

CATS User Guide

Version 7.1



SapientRazorfish
8/20/2018

Table of Contents

1	Introduction	7
1.1	Scope and Purpose	7
2	Installation.....	7
2.1	System Pre-requisites	7
2.2	CATS Installation	7
2.2.1	Installation on Windows	7
2.2.2	Installation on Mac	8
2.2.3	Installation Ubuntu	9
2.3	Android Emulator Installation	10
2.4	IOS Simulator Installation	10
2.5	Safari 10 on Mac	11
3	CATS Overview.....	11
3.1	Introduction	11
3.2	Technology Stack	11
3.3	Features	12
3.4	Advantage	12
3.5	Package Content	13
3.6	Execution Flow at High-Level	14
4	Data Files.....	15
4.1	Introduction	15
4.2	ORM - Object Repository Management Structure	15
4.3	TCM - Test Case Management Structure	16
4.3.1	SUMMARY Sheet	17
4.3.2	SCENARIOS Sheet	18
4.3.3	CONFIG Sheet	18
4.3.4	TEST CASE Sheet	19
4.3.5	BDD - Behavioral Driven Development	21
4.4	TDM - Test Data Management Structure	21
4.5	Manual Mapping Sheet	22
4.6	TCM UI	24
4.6.1	Enable Macros for Windows	24
4.6.2	Enable Macros for Mac	24
4.6.3	New Sheet with Template	25
4.6.4	Test Case Builder	25

4.6.5 Scenario Builder	26
4.6.6 Admin	27
4.6.7 Migrating to New TCM	29
4.6.8 TCM Validation	30
4.6.9 Easy Fill	31
4.6.10 Meta Data	31
4.6.11 Trouble Shooting	32
5 CATS Script Creation	32
5.1Test Case	32
5.1.1 Test Case	33
5.1.2 Control Flow	33
5.1.3 Object Reference	37
5.1.4 Selecting Actions	37
5.1.5 Highlighting Errors	38
5.1.6 Step Description	38
5.1.7 TestData, Expected Result, Service Source	38
5.1.8 FLAGs	39
5.1.9 Ignore Flag	39
5.1.10 Exit Flag	40
5.1.11 Production Monitoring Flag	42
5.1.12 Single Suite on Multiple Browser:	42
5.2Test Data	43
5.2.1 Hardcoded Data	43
5.2.2 Variables	43
5.2.3 TDM Reference	44
5.2.4 Random Number	45
5.3Scenario	45
6 CATS BDD.....	47
6.1CATS BDD	47
6.1.1 Installation	47
6.1.2 Config Changes	47
6.1.3 Create Feature	47
6.1.4 Step Def Creation from TCM	47
6.1.5 Create Step Definition	47
6.1.6 Using CATS Actions	48
6.1.7 CATS Suites Setup	49
6.1.8 BDD Execution	50
6.1.9 BDD Auto Assertion	50

6.1.10 BDD Report	51
7 CATS Configurations.....	51
7.1CATSConfig	51
7.1.1 Highlighting Elements	51
7.1.2 Direct Scenario Execution / Label restricted Execution	51
7.2CATSToolsInput	52
7.2.1 Excel Data File	52
7.2.2 CSV Data Files	52
7.2.3 YAML or JSON Data Files	52
7.2.4 TDM EXPORT	57
7.2.5 Empty TDM Cell Reading	58
7.2.6 Creative Default Pixel Variation	58
7.3CATSReport	58
7.3.1 Executive Report	58
7.3.2 Executive Report	59
7.3.3 Report Header and Threshold	59
7.3.4 Manual Mapping Report	59
7.3.5 XML Report	59
7.3.6 Take Screenshots	60
7.4CATSServiceTestConfig	60
7.5CATS License	60
7.5.1 SR License	60
7.5.2 Trial License	60
7.5.3 Permanent License	60
7.5.4 Renewal of License	61
7.5.5 Server License	61
7.5.6 CI/CD Server License	61
7.6BrowserConfiguration	61
7.6.1 Browser Capabilities	61
7.6.2 Firefox User Agent	62
7.6.3 Browser Proxy	63
8 CATS Execution Modes.....	63
8.1Parallel Execution	64
8.2Command Line Execution	64
8.3Labeled Execution / Target Regression	66
8.4Batch Execution	67
8.5Batch Sequence Execution	68

8.6Production Monitoring Execution	70
8.7Sauce Labs Execution	71
8.8Browser Stack Execution	73
8.9Grid Execution	74
8.10....Cross browser testing	76
8.11....Jenkins Execution	76
8.11.1 Jenkins Job Setup	76
8.11.2 Jenkins Slave Setup	82
8.12....Bamboo Execution	84
8.13....Test Link Integration	87
8.14....Test Rail Integration	87
8.15....Kanoah Integration	88
8.16....Step by Step Execution	89
8.17....Multiple Instance Execution	89
9 Analytics Testing	89
10 Service testing.....	90
10.1....Introduction	90
10.2....IO	90
10.3....NFS	90
10.4....FTP	91
10.5....SFTP	91
10.6....XML	91
10.7....IBM MQ	91
10.8....JSON	92
10.9....REST	92
10.10.SOAP	92
11 Mobile Testing	92
11.1....Windows	92
11.1.1 Android Emulator (Genymotion) Setup:	92
11.1.2 Execution	93
11.1.3 Actual Android Device Setup :	93
11.2....Mac	96
11.2.1 Machine Setup	96
11.2.2 Execution	96
11.2.3 Execution on iOS Real Device on Safari Browser	96
11.3....SeeTest Mobile Lab Execution	98

11.3.1 Android Device	98
11.3.2 iOS Device	101
11.4....Chrome Emulator	105
11.4.1 Responsive Testing	105
11.4.2 Supported Devices	105
11.4.3 Configuration Setup	106
12 CATS UI.....	106
12.1....Toolbar	106
12.2....Merge Tool	107
12.2.1 Baseline Data File	107
12.2.2 Merge Data File	107
13 Custom Actions	108
14 Help	108
14.1....Getting help	108
14.2....Trouble shooting	109
14.3....Best Practices	109
14.4....List of “Don’t do”	109
14.5....FAQ	109

Revision History

Revision	Date	Author(s)	Description
0.1	16/07/2015	Roopesh Kumar Tayaloor	Initial Draft
1.0	01/06/2016	Rakesh Horkeri	Added 4.0 Features
1.1	10/06/2016	Shishir Verma	Added Info for BDD feature
1.2	17/08/2016	Rakesh Horkeri	Added Scenario Exit and Chrome Emulation on Grid features (CATS 4.2)
1.3	21/12/2016	Ismail Ali	Exit Scenarios, Scenario and Test Case Iteration, Reports
1.4	30/1/2017	Ismail Ali	FAQ Section, Iterate Variables, Dynamic OR Variables, Service Testing TDM
1.5	14/03/2017	Ismail Ali	7, 7.3, 8.2
1.6	04/04/2017	Ismail Ali	7.3, 8.10, 8.11
1.7	25/07/2017	Ismail Ali	
1.8	27/12/2017	Ismail Ali	4.3.4, 4.6, 5.1.2, 9

1 Introduction

1.1 Scope and Purpose

The CATS user guide is the complete reference for the CATS installation, process flow, creating the test case, usage guide lines and troubleshooting guidelines based on the FAQ.

Intended Audience of this document are anybody who is interested to use or implement the CATS automation framework

Expectation from Audience:

- a. User to be aware of testing domain
- b. User to be aware of basic of DOM and XPATH construction
- c. CATS License for test execution

2 Installation

2.1 System Pre-requisites

- a. Windows, Linux, Mac with minimum 4B RAM (8GB Recommended)
- b. [JDK 1.8](#) installed (**JAVA 10 is not supported**)
- c. For Mobile Testing
 - I. [Appium](#) – The test automation tool for mobile applications
 - II. [Node JS](#) – The supporting installation for appium
 - III. [Android SDK](#)– The Android SDK package to support android emulation
 - IV. [GenyMotion](#) - The simulator for Android UI
 - V. [Oracle Virtual Box](#) - The virtual box to manage the multiple images of simulators
 - VI. [Mobizen](#) - The real device screen casting to desktop also install the mobizen from google play store

2.2 CATS Installation

2.2.1 Installation on Windows

- a. Download CATS from <https://lion.box.com/v/cats-download> and Extract it to some location
- b. Verify the JDK installation and PATH correctness

C:\ Select Administrator: C:\Windows\system32\cmd.exe
C:\Users\iali>java -version
java version "1.8.0_92"
Java(TM) SE Runtime Environment (build 1.8.0_92-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.92-b14, mixed mode)
C:\Users\iali>path
PATH=D:\Backup\Install\apache-maven-3.3.3-bin\bin;D:\Backup\Install\Java\jdk1.8.0_92\bin;C:\Windows\0\;D:\Backup\Install\Ruby21\bin;D:\Backup\Install\ruby21\drivers;C:\Program Files (x86)\Common F;D:\Backup\Install\Android\android-sdk\platform-tools;C:\Program Files (x86)\Lenovo\Access Connection\;D:\Backup\Install\Android\ sdk1\platform-tools;C:\Program Files (x86)\HP\Unified Functional Test\;C:\Users\iali\AppData\Local\Box Edit\;C:\Users\iali\AppData\Roaming\npm
C:\Users\iali>

- c. JDK bin path to be added to [system environment variables](#)
- d. Remove path “*C:\ProgramData\Oracle\Java\javapath*” from the PATH variable if it exist

- e. Disable Automatic Java updates (to avoid path overwrite),
 - I. Search *Configure Java* in Start Menu
 - II. Go to *Update* tab and uncheck the *Check for updates automatically*
 - III. Click on *Do Not Check* button from the popup
 - IV. Click on *Apply* and *OK*
- f. Go to *CATSIcore* folder and run *LaunchCATSUI.bat*
- g. Go to license tab on CATS UI and select SR Login, use your lion credentials xxxx@publicisgroupe.net / password and Project Name and click on Generate License
- h. In case you want to generate the license for CI/CD system please create a License Request through CATS Customer Portal.

