ORACLE INTERNAL

# Visual Regression Tool

Created by Manoranjan Sahu, last modified on Sep 13, 2024

**Note: The latest version of this tool now automatically generates source code that includes basic UI automation tasks, along with an integrated image-capturing mechanism. As a result, some of the manual steps previously outlined may no longer be necessary.**

**The primary objective of this enhancement is to streamline the development process, allowing developers to focus on higher-level tasks by eliminating the need to write repetitive UI automation code.**


- 1.1. Overview
- 1.2. High level design
- 1.3. FAQ
  - 1.3.1. When not to use this tool ?
  - 1.3.2. How do I install VisualRegression tool?
  - 1.3.3. How to integrate VisualRegression tool in my existing Selenium Test Suite?
  - 1.3.4. Where does visualRegression.takeScreenshot save the screenshot of my pages ?
  - 1.3.5. How do I run the tool to capture the screenshots of my pages?
  - 1.3.6. How to capture test images ?
  - 1.3.7. Where is the test report stored ?
  - 1.3.8. How do I interpret the test report ?
  - 1.3.9. What is ImageDiff Column ?
  - 1.3.10. How do I control passing score ?
  - 1.3.11. Few of my testcases are actually fine even though the tool predicts it to be FAIL. How do I bypass this?
  - 1.3.12. Any other way ?
  - 1.3.13. I have got huge no of testcases and thus images. How can I generate a report for only certain testcases?
  - 1.3.14. How about generating reports by batches? Is there any way ?
  - 1.3.15. I have now generated many reports. How can I merge them ?
  - 1.3.16. Can I compare any two images collected outside Selenium ?
  - 1.3.17. How do I change report's header text ? For example, I want to add build details, server,etc.
  - 1.3.18. How do change the color of error diffs in the image ?
  - 1.3.19. Setting the wait time
  - 1.3.20. Regarding screenshot image capture
  - 1.3.21. I do not want certain area of my page to be used for comparison. How can I control that ?
  - 1.3.22. About command line usage
  - 1.3.23. Is it already merged to develop ?
  - 1.3.24. HOW TO UPTAKE VISUAL-REGRESSION IN YOUR UIT PROJECT
- 1.4. Related articles
  - 1.4.1. User Interaction Testing - UIT


## 1.1. Overview

What is Visual Regression Testing ?

ORACLE INTERNAL

- A visual regression test checks what the user will see after any bug fixes have been merged by comparing screenshots taken before and after code changes.
- For more details,please go through VisualRegressionDemo.pptx and VisualRegressionDemo.mp4 for the zoom recording that was presented in the VBCS meeting on 12/Nov/2021.

## 1.2. High level design

Please refer VisualRegressionDesign.pptx

## 1.3. FAQ

### 1.3.1. When not to use this tool ?

**Answer**

Do not use this tool if

- You are looking for writing asserts similar to JUnit or QUnit.
- Your code is not yet certified by QE. In other words,you do not have baseline images to compare against.

### 1.3.2. How do I install VisualRegression tool?

**Answer:**

Suppose you are in L:\planning\SeleniumTest\JetDashboardsTest directory, you can execute below command

npm install ../visual-regression

### 1.3.3. How to integrate VisualRegression tool in my existing Selenium Test Suite?

I have the following "describe" block in my file dsbCustomRangeTest.js. How do I use VisualRegression's API

**Existing Code** Collapse source

```
1
const webdriver = require('selenium-webdriver');
```

```
const driver = new webdriver.Builder().withCapabilities(webdriver.Capabilities.chrome
```

ORACLE INTERNAL

```
        seleniumUtil.openPlanningURL(done);
    });
    describe('Dashboard CustomRange Test', () => {

        it('Open Custom Range Popup', done => {
            seleniumUtil.openCustomRangePopup(done);
        });

        it('Expand Custom Range Popup', done => {
           seleniumUtil.expandCustomRangePopup(done);
        });

    });

    after(done => {

        driver.quit();
        done();
    });
});
```

**Answer**

**After integration with Visual Regression** Expand source

### 1.3.4. Where does visualRegression.takeScreenshot save the screenshot of my pages ?

**Answer**

After you have made the above code changes, you are expected to do the following steps.

