete End to End journey over Device for majority of business flows which was major revenue generating application

It has ability to simulate the real-life like load from multiple geographic locations. Eggplant Functional can be integrate into any continuous integration (CI) software or application life-cycle management (ALM) tool.

Eggplant provides inbuilt package ( package com.testplant.testing), which helps to capture signature and agree to generate PDF files.

High level figure for Eggplant performance tool.



Custom code for create PDF is given below

```
Url url1010 = new
Url(protocol1,opusmobile_pt_a4,
"/opus/createpdf");

QueryData queryData1010 = new QueryData();

queryData1010.add ("action",
"wirelesscustomeragreement");

queryData1010.add ("idx", "0");

queryData1010.add("_om_req_id",
getString("om_req_id") /* Rule: Correlate
om_req_id */ + ":-1636014098");

url1010 = url1010.withQuery(queryData1010);

try (Request request1010 =
getWebBrowser().createRequest(HttpMethod.PO
ST, url1010, 1010))

{
```

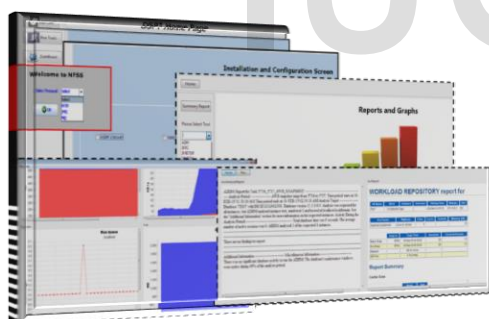
```
request1010.setHeader("User-Agent",
"OMIPAFILENAME/2108.7984 CFNetwork/1209
 Darwin/20.2.0");
request1010.setHeader("Content-Type",
"text/xml; charset=utf-8");
request1010.setHeader("Soapaction",
getString("url") +
"/opus/createpdf?action=wirelesscustomeragree
ment&idx=0&_om_req_id=" +
```

```
getString("om_req_id") /* Rule: Correlate
om_req_id */ + ":-1636014098");
String postDataString1010 = "PDF Request";
request1010.setRequestBody(postDataString1010
);
string postDataString1072 = "["+"\"Opening lean
RIA in OPUS\","+"\"RIA Request undefined\","
+ "\"Successfully generated RIA, Notifying
Native to open RIA PDF\\"" + "]"";
```

**EGGPLANT SUITE FEATURE:**

- Eggplant performance test solution is capable to capture E2E functionality over Device
- It captures response time for each navigation steps and can be verified with time taken for completing the journey in device for each step.
- It helps to capture images which helps to complete flows in which PDF and Bill summary pages are involved.

- Eggplant functional and Performance tools helped to simulate the exact journey of the flows involved which gives a clear understanding for all performance issues for particular transactions.
- Eggplant provide methods for connecting to a variety of iOS devices and iOS versions
- Following table provides glimpse where eggplant/functional tool can be use



Tool	Paid / Open Source	Native Apps	Web	Hybrid Apps	Android	IOS	Windows	BlackBerry	Library / Tool
EggPlant	Paid	Image Based	Image Based	Image Based	Y	Y	Y	Y	Tool

**STAGE OF READINESS**

- Solution is ready
- Been implemented in Device IPAD Testing
- Eggplant experienced skilled resources available.

**TARGET MARKET /CUSTOMER**

- Growing Market where majority of business is through any device (IPAD, Mobile) or technology, at DevOps speed, with user-centric test automation
- Performance testing under different network condition – from congested 3G to fast LANs

**USEFUL FEEDBACK FROM OPUS**

**STAKEHOLDERS**

- “Very helpful to do performance testing for highly volume business flows and emulate the real user experience till end of journey.

## ANNEXURE 1

### EGGPLANT PERFORMANCE TOOL

Eggplant Performance is a set of tools for load and performance testing of application clients, servers, networks, and network devices. By measuring performance at these different layers of a system under test (SUT), eggplant Performance provides a uniquely comprehensive analysis of both the user experience of an application and the capabilities of the supporting hardware, software, and network.