2.2.2 Installation on Mac

- a. Download CATS from <https://lion.box.com/v/cats-download> and Extract it to some location
- b. Set up Java Path on Mac :
 - I. Download the latest JDK and install
 - II. Open *.bash_profile* with any editor such as nano editor (command is : nano *.bash_profile*)
 - III. Set your JAVA_HOME and BIN path using the export command as depicted in the screen shot below.

Note : JAVA_HOME should be till ..Home and BIN path should be till ..bin

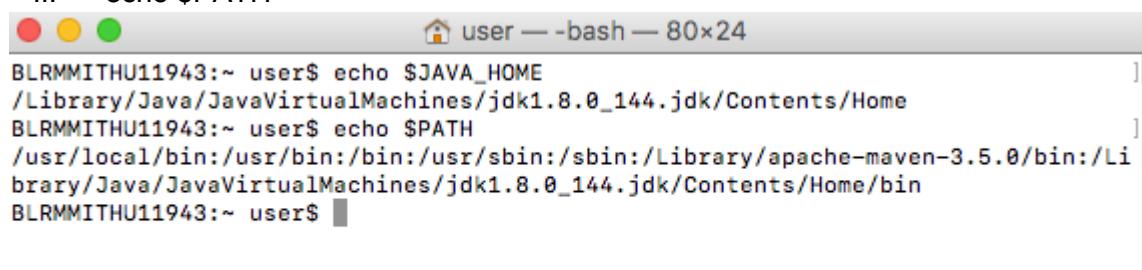


```
user — nano .bash_profile — 80x23
GNU nano 2.0.6          File: .bash_profile

export JAVA_HOME=/Library/Java/JavaVirtualMachines/jdk1.8.0_144.jdk/Contents/Home
export PATH=${PATH}:/Library/apache-maven-3.5.0/bin:$JAVA_HOME/bin

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Page  ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^V Next Page  ^U UnCut Text  ^T To Spell
```

- c. Verify the JDK installation and PATH by using following commands
 - I. echo \$JAVA_HOME for Java_Home
 - II. echo \$PATH



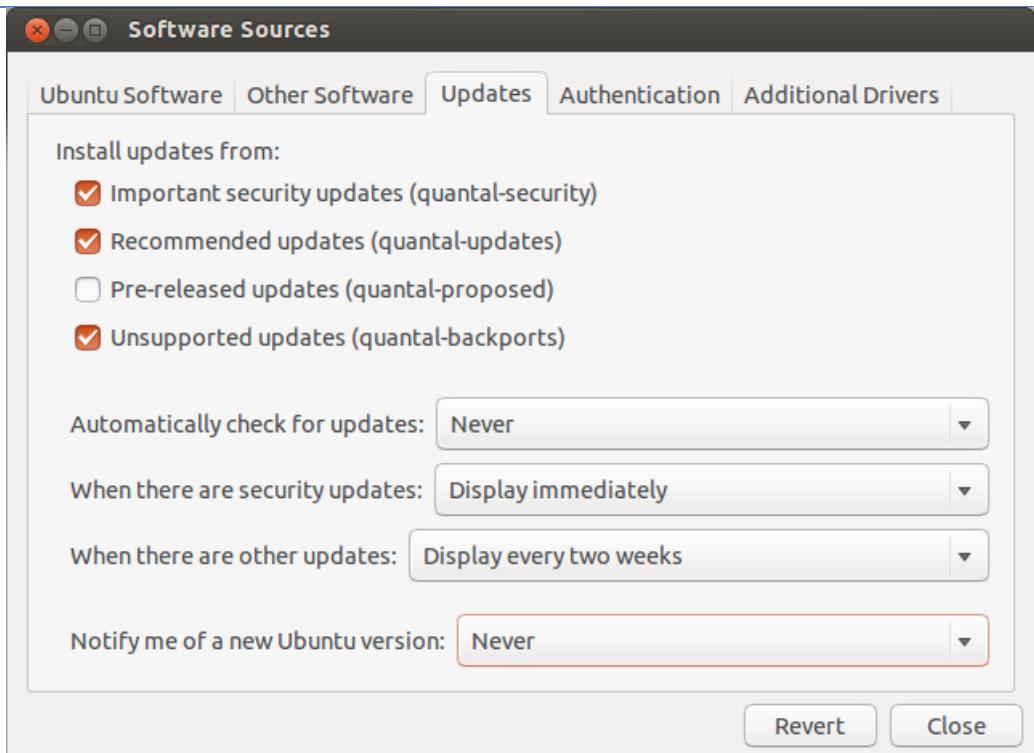
```
user — -bash — 80x24
BLRMMITHU11943:~ user$ echo $JAVA_HOME
/Library/Java/JavaVirtualMachines/jdk1.8.0_144.jdk/Contents/Home
BLRMMITHU11943:~ user$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/Library/apache-maven-3.5.0/bin:/Library/Java/JavaVirtualMachines/jdk1.8.0_144.jdk/Contents/Home/bin
BLRMMITHU11943:~ user$
```

-
- d. Disable Automatic Java updates (to avoid path overwrite),
 - I. Open System Preferences.
 - II. Click Java to open the Java settings panel
 - III. Click on the "Update" tab
 - IV. Untick the box "Check for updates automatically"
 - V. Click "Do not check" in the pop-up window.
 - VI. Click "Apply" and then "OK".
 - e.
 - f. Go to **CATSILicense** folder you will find **UserMacGen.gen** file which should be sent to catslicensingteam@sapient.com if you don't have valid license already
 - g. License team will revert with three files which needs to be placed at **CATSILicense**. Now CATS is ready to use with valid license
 - h. To run cats from the terminal Go to **CATS!core** folder and run **start_cats.sh**
 - IV. To launch the CATS UI go to **CATS!core** folder and run
LAUNCHCATSUI.sh(command : sh LaunchCATSUI.sh)

2.2.3 Installation Ubuntu

- V. Download CATS from <https://lion.box.com/v/cats-download> and Extract it to some location
- VI. Set up Java Path on Mac :
 - I. Download the tarball from Oracle website
 - II. Unzip it by sudo tar -xvpzf fileName -C /installation_folder_name
 - III. Change the files permission and ownership
 - IV. In terminal enter **sudo -H gedit /etc/profile**
 - V. Append following two entries to /etc/profile and save

```
JAVA_HOME=/usr/lib/jvm/<your jdk version>
PATH=$PATH:$HOME/bin:$JAVA_HOME/bin
export JAVA_HOME
export PATH
```
- VII. Verify the JDK installation and PATH by using following commands
- III. echo \$JAVA_HOME for Java_Home
- IV. echo \$PATH
- VIII. Disable Automatic Java updates (to avoid path overwrite)



- IX. Go to **CATSILicense** folder you will find **UserMacGen.gen** file which should be sent to catslicensingteam@sapient.com if you don't have valid license already
- X. License team will revert with three files which needs to be placed at **CATSILicense**. Now CATS is ready to use with valid license
- XI. To run cats from the terminal Go to **CATSICore** folder and run **start_cats.sh**
- XII. To launch the CATS UI go to **CATSICore** folder and run **LAUNCHCATSUI.sh** (*command : sh LaunchCATSUI.sh*)

2.3 Android Emulator Installation

- a. Install Appium, Node JS and Android SDK as mentioned in 2.1.c
- b. Open SDK manager and install the below packages
 - I. *Android SDK Tools*
 - II. *Android SDK Platform-tools*
 - III. *Android SDK Build-tools*
 - IV. *Google USB Driver*
- c. Add new environment variable **ANDROID_HOME** and value should be SDK location pointing to the folder to (*drive location\Android\android-sdk*)
- d. Append the environment PATH **%ANDROID_HOME%\platform-tools**

2.4 IOS Simulator Installation

Xcode:

- a. Download Xcode from <https://developer.apple.com/xcode/downloads/>
- b. Open Xcode after installation, go to Devices menu, check your required simulators are available. If not please download the necessary simulators by clicking on New Simulator >> Get simulator
- c. Next, install Node (npm will be installed with node): <https://nodejs.org/download/>
- d. Install Appium (<https://bitbucket.org/appium/appium.app/downloads/>)

Android SDK:

- e. Download the installable from here <https://developer.android.com/sdk/index.html#Other>

-
- f. After installing, in the Android SDK Manager, select the packages you want to use, and download them. Same as windows installation.
 - g. Install Appium (<https://bitbucket.org/appium/appium.app/downloads/>)

2.5 Safari 10 on Mac

Starting with Safari 10 on OS X El Capitan and macOS Sierra, Safari comes bundled with a new driver implementation that's maintained by the Web Developer Experience team at Apple. Safari's driver is launchable via the /usr/bin/safaridriver executable, and most client libraries provided by Selenium will automatically launch the driver this way without further configuration.

- Ensure that the Develop menu is available. It can be turned on by opening Safari preferences (Safari > Preferences in the menu bar), going to the Advanced tab, and ensuring that the Show Develop menu in menu bar checkbox is checked.