1. If your project is L:\planning\SeleniumTest\JetDashboardsTest , open a GitBash there and create a file project.properties with the content as described in the codeblock.
2. sh ../visual-regression/shell/runVisualRegression.sh initSuite customRangeTest
3. The tool will create a directory 'customRangeTest' in the PROJECTDIR (L:\planning\SeleniumTest\JetDashboardsTest) and creates customRangeTestSuite.json

**project.properties** Collapse source

```
#########################################################################
```

```
#This is the configuration file which contains #
#what selenium scripts that need to be executed.
```

ORACLE INTERNAL

```
# #
#############################################################################
#This is your project directory where the selenium scripts lie
PATH_TO_VISUAL_REGRESSION=../visual-regression
PROJECTDIR=../jetDashboardsTest
```

### 1.3.5. How do I run the tool to capture the screenshots of my pages?

**Answer**

sh ../visual-regression/shell/runVisualRegression.sh collect customRangeTest dsbCustomRangeTest

Here dsbCustomRangeTest is the entry you would have made in the scripts node of package.json of your project. This is shown in the below codeblock.

Collapse source

```
{
  "name": "UIT",
  "version": "1.0.0",
  "description": "Selenium test for Dashboards and Infolets",
  "main": "pov bar",
  "scripts": {
    "dsbCustomRangeTest": "mocha --timeout  1000000 js/**/dsbCustomRangeTest.js --rep

  },
  "author": "Oracle",
  "license": "ISC",
  "devDependencies": {
    "gulp": "^4.0.0",
    "mocha": "^9.1.3"
  },
  "dependencies": {
    "canvas": "^2.8.0",
    "chai": "^4.3.4",
    "chai-as-promised": "^7.1.1",
    "chai-http": "^4.2.0",
    "chai-webdriver": "^1.2.0",
    "mocha-simple-html-reporter": "^1.1.0",
    "selenium-webdriver": "^3.6.0",
    "visual-regression": "file:../visual-regression"
  }
}
```

The tool will store 5 images for each page in the directory customRangeTest/data/training. So there will be 5 screenshots for 'Open Custom Range Popup' and 'Expand Custom Range Popup'

ORACLE INTERNAL

## 1.3.6. How to capture test images ?

I have now made some bug fixes in source code which impacts custom range popup UI. I want to run the visualRegression tool to capture the images in test mode?

**Answer**

sh ../visual-regression/shell/runVisualRegression.sh test customRangeTest dsbCustomRangeTest

Like in collection mode, the tool will store 5 images for each page in the directory customRangeTest/data/testing. So there will be 5 screenshots for 'Open Custom Range Popup'

and 'Expand Custom Range Popup'. Please note collection and testing images are stored in customRangeTest/data/training and customRangeTest/data/testing folders respectively.

## 1.3.7. Where is the test report stored ?

**Answer**

Given that you have now run the tool in collection and test mode, you can generate the reports using the below command.

sh ../visual-regression/shell/runVisualRegression.sh generateReport customRangeTest

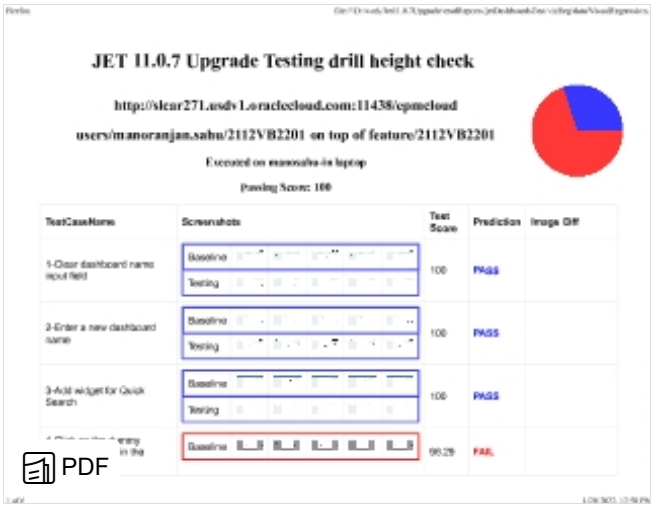
customRangeTest----> Name of the testsuite.