Eggplant Performance supports numerous protocols, communication interfaces, data formats and programming languages, enabling performance testing of many types of servers and distributed applications, including: Web servers, databases, EJB servers, Microsoft .Net services, SOAP, Citrix servers, mail servers, telnet services, mainframe systems, proprietary applications, or load testing by automating the user interface.

Eggplant Performance includes a number of powerful and easy-to-use tools that facilitate the creation of dynamic Virtual Users. These tools allow one to create complex, advanced tests where each Virtual User has its own unique data and timings. Additionally, groups of Virtual Users can run different scripts or workflows within the same test, simulating a variety of user stories. By acting as an engine for coordinating the execution of dynamic and variable Virtual Users, eggplant Performance simplifies the process of deploying realistic performance tests against a SUT and its environment.

The applications comprising the eggplant Performance set of tools are eggplant Performance Studio, Test Controller and Analyzer.

Studio is used to create the different components of a load test like workspace, scripts, workflows, test, data configurations.

The eggplant Performance testing environment consists of a controller machine, where tests are designed and initialized

Analyzer is then used to view completed test runs and their results, as well as generate a wide variety of charts and tables to analyze the data.

While recording and shakeout, we need eggplant functional to access the mobile devices (IPAD). It is mandatory to use eggplant performance and functional tool on same machine for recording the required functionality.

### FEATURES OF EGGPLANT PERFORMANCE TOOL

- Ability to filter/split data by injector/script/group
- Improved reporting, including PDF reports
- New comparison charts allow the user to compare multiple test runs and easily identify trends
- Powerful new chart creation
- Key Performance Indicators can be defined and reported on

### CREATING AN EGGPLANT PERFORMANCE TEST

#### 1. CREATE A WORKSPACE

Before you can create an eggplant Performance test, you must either open an existing Workspace or add a new one. The Workspace contains all environment-specific data such as injector details, default editors, compiler locations, and Custom Virtual User types.

#### 2. CREATE A VIRTUAL USER

Before you will be able to generate scripts for use in your test, you must have a Virtual User available.

Creating a new Virtual User type is done in preparation for script creation, but it is not necessary to do before creating each new script.

### 3. CREATE A PROJECT

Eggplant Performance is a project based load testing tool. Scripts, workflows, data and tests are all held within a named project. You must either open an existing project, or create a new one.

### 4. CREATE A NEW SCRIPT

Virtual User scripts can be created by using a script recorder, from a standard template, or by importing an existing source file. In these instructions we will be creating a Virtual User script from a recording.

### 5. GENERATE A SCRIPT FROM RECORDING

Recording a Web session simply creates a raw recording. This recording is processed by the eggplant Performance Web script generator to create the final Web script.

### 6. CREATE A WORKFLOW

It is possible to create a simple test consisting of one script directly, without using a Workflow. However, using a workflow gives you a greater degree of control over your scripts when running a test. In a workflow, scripts can be placed in 3 different categories: initial, repeated, final. Scripts placed in the initial section will only be run once at the start of a test. Scripts in the repeated section are iterated multiple times as specified in VU Group Runtime Settings, and final is the last script to be run once at the very end.

### 7. CREATE AND CONFIGURE A TEST

Designing a test allows you to configure how your Virtual User scripts will be used during the execution. A set of scripts can be reused and configured in different ways to allow an infinite number of different tests. We can clone, rename tests as per requirement for configuring multiple tests. We can maintain data configuration for different scripts very easily which helps us to execute tests effectively.

### WHY EGGPLANT PERFORMANCE

In one of the leading project for US telecom giant, said applications have "Device based" flows that need to perform load tests on mobile devices and for these applications we were using Load runner tool. However, some of the functionality in the application was not supported by load runner tool. The function was to "capture signature" which does not render on desktop machines. Eggplant performance tool easily capture the signature from device.

Secondly the reason to switch to eggplant tool due to the higher cost of license for load runner. Eggplant performance tool is cheaper than load runner and can simulate any load you need.

### REFERENCES

<https://eggplantsoftware.com>