- Enable Remote Automation in the Develop menu. This is toggled via Develop > Allow Remote Automation in the menu bar.
- Authorize safaridriver to launch the webdriverd service which hosts the local web server. To permit this, run /usr/bin/safaridriver once manually and complete the authentication prompt
- Run below shell command to verify if webdriverd is loaded
 - `launchctl list | grep webdriverd`
- Incase webdriver is not loaded, please load using following shell command
 - `launchctl load /System/Library/LaunchAgents/com.apple.webdriver.plist`

For more information related to the latest WebDriver Support for Safari 10 you may refer to [WebDriver Support in Safari 10](#)

3 CATS Overview

3.1 Introduction

CATS (Comprehensive Automation Test Solution) is the internal sapient framework for test automation. The CATS automation framework serves as a single solution for various automation testing like functional, creative, mobile web, mobile app and services. It also lets the user perform execution on various platforms like windows, Linux, Mac, and Android. User can also run the same script on the various browsers, emulators, simulators, cloud base devices and browsers (Sauce Lab, Browser Stack and SeeTest)

3.2 Technology Stack

CATS is java based automation framework using open source libraries and services. Below is the list of technology stack used

- a. Java jdk 1.8 / jre1.8 - for core framework development
- b. Jsoup HTML Parser - for object properties reading for OR Tool
- c. Selenium Webdriver - for web based application automation testing
- d. Appium - for mobile app testing and mobile web execution on actual devices
- e. TestNG - for execution flow control and multithread execution
- f. Jasper - for runtime report and execution report
- g. Jenkins - for CI build scheduling
- h. Log4j - for log creation during execution

-
- i. GenyMotion - for android device emulation
 - j. Jersey - for API and services testing

3.3 Features

- a. Keywords enabled excel driven framework with more than 250 frequently used actions/keywords
- b. Data driven capability to run the same test or scenario for multiple set of data combination
- c. Functional and Creative testing using Selenium web driver
- d. Cross browser testing on different browsers (IE, Firefox, Chrome Safari) and OS (Windows, Mac, Linux)
- e. Services automation testing
 - I. *JSON rest services*
 - II. *XML rest services*
 - III. *ESB services*
 - IV. *IBM MQ*
 - V. *File IO*
- f. Execution support on cloud
 - I. *Browser Stack*
 - II. *Sauce Lab*
 - III. *See Test*
- g. Mobile testing on iOS and Android. Both on devices and emulators/simulators
- h. Headless browser execution using Panthom JS
- i. Macro enabled TCM to minimize the user errors
- j. Testing across multiple systems in parallel using selenium Grid
- k. Jenkins integration for schedule run
- l. Conditional steps execution using IF ELSE
- m. BDD (Gherkin) way of designing the test cases
- n. Generic e-commerce sanity test suite
- o. Production monitoring based execution report
- p. Custom actions template to extend the CATS capability and features
- q. Screen captures on every failure
- r. Execution report in HTML, PDF and Excel format
- s. Runtime dynamic object property change support
- t. Local, Global, Suite Variables support
- u. Control parallel and sequential execution of the suites

3.4 Advantage

CATS framework is created based on the industry experience on the automation need and common challenges of automation implementation, skill set required etc.,

- a. Using CATS framework even manual tester can start creating script in 2days of training
- b. CATS is excel based framework and it is managed through three different workbooks that are internally referenced but externally independent
 - I. *TCM - Test Case Management*
 - II. *TDM - Test Data Management*
 - III. *ORM - Object Repository Management*
- c. Single framework for desktop, mobile and services testing
- d. Macro enabled TCM helps to build the test cases more faster and error free
- e. Re-usability of the test case in different scenarios
- f. Same functional script can be used for the multi-lingual site provided the functionality remains same

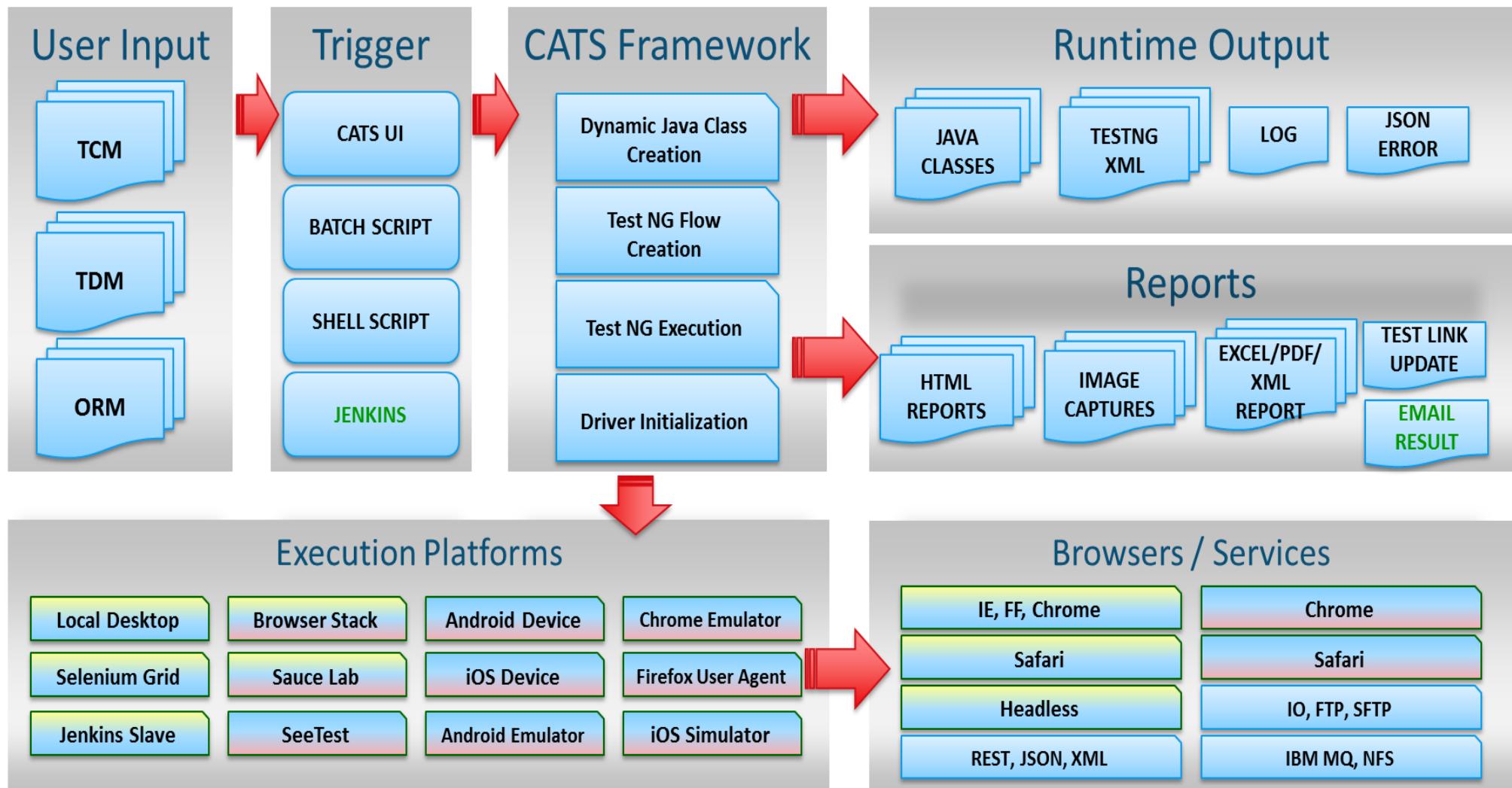
-
- g. Dedicated support team to address the quires and challenges
 - h. Scheduled trainings and Just-In-Time trainings

3.5 Package Content

The CATS framework package folder contains many folders and sub-folders. Below are the list of folder you would need to know

Folder Name	Description
Core	This directory contains the core of the CATS tool engine
DataFiles	The default location where TCM, TDM and ORM are kept
Drivers	IEDriver, PanthomJS, Chrome Driver, Geckodriver for Safari and Edge driver
Input	This folder contains error output files generated during data files processing
License	This folder contains License Files.
Output	Execution reports
core\logs	This folder contains log files.
Selenium Node	Selenium node to run the test cases in grid fashion
Selenium Server	Selenium Server to run the test cases in grid fashion

3.6 Execution Flow at High-Level



4 Data Files

4.1 Introduction

CATS framework consumes the user input through three data files (workbooks) namely TCM, TDM and ORM. Current CATS capability restricts the user to use only one set of TCM, TDM and ORM for a particular execution. However there is no restriction to add the number of sheets to each workbook.

Although we have three independent workbooks, they are internally referenced during the data processing (Excel to JSON) as well as in TCM UI and Macros.

4.2 ORM – Object Repository Management Structure

The OR maintains the object properties of each element being used in TCM. Object properties are mandatory to perform any action on the specific element. There are many properties used to locate the elements in CATS like HTML ID, X-Path, CSS Selector, Name, Class Name, Link Text, Tag Name, Partial Link Text, etc.,

Every element mentioned in the ORM is referenced in TCM using its Sheet Name > Page Name > Element Name. Highlighted in red are optional it has not impact on reference and execution.

Multiple sheets can be created to maintain the object properties logically for example each sheet could be referring to the application page objects or objects involved in the feature. It is suggested to create sheets based on the module for easy reference.

Multiple pages can be added in the each sheet it is to simplify the objects involved in the specific section or feature.

Column Name	Description
Sheet Name	It is used in TCM OR column
PAGE	Page name to logically group the elements within the sheet
ELEMENTNAME	Name of the element reference which will be used in TCM as reference
DYNAMIC PROPERTY	xpath, htmlid, classname or name with variable or TDM reference for dynamically changing runtime properties
HTMLID	HTML Id of the element
XPATH	X-Path of the element
CSS SELECTOR	CSS Selector path of the element
Name	Name attribute of the element
CLASSNAME	CSS Class name of the object
LINKTEXT	Link text of the element
TAGNAME	Tag Name of the element
PARTIAL LINKTEXT	Partial link text of the element.