This generates the report for images captured in testsuite 'customRangeTest'

## 1.3.8. How do I interpret the test report ?

**Answer**

The following block explains the report



ORACLE INTERNAL

Please refer VisualRegressionDemo.pptx or VisualRegressionDemo.mp4

### 1.3.10. How do I control passing score ?

**Answer**

You can set the tolerance level by setting SCREENSHOT_ERROR_VECTOR_SPACE_TOLERANCE_PERCENT in the customRangeTestSuite.json present in your suite folder.

If it is set to 5, then any score greater than equal 95 will be considered as PASS by the VisualRegression Tool.

### 1.3.11. Few of my testcases are actually fine even though the tool predicts it to be FAIL. How do I bypass this?

**Answer**

Yes , you can execute the following command to do so.

sh ../visual-regression/shell/runVisualRegression.sh learnFromUser customRangeTest1,2,3

customRangeTest----> Name of the testsuite.

1,2,3 --→These are the TestCaseName sequences which you want to override.

### 1.3.12. Any other way ?

**Answer**

Alternatively , you can use the forbidden region feature of the tool. This is explained in the following codeblock. Please look the description of the node SCREENSHOT_FORBIDDEN_REGION

Collapse source

```
{
    "TESTSUITE": "Visual Regression Testing", //Enter a meaningful name for this test
    "SERVER": "ChangeMe", // Server URL against which test was exeecuted
    "BUILD": "ChangeMe", // Build info
    "COMMENT": "ChangeMe",// Any other comment
    "PROJECTDIR": "../jetDashboardsTest",// Do not change this
    "BASEDIR": "sampleTest", // Do not change this
    "EXECUTION_MODE": "generateReport", // Do not change this
    "NO_OF_SCREENSHOT": 5, // No of screenshots to take
    "SCREENSHOT_ERROR_VECTOR_SPACE_DIMENSION": 25, // Set this to square of NO_OF_SCR
    "SCREENSHOT_ERROR_VECTOR_SPACE_USE_HIGHEST_BASIS": true,
```

```
    "SCREENSHOT_ERROR_VECTOR_SPACE_TOLERANCE_PERCENT": 0, // Tolerance score
    "SCREENSHOT_ERROR_IMAGE_PIXEL_COLOR": {
```