Example of Dynamic OR:

<i>Dynamic property using OR Variable</i>
<code>xpath://div[@class='prd']//span[@text()=//ItemName]]</code>
<i>Dynamic Property using both OR variable and global variable</i>
<code>xpath://div[@class='prd']//span[@text()='{\$ItemName}'] ../img[@title='{\$color}']</code>
<i>Dynamic property using global variable</i>
<code>htmlid:{{\$checkoutbuttonid}}</code>
<i>Dynamic property using TDM reference</i>
<code>xpath://input[@name='{\$Home.Product.Field1}']</code>
<i>Dynamic property for more than one locator</i>
<code>xpath://a[text()='{\$orderNumber}']; htmlid:ordernum_{orderNumber}_lnk; classname:{\$order_confirmation}</code>
<i>Dynamic property using dynamic TDM reference</i>
<code>xpath://input[@name='{\$Checkout.Product.<<Language>>}']</code>

Refer CATS action reference for more details

4.3 TCM – Test Case Management Structure

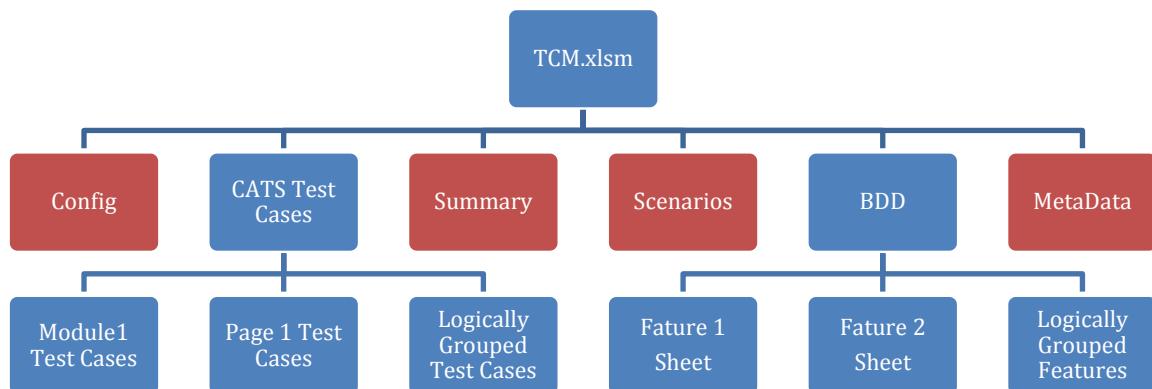
Test Case Management (TCM) is a crucial and important task in the test automation. While TCM being at center, all other tools and processes will be around this.

For CATS TCM is managed in workbook and it is macro enabled to make the test case creation easier. TCM consist of many sheets of which some of them are mandatory and should not be renamed, below are the various sheets present in TCM workbook. The names highlighted in red are mandatory sheets and names cannot be changed.

- a. SUMMARY
- b. CONFIG
- c. METADATA - Hidden sheet used for macros and dropdowns
- d. SCENARIOS
- e. Test Case Sheet - can be n-numbers
- f. BDD Sheet - can be n-numbers

Sheet names **must not** contain underscore or comma

Macros are enabled to make the OR reference, Action List, Mandatory field validations and TCM UI (for test case, scenario and suite builder) however the test steps are to be added manually.



4.3.1 SUMMARY Sheet

The summary sheet is used to define the test suites, the actual execution flow sequence will begin from here.

User needs to provide the suite level configuration details in this sheet.

Column Name	Description
RELEASE	This hold the release# or build # which is used to print in the execution report.
TEST SUITE	Name of the test suite which is mandatory and it should not have any special characters.
TEST CASES - SCENARIOS	This column contains reference to the SCENARIO ID or Test Case ID which needs to be executed. Multiple scenario or test case will be separated by comma. Example: SCENARIO_CHECKOUT, Sheet Name followed by underscore followed by ID
RUN	Whether suite is to be executed or not. Valid value could be "Y" or "N"
CONFIG	Configuration ID must be provided which holds the channel and browser information
BATCH	Suites with the same batch name will be grouped for execution. Batch name suffixed with _SEQ will be executed in sequential mode this is mainly used when you want to execute suite one by one due to inter dependency.
OWNER	Optional. It is for user reference
SUITE VARIABLES	User will have access to some of runtime variables like CURR_SUITE, CURR_TCNAME, CURR_SCENARIO, CURR_BROWSER, CURR_ENV (if passed through cmd arg) along with the user can also define additional variables prior to execution start. KEY1=VALUE1;KEY2=VALUE2
DESCRIPTION	Optional. It is for user reference

LABELS	User can define the labels and labels can be used during to filter which test suite to be executed through command prompt. While labels are used for execution RUN column is ignored during execution
--------	---

4.3.2 SCENARIOS Sheet

The scenario sheet is used to form the scenarios from the re-usable test cases. You can write the re-usable test cases which can be combined to form a complete scenario.

Column Name	Description
ID	Unique ID to identify the scenario in suite
SCENARIO NAME	Brief name of the scenario which will be reflecting in the report
SCENARIO DESCRIPTION	Optional. It is for user reference
TEST CASES	List of test cases separated by comma. Test data of the test case can be overridden and the same test case can be iterated for multiple test data. Example: Sheet Name followed by underscore followed by ID. Example SheetName_ID=Field5 or SheetName_ID=Field4;5; 10 or SheetName_ID=5 TO 15; 20;25; 30 to 35; 50 TO 100 BY 10
TEST DATA KEY	Test data of all the test cases used within the scenario will be overridden with the Test Data Key provided here. If more than one Test Data Key separated by semicolon will iterate the same scenario for number of data provided here. The test data key range can be given like 1 TO 15 and within range you can define the interval as well like 20 TO 30 BY 5 this will result in (20, 25 and 30)
SCENARIO VARIABLE	Scenario variables work like suite variables, only difference is the scope of this variable would be limited to the scenario and if the same variable name is used Suite variable as well then the Scenario Variable takes highest precedence.
LABELS	User can choose to run the scenarios directly without creating the Suites. Labels must be passed through command line argument or SCENARIO_LABELS in CATS Config. Scenario matching those labels will be used for the execution.

4.3.3 CONFIG Sheet

Config sheet holds the information related to the mobile, remote browser additional capabilities etc.,

Column Name	Description
ID	Unique ID to identify the driver configuration
CHANNEL	Channel where the execution needs to be performed
BROWSER	Browser on which the execution needs to be performed
VIEWPORT	View port to be provided when user wants to perform Creative testing at specified window size for responsive testing

DEFAULT	When user performs command line execution on scenario labels, the default browsers marked as Yes will be used for execution
ANALYTICS	Set this flag as Yes only when you want to perform Analytics testing as this will enable the proxy to capture all the request and responses which will slow down the execution.
NAME	Device Name configured in AVD, GenyMotion or Valid Chrome Emulator Name
VERSION	As configured in AVD/GenyMotion
DRIVER	Valid values are iOSDevice, iOSSimulator, AndroidSimulator, AndroidDevice, SauceLabsRemoteWebDriver, BrowserStackRemoteWebDriver, ChromeMobileEmulator
PLATFORM	Android or iOS
UDID	IP of the IOS Device only, leave it blank for other devices
BUNDLEID	ID of the app
APPPATH	Absolute path of the application (in the case of browser, it is the chrome path or Safari browser path)
PROTOCOL	
SERVER	
PORT	
CONTEXT	
PKGNAME	Applicable for only Android application, it is the package name
ACTIVITY	Applicable for only Android application, it is the activity name
DESCRIPTION	Optional. It is for user reference
REMOTEWEBDRIVER_URL	
CAPABILITIES	
APPUIMPATH	

4.3.4 TEST CASE Sheet

TCM can have as many as test cases sheets one needs, and every single test case sheet can hold as many as test cases user wants.

Column Name	Description
TC ID	Unique test case identifier of the test case within the sheet
TEST CASE NAME	Unique test case name which is used in reporting as well as in java class name
CONTROL FLOW	This column is used to write the control block or conditional steps. Valid values are IF, ELSE IF, ELSE, END IF, IF NOT, ELSE IF NOT. IF control must have END IF
OR	Sheet name from the ORM workbook from which page reference will be used
PAGE	Page name from the ORM sheet from which the element reference will be used

ELEMENT REF	The element name of the object to be used in the action
ACTION	Predefined action keyword to be selected or entered. This action will be performed on the element reference provided. The action keyword could be part of custom action as well
TEST DATA	Test data to be passed for this action. The test data could be hard coded value, TDM reference (\$SheetName.KEY.FieldName), Local Variable ({{VariableName}}), Global Variable ({{{VariableName}}})), Suite Variable (<<CURR_SuiteName>>)
EXPECTED RESULT	Expected result that needs to be compared against actual result during verification test steps. Expected results could be hard coded value, TDM reference (\$SheetName.KEY.FieldName), Local Variable ({{VariableName}}), Global Variable ({{{VariableName}}})),
SERVICE CONFIG	Key column reference from TDM Service Config sheet. It is used only for services
SERVICE SOURCE	This column is used to pass the input for services testing
DESCRIPTION	Short description of the step to be written in report
PRIORITY	Optional. For future reference
FLAGS	Based on the value selected, the action will vary. Valid values are IGNORE TESTCASE, IGNORE REPORT, IGNORE TESTSTEP, EXIT TC, EXIT SCN, EXIT SUITE, REPORT
REMARKS	Optional. It is for user reference

4.3.4.1 Direct Element Reference in TCM

It is a feature which ease the user to provide the Element Ref directly in the TCM sheet rather than defining the variable in OR and use it from TCM .For Example:

In TCM sheet pass the DirectRef in OR column and then express either xpath, htmlid, classname, name followed by double colon(::) in ElementRef as shown below:

	DirectRef	//*[@id='searchTerm']	ENTER	Hoods	
	Custom		SETORVARIABLE	{{\$searchTerm}}	searchTerm
	DirectRef	xpath://*[@id='{{searchTerm}}']	ENTER	Hoods	
	DirectRef	HTMLID::searchTerm	ENTER	Hoods	
	DirectRef	searchTerm	ENTER	Hoods	
	DirectRef	CLASSNAME::middle_blueSmall_input	CLICK		
	DirectRef	className::btn_searchForm	FLUENTWAIT		20
	DirectRef	className::btn_-	CLICK		

Also if string is passed in Element Ref which doesn't contains the four types said above then it is considered as xpath, htmlid or classname according to the basic format of the respective types.

Also dynamic variable can also be passed like localvariable({{variableintwobraces}}) and global variable({{{variableinthreebraces}}}). The variables can be set using setorvariable.

In the reports if the variable is not given any element type .It will be automatically updated that what type of element type is considered for calculation in reports as given below.

	: Enter <Hoods> into <./*[@id='searchTerm']>	ENTER	Hoods	
PASSED	DirectRef.xpath::./*[@id='searchTerm']	<0.1 sec		
is s PASSED	Custom	<0.1 sec		

The functionality is same as the Dynamic Property with a change that it excepts the element type even if it doesn't contains any dynamic variable, which is not in the case of Dynamic Property as it processes the element type only if it contains some dynamic variable.