ORACLE INTERNAL

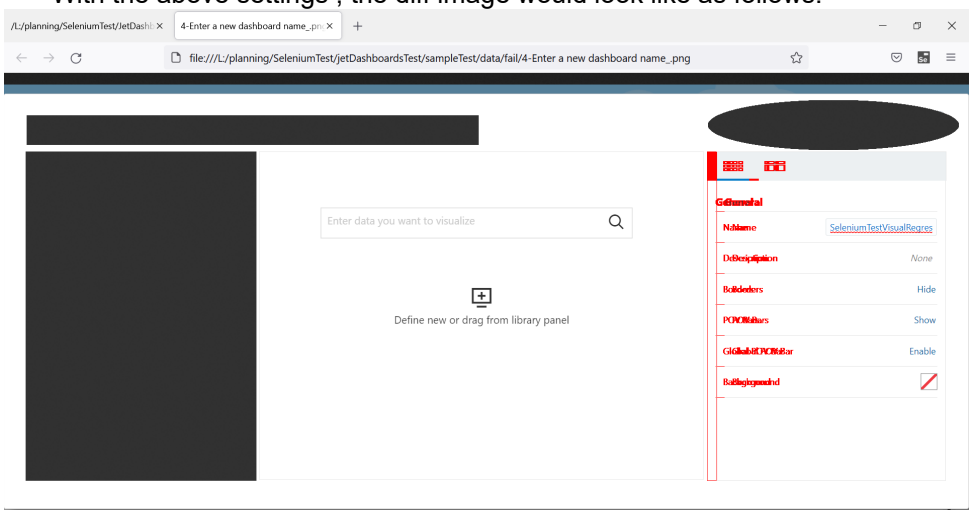
```
            "a": 255
        },
        "SCREENSHOT_FORBIDDEN_REGION": {
            "APPLY": true, // Set this to true. If it is set to false, forbidden feature
            "PIXEL_COLOR": {
                "r": 0,    // Color of the forbidden region
                "g": 0,
                "b": 0,
                "a": 200
            },
            "REGIONS": [{ // These regions are global. Thus it applies to all testcases o
                    "type": "Rectangle",
                    "apply": true, // Set this to true and use appropriate coordinates to
                    "metaData": {
                        "topLeftX": 45,
                        "topLeftY": 60,
                        "width": 900,
                        "height": 60
                    }
                }, {
                    "type": "Rectangle",
                    "apply": false,
                    "metaData": {
                        "topLeftX": 1360,
                        "topLeftY": 220,
                        "width": 550,
                        "height": 580
                    }
                }, {
                    "type": "Rectangle",
                    "apply": true,
                    "metaData": {
                        "topLeftX": 43,
                        "topLeftY": 132,
                        "width": 460,
                        "height": 655
                    }
                }, {
                    "type": "Ellipse", // With this circle can be constructed by setting
                    "apply": true,
                    "metaData": {
                        "centerX": 1650,
                        "centerY": 80,
```

```
                    "semiMajorAxis": 250,
                    "semiMinorAxis": 50
```

ORACLE INTERNAL

```
                "type": "Polygon",
                "apply": false,
                "metaData": {   // Enter the vertices coordinate in anticlockwise man
                    "vertices": [{
                            "x": 100,
                            "y": 100
                        }, {
                            "x": 300,
                            "y": 400
                        }, {
                            "x": 500,
                            "y": 600
                        }]
                }
            }],
        "PAGES": [] // Forbidden regions can be defined at testcase level too. This t
    },
    "WAITING_FOR_SCREENSHOT": 5000, // Amount of time in milliseconds to wait before
    "SCREENSHOT_INTERVAL": 500, // Amount of time in milliseconds to wait between two
    "MULTI_PART_SIZE": 7 // batch size of the chunk of testcases to be used for multi
}
```

With the above settings , the diff image would look like as follows.




## 1.3.13. I have got huge no of testcases and thus images. How can I generate a report for only certain testcases?

### Answer

The following command does the job.

sh runVisualRegression.sh generateRangeReport customRangeTest 2 5

ORACLE INTERNAL

2 ----> Start Index of the testcase

5 ----> End Index of the testcase

It will create a report for the interval [ 2, 5 ]

### 1.3.14. How about generating reports by batches? Is there any way ?

**Answer**

Yes, execute the following command.

sh ../visual-regression/shell/runVisualRegression.sh generateMultiReport customRangeTest 3

customRangeTest----> Name of the testsuite.

This generates the report for images captured in testsuite 'customRangeTest'

3 ----> Denotes the chunk of the testcases which should be used for reports

Each chunk size is given by MULTI_PART_SIZE property

### 1.3.15. I have now generated many reports. How can I merge them ?

**Answer**

Try the following command.

sh ../visual-regression/shell/runVisualRegression.sh mergeReports customRangeTest model_1.json,model_2.json

customRangeTest----> Name of the testsuite.

This generates the report for images captured in testsuite 'customRangeTest'

model_1.json,model_2.json ----> These are the models that were generated using generateReport verb in previous steps.

These model files would be present in customRangeTest/data folder.