4.3.5 BDD – Behavioral Driven Development

CATS provides support for BDD by specifying the scenarios in terms of desired behavior of the feature, component or unit. We have researched and analyzed on other frameworks which run tests written in BDD style - such as Cucumber, to ensure that transition to CATS from such frameworks is effortless.

This feature only allows the business people to specify the behavior and not be concerned about the test-cases. And so, it does not generate the test-case boiler plate code like Cucumber framework. The test-cases are assumed to be mapped to their corresponding behavior by testers. While this approach achieves the same outcome as any other framework, the style of execution is not alike.

Column Name	Category	Description
ID	Feature	Unique feature identifier to refer in the test suites
FEATURE NAME	Feature	Name of the feature from development story
FEATURE DESCRIPTION	Feature	Description of feature and used only for reference and readability purposes
SCENARIO NAME	Scenario	A feature can have multiple scenarios and every scenario is identified by its name. Every non-blank value in this column would start a new scenario within the feature
CLAUSE	Step Definition	Defines gherkin type steps like GIVEN, THEN or WHEN. Scenario can have multiple step definitions.
DESCRIPTION	Step Definition	The description of the step-definition in business language
TEST CASE	Step Definition	Step definition must be mapped to CATS Test Case ({TC Sheet Name}_{ID})
TEST DATA KEY	Step Definition	Test data of all the test case will be overridden by the field reference given here.

4.4 TDM – Test Data Management Structure

Test data comes into picture when you are dealing with the application where you don't have the same set of data available all the time, the same test case needs to be executed for the multiple

iterations having different test data for test case iteration or the same scenario needs to be executed for different test data combinations etc.,

Data parameterization is one of the recommended practice to avoid the lot of script maintenance, hence CATS allows user to pass the test data to the script in multiple ways.

- a. Hardcoded Test Data
- b. Test Data from TDM
- c. Test data stored in local or global variable

CATS allows user to maintain the test data in workbook. The values in the first column is referred as KEY and the values in the first row referred as FIELDS. The data or value found at the intersection of KEY and FIELD will be pulled during execution if the TDM reference is given in the Test Data or Expected Result column in the TCM.

The valid TDM reference in TCM begins with dollar followed by sheet name followed by dot followed by Key followed by dot followed by Field. Example: \$HomePage.UserName.Field1

Column Name	Description
KEY	The unique identifier maintained in the first column in every sheet. The same key name can be used in different sheet
FIELD	The unique identifier maintained in the first row in the every sheet. In case the user wants iterate the test case or scenario for different test data then the field name should end with "FIELD" followed by a NUMBER. Eg., FIELD1, FIELD2, CARD_FIELD1, etc., If the data used here is not going to be iterated then it can be name anything, for example: US_Label, FR_Label, etc.,
ER	Same as FIELD. The unique identifier maintained in the first row in the every sheet. In case the user wants iterate the test case or scenario for different test data with different expected result then user should mention the ER followed by a NUMBER. Eg., ER1, ER2, CARD_ER1, etc. During the iteration the ER suffix number will be replaced with the Field suffix number. When the data referred is Field3 then the ER3 column will be referred for expected result

Here is the example of providing the test data

KEY	Field1	Field2	ER1	ER2	QA	UAT
Login	Roopesh T	Roopesh			Testqa	testuat
Password	Wrong Pwd	Right Pwd	Wrong password	Right password	\$apientQ A	Cl!3ntUat

4.5 Manual Mapping Sheet

Manual mapping sheet is to be updated by the tester, where CATS test cases, scenarios or suites must be mapped to manual test cases. Based on the automation status the manual mapping status

will be updated in this spread sheet and overall coverage will be published to the Executive Summary.

Manual mapping sheet is also repurposed to update the execution status in Test Management system like Test Link or Test Rail.

User should set *DISPLAY_MANUAL_MAPPING_REPORT=Yes* in *CATSReport.properties*. Refer the screenshot for further clarification:-

1	CATS_SUITE_NAME	CATS_SCENARIO_ID	CATS_TC_ID	MANUAL_ID	CURRENT_RUN_STATUS	CONSOLIDATED_RUN_STATUS	EXECUTION_DATE
2	MySuite				PASSED		17/07/2017 12:35:31
3	MySuite	SCENARIOS_6			PASSED		17/07/2017 12:35:31
4	MySuite	SCENARIOS_6	Google_1		PASSED		17/07/2017 12:35:31
-							

Same test case executed against different Config (for different browser) will be tracked in separate columns, see the screenshot below where same suite is executed against more than browser.

CONFIG_AndSimulator	CONFIG_FF1	CONFIG_IE	CONFIG_Chrome
	PASSED	PASSED	
	PASSED	PASSED	
	PASSED	PASSED	

If the execution status is *FAILED* or *SKIPPED* corresponding to at least 1 config in a row , the *CONSOLIDATED_RUN_STATUS* will be *FAILED* or *SKIPPED* respectively , else it will be *PASSED*.

It has mainly 5 columns-

CATS_SUITE_NAME = #SuiteName as mentioned in TEST SUITE column of Summary sheet in TCM

CATS_SCENARIO_ID=#Scenario name as mentioned in TEST CASES - SCENARIOS column of Summary sheet in TCM

CATS_TC_ID=#TestCase name as mentioned in TEST CASES column of SCENARIOS Sheet in TCM

CONSOLIDATED_RUN_STATUS=#This will display the consolidated status corresponding to all the configs

EXECUTION_DATE=#This will display the date and time of status update

Manual Mapping sheet path can be configured by changing

MANUAL_MAPPING_SHEET_PATH in *CATSReport.properties*.

4.6 TCM UI

The TCM UI is a macro based VBA form, which is created for those who feels more comfortable with UI based application than the excel sheet. CATS5.0 is released with new TCM version 2016.021. The TCM UI has six different tabs and the tab names are listed below.

- a. TC Builder
- b. Scenario Builder
- c. Migration
- d. Validation
- e. Admin

Since the TCM UI is created using excel VBA, it will work only in windows and you will not be able to use it in any other OS that doesn't support VBA. The excel application should be macro enabled and every time the

4.6.1 Enable Macros for Windows

- I. Click the File tab
- II. Click Options
- III. Click Trust Center, and then click Trust Center Settings
- IV. In the Trust Center, click Macro Settings
- V. Make the selection "Enable all macros"
- VI. Select the checkbox "Trust access to the VBA project object model"
- VII. Click OK.

The TCM UI can be access is multiple ways

- I. When workbook is opened first time it will popup (if macro already enabled)
- II. Click ADD-INS tab, CATS then TCM UI
- III. Shortcut *Ctrl+Shift+C*

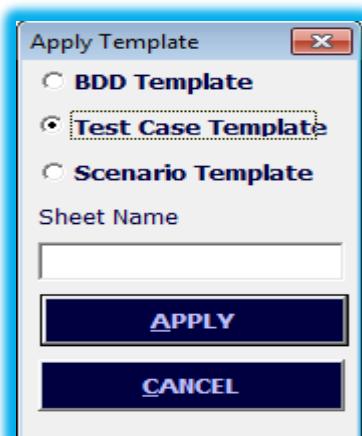
4.6.2 Enable Macros for Mac

- I. Open Excel Menu
- II. Click on Preferences
- III. Go to Authoring section and click on View
- IV. Go to In Ribbon, Show section and select the checkbox against Developer Tab
- V. Now from Preference dialog select Security & Privacy icon under Sharing and Privacy
- VI. Under Macro Security section select Enable All Macros

The TCM UI can be access is multiple ways

- I. When workbook is opened first time it will popup (if macro already enabled)
- II. Click Developer Tab tab, Click on Macros then select OpenCATSUI and click on Run

4.6.3 New Sheet with Template



As the sheet is macro enabled and the TCM contains various sheet types (Summary, Config, Scenario, Test Case and BDD) it is imported to associate the template to each sheet available in the TCM.

However user is expected to add two type of sheets frequently Test Case Template and BDD template. Unless the templates are associated the macro will not be enabled on that particular sheet or the sheet will behave / validate the content based on the template associated to it.

While adding the new sheet to the workbook you will be prompted to select the template and name of the sheet you want to name it to.

4.6.4 Test Case Builder

User can add the test case and the steps using the TC Builder tab. prior to using this sheet user should ensure that.

- a. At least one sheet with test case template associated exist
- b. OR update should be done from Admin Tab
- c. TDM reference should be loaded from Admin Tab
- d. Variables should be loaded from Admin Tab and this needs to be frequently updated to reflect the newly added variables

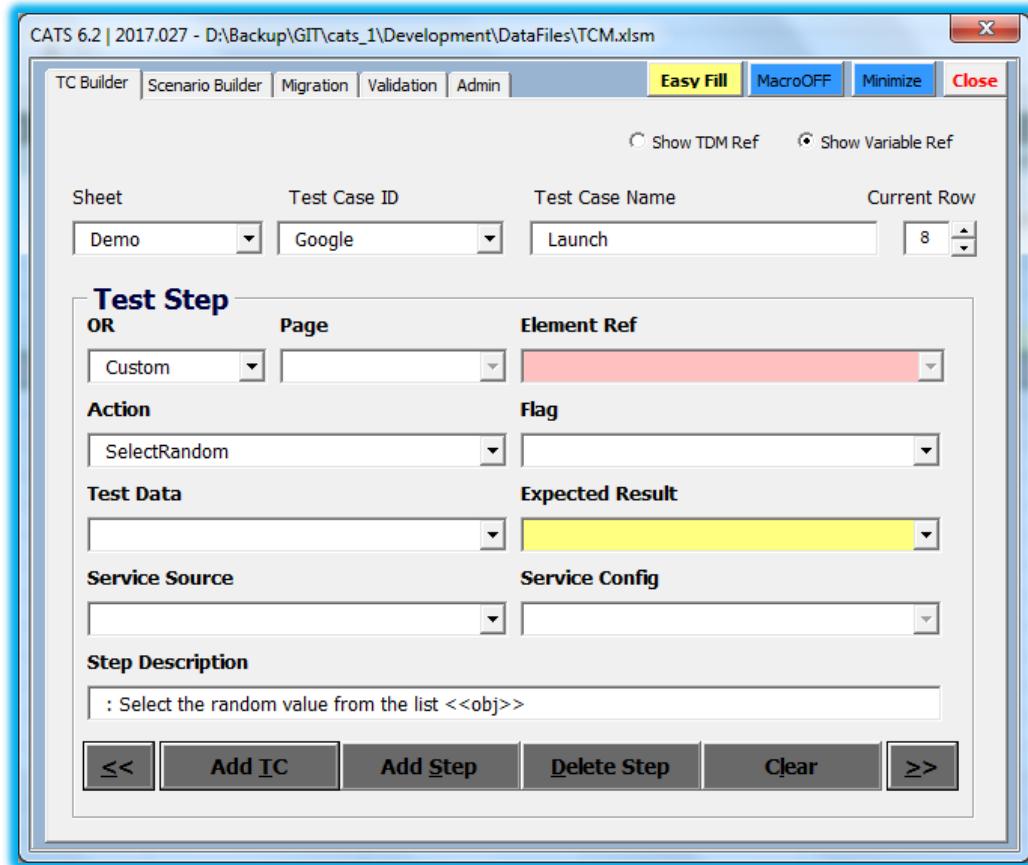
Select the Sheet where you want the add the test case, Enter the test case ID, Test Case name, Current Row (number where the test current test case / step to be added).