### 1.3.16. Can I compare any two images collected outside Selenium ?

**Answer**

Yes, you can. Execute the following command.

```
sh ../visual-regression/shell/runVisualRegression.sh compareImages customRangeTest img1.png img2.png diff.png
```

ORACLE INTERNAL

The diff img would be saved as diff.png in customRangeTest' folder

### 1.3.17. How do I change report's header text ? For example, I want to add build details, server,etc.

**Answer**

The following block explains that.

**customRangeTestSuite.json** Collapse source

```
{
    "TESTSUITE": "Explain what this suite does",
    "SERVER": "Provide the server details",
    "BUILD": "Provide build details",
    "COMMENT": "Any more comments",
    "PROJECTDIR": "../jetDashboardsTest",
    "BASEDIR": "sampleTest",
    "NO_OF_SCREENSHOT": 5,
    "SCREENSHOT_ERROR_VECTOR_SPACE_DIMENSION": 25,
    "SCREENSHOT_ERROR_VECTOR_SPACE_USE_HIGHEST_BASIS": true,
    "SCREENSHOT_ERROR_VECTOR_SPACE_TOLERANCE_PERCENT": 0,
    "SCREENSHOT_ERROR_IMAGE_PIXEL_COLOR": {
        "r": 255,
        "g": 0,
        "b": 0,
        "a": 255
    },
    "SCREENSHOT_FORBIDDEN_REGION": {
        "APPLY": false,
        "PIXEL_COLOR": {
            "r": 0,
            "g": 0,
            "b": 0,
            "a": 200
        },
        "REGIONS": [{
                "type": "Rectangle",
                "metaData": {
                    "topLeftX": 45,
                    "topLeftY": 60,
```

```
                    "width": 900,
                    "height": 60
```

ORACLE INTERNAL

```
                "type": "Rectangle",
                "metaData": {
                    "topLeftX": 43,
                    "topLeftY": 132,
                    "width": 460,
                    "height": 655
                }
            }, {
                "type": "Ellipse",
                "metaData": {
                    "centerX": 643,
                    "centerY": 432,
                    "semiMajorAxis": 400,
                    "semiMinorAxis": 200
                }
            }, {
                "type": "Polygon",
                "metaData": {
                    "vertices": [{
                            "x": 100,
                            "y": 100
                        }, {
                            "x": 300,
                            "y": 400
                        }, {
                            "x": 500,
                            "y": 600
                        }]
                }
            }],
        "PAGES": []
    },
    "WAITING_FOR_SCREENSHOT": 5000,
    "SCREENSHOT_INTERVAL": 500,
    "MULTI_PART_SIZE": 7
}
```

## 1.3.18. How do change the color of error diffs in the image ?

**Answer**

Provide colors of your choice in this node of the customRangeTestSuite.json

"SCREENSHOT_ERROR_IMAGE_PIXEL_COLOR":{

ORACLE INTERNAL

- "b":0,
- "a":255

}

## 1.3.19. Setting the wait time

If I do not pass waitingTime in the API visualRegression.takeScreenshot(driver,done,'Open Custom Range Popup',waitingTime), what is the value the tool defaults to ?

**Answer**

If you do not pass waitingTime, the tool uses the value as defined in customRangeTestSuite.json

"WAITING_FOR_SCREENSHOT":5000

## 1.3.20. Regarding screenshot image capture

I see that the tool takes 5 screenshots of my page as denoted by this attribute "NO_OF_SCREENSHOT":5 in the customRangeTestSuite.json. Does it capture the image all at once?

**Answer**

Yes the no of screenshots are controlled by NO_OF_SCREENSHOT. It waits certain amount of time in between taking screenshots of your pages.

This is defined in "SCREENSHOT_INTERVAL":500 of the file customRangeTestSuite.json

## 1.3.21. I do not want certain area of my page to be used for comparison. How can I control that ?

**Answer**

Yes, you can control that. You can define various geometrical shapes in your page in the customRangeTestSuite.json and instruct the tool not to use it comparison.