When you load the TDM, the Test Data and Expected Result section will allow you to choose the TDM reference from the list, however the field ***suffix must be added manually.***

If you want to refer the previously created local or global variable then click on the button "TDM Ref" the test data and expected result list will be loaded with the variable names.

After completion of test case and test step addition you may need to perform TCM validation to ensure the correctness of the mandatory fields

Field highlighted in pink is mandatory field for the selected action and highlighted in yellow is an optional field



Now, test data and expected result columns in TCM sheet will also display the drop down list provided user selects the checkbox “Enable TDM/Variable List in TCM”. To toggle between TDM reference and Variable reference user needs to change the radio button selection. Refer the above screenshot.

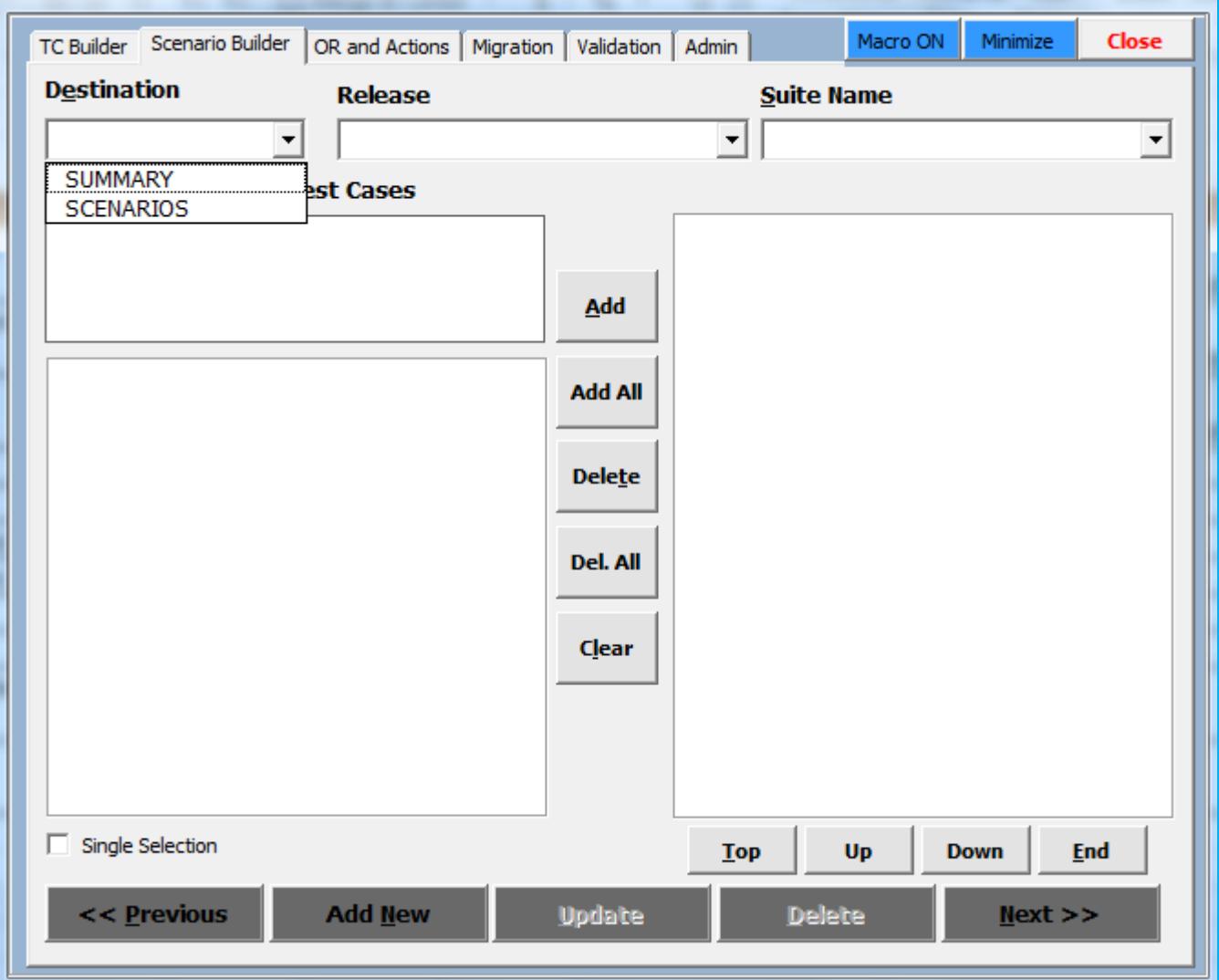
For TDM reference user must enter the numeric prefix manually if present in TDM. Sometimes the list in the TCM column might not get updated in such case you need to go to different sheet and then come back to the same sheet.

4.6.5 Scenario Builder

The scenario builder sheet lets the user add the test scenarios and the test suites. Based on the Destination sheet selected the field names and behavior will change.

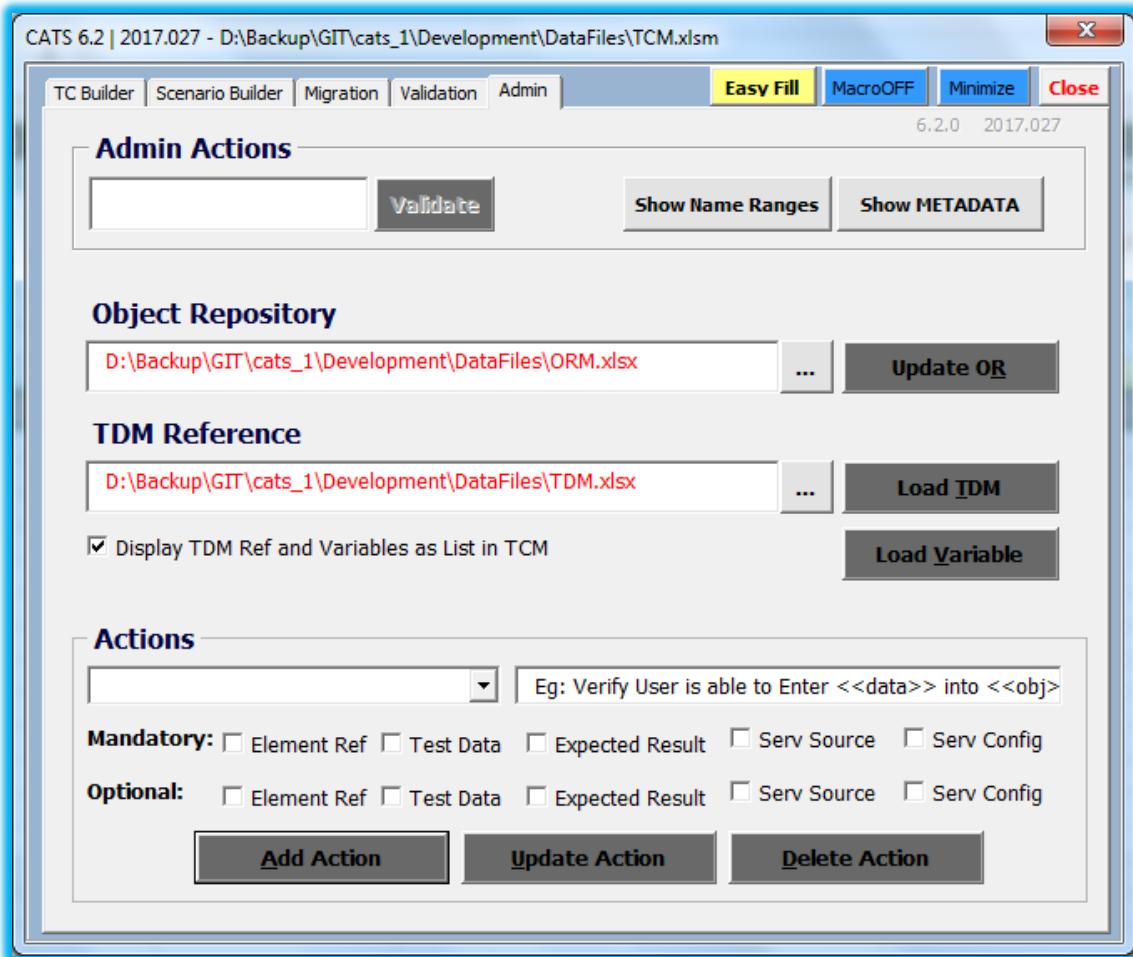
User can select the source sheets to load the test cases and scenarios in the below list which can be used to construct the scenario or suite. Either use the buttons or double clicks to add and remove from one section to other.

To add the multiple test cases to form the scenario, the user can do multiple selection in the same order as they want to create the flow and the selection order will be retained while adding to scenario or suite.