You can add your shapes inside REGIONS node of SCREENSHOT_FORBIDDEN_REGION.

{

- "type":"Rectangle",
- "metaData":{
  - "topLeftX":45,
  - "topLeftY":60,
  - "width":900,
  - "height":60

}

}

Other geometrical shapes like Ellipse and Polygon are supported by the tool.

ORACLE INTERNAL

**Answer**

sh ../visual-regression/shell/runVisualRegression.sh

## 1.3.23. Is it already merged to develop ?

**Answer**

The necessary 3rd party licenses have already been procured. This is merged to develop.

## 1.3.24. HOW TO UPTAKE VISUAL-REGRESSION IN YOUR UIT PROJECT

**Answer**

1. Go through the above Q&A thoroughly.

2. Open GitBash in planning/SeleniumTest/tools/visual-regression

3.1 To integrate Visual Regression in UIT_Mocha Selenium Project
sh initial_setup.sh -u sbankara -p ../../UIT_Mocha -v ../visual-regression -s ../UIT_Mocha -t ../tools

sh initial_setup.sh -u rperugu -p ../../UIT_Mocha -v ../visual-regression -s ../UIT_Mocha -t ../tools

3.2 To integrate Visual Regression in JetDashboardsTest Selenium Project
sh initial_setup.sh -u manosahu -p ../../JetDashboardsTest -v ../visual-regression -s ../JetDashboardsTest -t ../tools

4. Modify package.json to include the following dependencies
"visual-regression":"file:../visual-regression"

5. npm install visual-regression

7. sh initSuite.sh suite1 to create a new TestSuite that tests member selector

10. Play and record using SeleniumIDE browser plugin and the save the file as suite1-mocha.js in suite1 directory.
Please note the suffix "-mocha.js". This is expolited by the CodeGeneration Framework to generate code.

11. If there are IFrame index issues, please change it in suite1-mocha.js and execute step#10 again.

11.1 sh genCode.sh suite1 This generates source code and save it in suite1 folder with the name suite1.js.

12. Modify runc.sh , runt.sh and runct.sh to include "suite1" in your testsuites repertoire.

The following shows the contents of runc.sh

```
------------------
#!/bin/sh
echo "Sourcing utils.sh"
source ./utils.sh
```

```
function executeVisionTestSuites()
```

ORACLE INTERNAL

```
   ##  At Step#7 , suite1 was created to test member selector.

   cookAndExecuteBaselineTestSuite baseline_vision   suite1
   cookAndExecuteBaselineTestSuite baseline_vision   suite2

}


function kickOff()
{
   copyScrpts
   executeVisionTestSuites
   removeScrpts
}
```

kickOff

---

Similarly make changes in runt.sh and runct.sh

13. The following files  are either created or modified in the above steps.These are present in your project directory.
    These  must be committed to GIT.
    package.json
    project.properties
    project-automation.properties
    copyScrpts.sh
    removeScripts.sh
    runc.sh
    runt.sh
    runct.sh
    automation/baseline_server.properties
    automation/test_server.properties

14. If you want to remove the visual-regression and automation files from your project,do the following.
    Open GitBash in your project.
    sh removeScripts.sh
15. You can always get it back with the following command
    sh copyScripts.sh

16. sh genReport.sh suite1  . This is the crux of this entire work. This compares the images in training and testing folder under suite1 and generates the report

## 1.4. Related articles

ORACLE INTERNAL

kb-how-to-article

## 2 Comments

Manoranjan Sahu

zoom recordings are  available at \\slc16cwm.us.oracle.com\visual_regression

Ratnalarao Perugu

if you are getting below error while running "npm install" please set up the updated registry and re run the npm install .

Error :  "npm ERR! code UNABLE_TO_VERIFY_LEAF_SIGNATURE"

~~npm config set registry https://artifacthub-tip.oraclecorp.com/api/npm/npmjs-remote~~

npm config set registry https://artifacthub-phx.oci.oraclecorp.com/api/npm/npmjs-remote