4.6.6 Admin

Object Repository: Object Repository update is required to populate the object reference in the test case sheet. If the new object reference is added to the ORM then you may have to update the OR to reflect the newly added object reference. Post OR update user will get the confirmation message OR Refresh is complete.



Load TDM: This is used to load the Test Data References (\$Sheet.Key.Field) used in the TDM reference selected and the test data keys will be populated in the TCM columns Test Data, Expected Result, Service Config and Service Source, if more than one test data follows the same naming pattern but there is a numeric suffix then suffix has to be added by the user.

Load Variable: This is used to scan through the variables used in the TCM and can be used to populate them into TCM columns Test Data, Expected Result, and Service Source.

Display TDM Ref: This check box must be selected in order the display the list in the TCM columns Test Data, Expected Result, and Service Source. Show TDM vs Variable in TC Builder tab radio button controls what to list in those columns.

Actions: Actions section is used to add or remove the custom actions and to update the description of the core actions. Once the custom action is created user can add the custom action name so that the user can start seeing them in the Action column in the test case template. Enabling the checkbox will let the user know while using this action about the mandatory fields.

Only admin user can enter the core actions and anybody would be able to add the core actions. Description allows keywords <<obj>> and <<data>> which will be replaced with object reference and test data used respectively

4.6.7 Migrating to New TCM

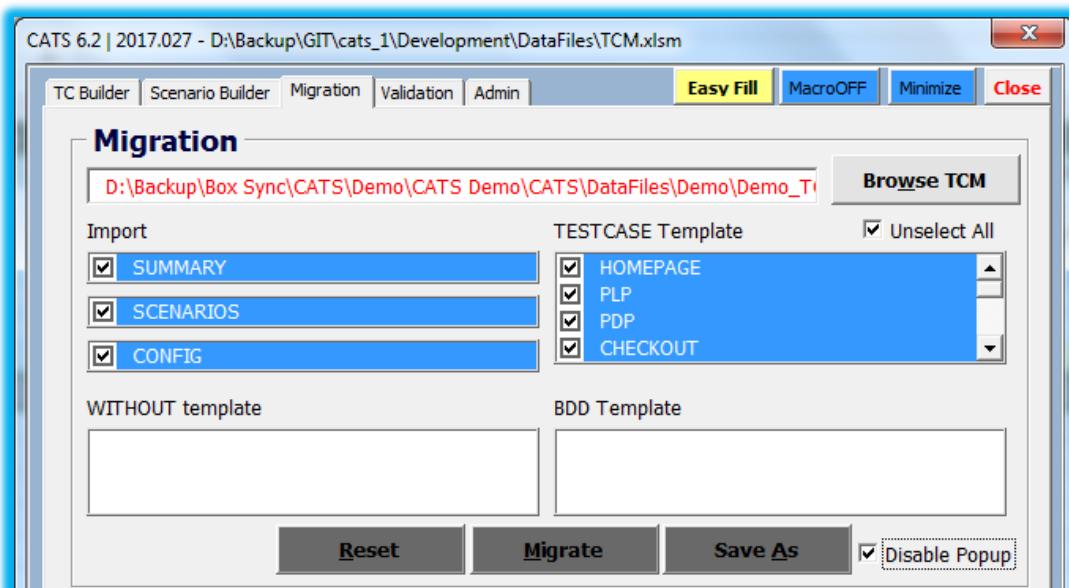
Migration: Every time new TCM version is released the user should migrate their old TCM to new version to get the benefit of new features and the format. This can be done easily by using the migration tab.

User can import only one sheet for SUMMARY, CONFIG and SCENARIOS. However user can import any number of sheets associated to Test Case Template, BDD template and without template.

If previous TCM has the templates associated then the sheets will be displayed in the respective template list box else all the sheets in the workbook will appear in WITHOUT template and TESTCASE template.

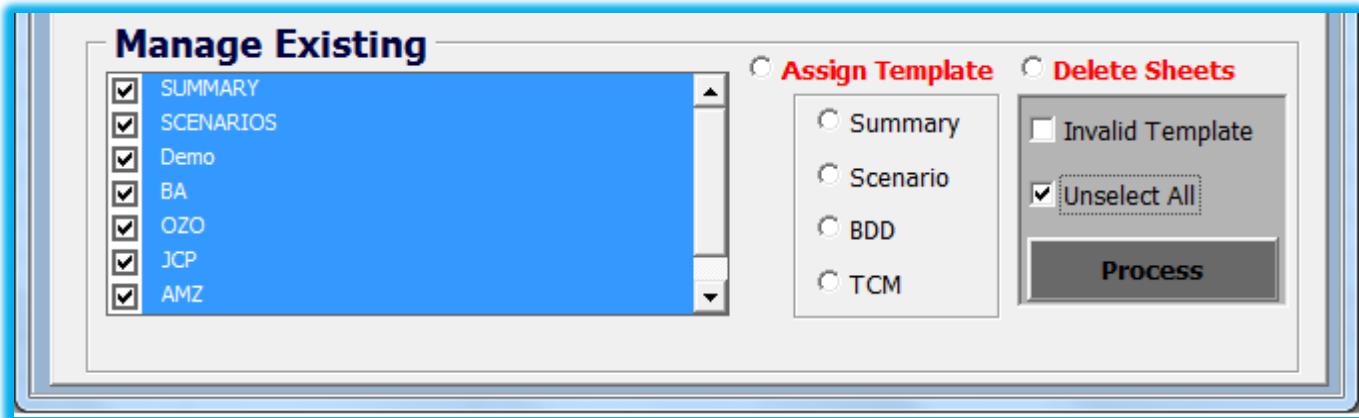
It is advised to update the OR prior to Migration process as the validation takes place during import process will result in too many errors.

Disable Popup checkbox will prevent the migration confirmation of individual test case template.



You have to wait till you get the final message "Migration Complete", however you will be notified upon migration completion of individual sheet.

Note: New and Old TCM file names must be different, as excel doesn't support to open the files with the same names.



Assign Template: There might be case where you would have moved the sheet from TCM without following the migration process or you might have created sheet without any template. As mentioned earlier if template is not associated then you might not get the benefit of macros in that sheet, hence you can associate the template for such sheets using Manage Existing section.

You can use the filter checkbox to filter sheets without templates.

Delete Sheets: TCM can have as many sheets as excel allows, however when it comes to removing the unused or redundant sheet, it is always painful. Using manage existing section you can select the sheets you want to delete and click on delete sheets. *This process is cannot be undone.*

Note: TDM and OR doesn't require migration, it can be used as it is. However for OR you can refer if new columns are added to support additional features refer ORM - Object Repository Management Structure.

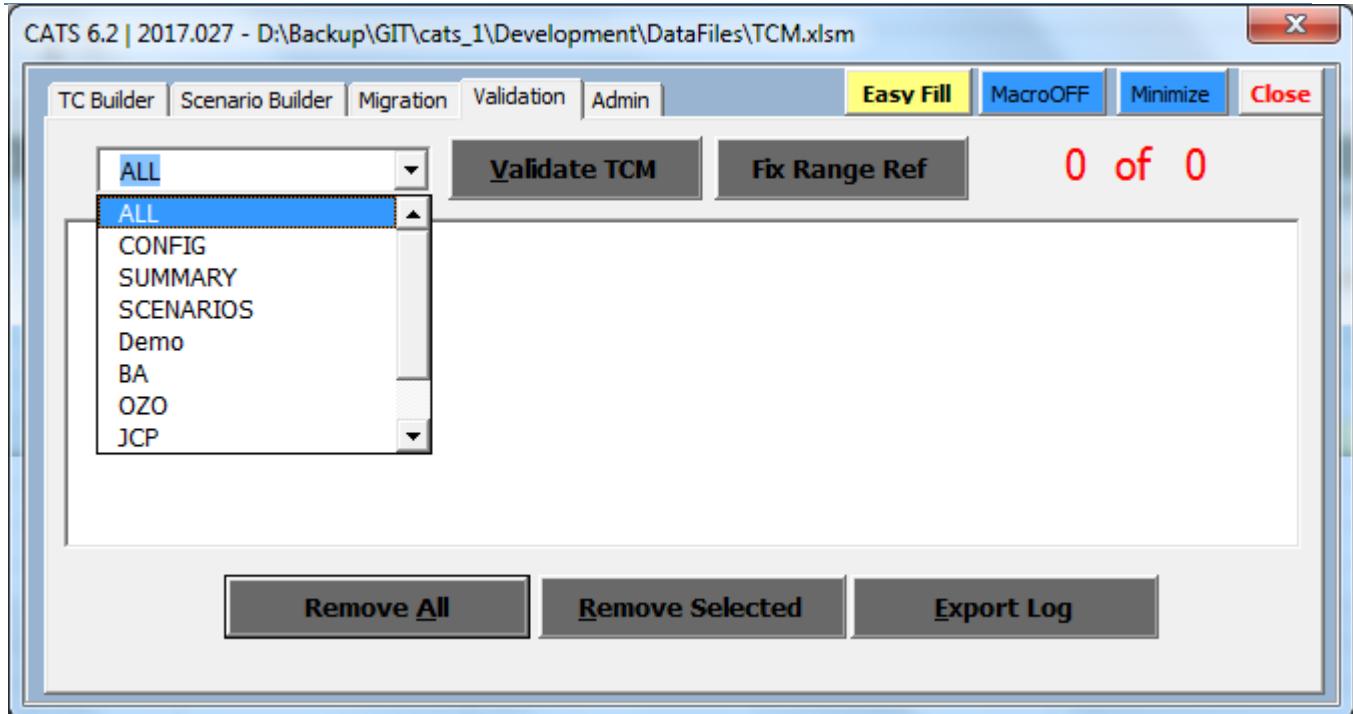
4.6.8 TCM Validation

User can perform the validation against all the sheets or selected sheet, it will list out all the errors below, and upon selection of error it will take user to the specific cell for user to fix the error. Select the "ALL" or specific sheet name from the list and click on **Validate TCM**

The error log can be exported to another spread sheet by clicking on **Export Log**

During migration process or if user has copied or moved the sheet from other TCM then there might be chances the reference from old TCM continue to exist. User can click on **Fix Range Ref** to clean up the external and invalid reference range in the TCM.

TCM Validation for the active sheet can be performed using **ADD-INS > CATS > Validate TCM** or **Ctrl + Shift + V**



4.6.9 Easy Fill

Easy fill is newly introduced feature to help the user to enter create the scripts using a single fields. User can invoke Easy Fill option either from menu *ADD-INS > CATS > Easy Fill* or *Ctrl + Shift + E* or from *TCM UI*.

As user continue to navigate to different columns the easy fill will changes its content or validation applicable to that column, for example when user is on Action column it will have list of actions, whereas when user moves to (using TAB) Control Flow column then it will have predefined control flow keywords and so on.

A	B	C	D	E	F	G
TC ID	TEST CASE NAME	CONTROL FLOW	OR	PAGE	ELEMENT REF	ACTION
1 Google	Launch		Custom		CATS EasyFill	
2				Google		
3				Home		
4						
5			Custom			
6			Custom			
7			Custom			
8						
9						
10						
11						

ACTION

- AcceptAlert
- AEMCompEditorMove
- AEMCompFinderImageDragDrop
- AEMCompFinderImageMoveToTop
- AnalyticsEnd
- AnalyticsStart
- AnalyticsVerifyTag
- AnalyticsVerifyTagFrom

4.6.10 Meta Data

Admin tab allows user to see the METADATA sheet which is highly hidden and it holds the many information used in the macros.

- Action List Table
- TDM Reference
- Service Data Reference
- Variable Reference
- OR Reference
- Column List Values
 - Browsers

-
- II. Channels
 - III. Control Flow
 - IV. Flag
 - V. Clause

Every range in this sheet has defined range, disturbance to the named range could cause the unexpected results. User can see the named ranges by clicking on the *show METADATA Name Range* button

Admin password is: catsadmin

4.6.11 Trouble Shooting

- a. Every single event like adding new sheet, updating sheet, changing selection etc., triggers the macros. In case you want to disable the events tracking you can click on “Macro On” button from TCM UI then click on the Maximize the button and continue working with TCM without macros disturbance.
- b. ADD-INS CATS might appear more than once, it could be due to old TCM version opened simultaneously. Please close all the excel sheets and open the latest one
- c. ADD-INS > CATS contains duplicate menus, again it is due to old TCM hence close all the excel workbooks and open the new TCM and follow migration process.
- d. Scenario builder shows the suite and scenarios in red color even though the sheet exist. This issue could be due to template mismatch.
- e. After OR update the new OR reference is not reflecting in the test case templates. This could be due to old reference you need to Fix Range Ref

5 CATS Script Creation

5.1 Test Case

The CATS script creation involves many steps of which test case creation is one of the critical activity. It decides the complexity involved in maintenance, execution and the reusability of the script.

There are many approaches available to create the automation scripts, some of the popular approaches are listed below.

- a. Page Object Model
- b. Component Based Approach
- c. Feature Based Approach

Irrespective of the model you want to adopt what you need to consider is the reusability of the test steps. To identify the reusable steps one should get familiar with the application beforehand.

You can either use spreadsheet to construct the test case or the direct spread sheet TCM template, we will be covering the way how to write the test case using spread sheet as test case creation using TCM UI is self-explanatory



Setup:

1. Make sure you have all the TCM, OR, TDM files in the folder.
2. Open the TCM.xlsx
3. If you do not have any test case sheet in TCM, create a new sheet with TCM template.
4. Add the test cases
5. Make sure CATSConfig.properties, CATSToolsInput.properties and CATSReport.properties are configured properly

5.1.1 Test Case

When user enters the TC ID it will be considered as start of new test case and it will terminate till the next TC ID appears. As soon as TC ID is entered the mandatory fields will be highlighted.

Add First Test Case:

- a. Enter the Test Case ID, Test Case Name to create the new test case
- b. Select the OR, PAGE and ELEMENTREF (all three fields are reference to ORM)
- c. Select the action to be performed on the object (CATS Action Reference)
- d. Provide the Test Data if required. It can be hard coded data, local variable, global variable, suite variable or TDM reference
- e. Provide the Expected Data / Result if required. It can be hard coded data, local variable, global variable, suite variable or TDM reference
- f. Provide the description of the step if you want to modify any
- g. Flag column can be selected if you want to set exit condition or ignore condition
- h. Select the Service Config ID if the step is related to services testing
- i. To add additional test steps to the test case continue to perform steps b through h in subsequent rows

5.1.2 Control Flow

5.1.2.1 IF...Else...End If

When you want execute certain steps only when some condition is met then you can use the control flow column. Every control flow begins with IF must be closed with END IF.

TC ID	TEST CASE NAME	CONTROL FLOW	OR	PAGE	ELEMENT REF	ACTION	TEST DATA
1	PMFEATURES		Custom			LAUNCHSITE	\$Verify.URL.Field1
						IMPLICITWAIT	10
						VALIDATETITLE	
						VALIDATETITLE	
						SCROLLPAGEDOWN	
						SCROLLPAGEUP	
						REFRESH	
		IF IF NOT ELSE IF ELSE IF NOT ELSE ENDIF	amePage	Home	VerifyCSS	VALIDATECSSVALUE	font size
					VerifyCSS	VALIDATECSSVALUE	font size
					VerifyCSS	VALIDATECSSVALUE	font weight
					VerifyCSS	VALIDATECSSVALUE	color
					VerifyCSS	VALIDATECSSVALUE	color

As seen in figure above, the CONTROL FLOW Column is a drop down. It has various clauses like IF, ELSE IF, ELSE and ENDIF and all the negations of it.

A	B	C	D	E	F	G	
TC ID	TEST CASE NAME	CONTROL FLOW	OR	PAGE	ELEMENT REF	ACTION	TEST DA
1	IF		Custom			LAUNCHSITE PAGELOADWAIT	
		ZohoHome	Home	Login	CLICK		
		Custom			WAIT		
	IF	ZohoHome	Home	UFrameChange	SWITCHIFRAMELOCATOR		
				UVerifyEleFrame	VERIFYELEMENTPRESENT		
		Custom			SWITCHTODEFAULT		
	END IF	ZohoHome	Home	UVerifyEleFrame	VERIFYELEMENTNOTPRESENT		
		Custom			SWITCHIFRAME		
		ZohoHome	Home	UVerifyEleFrame	VERIFYELEMENTPRESENT		
		Custom			SWITCHTODEFAULT		

The figure above shows as an example usage of condition if. The highlighting in yellow is for illustration purpose only. If you see the step SWITCHIFRAMELOCATOR it is subjected to condition IF in the CONTROL FLOW Column.

And every IF clause mentioned must have a closing END IF.

This is analogous to the closing brace of and if block in programming. So the END IF is marked at 3 steps ahead at VERIFYELEMENTNOTPRESENT. So the yellow region acts like if block now.

The functionality is simple, CATS executes the step SWITCHIFRAMELOCATOR and based on its result, executes the following steps listed till ENDIF is encountered. I.e. if it passed the next three steps would be executed, else they would be skipped and step next to ENDIF would be executed next.

Similarly one can use IF ELSE condition also.

2	IFELSE		Custom			LAUNCHSITE PAGELOADWAIT	http://www.zoho.com
		ZohoHome	Home	Login	CLICK		
		Custom			WAIT		5
	IF	ZohoHome	Home	UFrameChange	SWITCHIFRAMELOCATOR		
				UVerifyEleFrame	VERIFYELEMENTPRESENT		
		Custom			SWITCHTODEFAULT		
		ZohoHome	Home	UVerifyEleFrame	VERIFYELEMENTNOTPRESENT		
		Custom			SWITCHIFRAME		0
		ZohoHome	Home	UVerifyEleFrame	VERIFYELEMENTPRESENT		
		Custom			SWITCHTODEFAULT		
		ZohoHome	Home	UVerifyEleFrame	VERIFYELEMENTNOTPRESENT		
	ELSE			UVerifyEleFrame	SWITCHIFRAMENAME zohoiam		
		Custom			VERIFYELEMENTPRESENT		
		Custom			SWITCHTODEFAULT		
		ZohoHome	Home	ULNKEmail	CLICKNCCLICKLINK	ShowTime	
				ULNKEmail	CLICKONSUBELEMENT	SHOWTIME	5
	END IF						

As shown in the figure, if the IF step fails, the steps from where ELSE is listed to the ENDIF will be executed.

3 IFELSELADDER		Custom		LAUNCHSITE	\$Verify.URL.Field1	
				IMPLICITWAIT		10
	IF			VALIDATETITLE SCROLLPAGEDOWN SCROLLPAGEUP REFRESH		
	ELSE IF	HomePage	VerifyCSS VerifyCSS VerifyCSS VerifyCSS VerifyCSS VerifyCSS	VALIDATECSSVALUE VALIDATECSSVALUE VALIDATECSSVALUE VALIDATECSSVALUE VALIDATECSSVALUE VALIDATECSSVALUE	font size font size font weight font weight color color	
	ELSE IF		VerifyCSS VerifyCSS	VALIDATECSSVALUE VALIDATECSSVALUE	font family font family	
		Custom		CHECKTEXTEXIST CHECKTEXTEXIST CHECKLINKEXIST CHECKLINKEXIST CHECKPARTIALLINKEXIST CHECKPARTIALLINKEXIST		
	ELSE IF			CLICKLINK CLICKLINK NAVIGATEBACK	Customer Assistance Cust sistance	
	ELSE			CLICKPARTIALLINK CLICKPARTIALLINK	Assistance Asance	
	END IF			CAPTURESCREEN	SanityTest	

Similarly one can use IF ELSE LADDER also as shown in figure above.

4 NESTEDIFELSELADDER		Custom		LAUNCHSITE	\$Verify.URL.Field1	
				IMPLICITWAIT		10
	IF			VALIDATETITLE SCROLLPAGEDOWN SCROLLPAGEUP REFRESH		
	IF	HomePage	VerifyCSS VerifyCSS VerifyCSS VerifyCSS VerifyCSS VerifyCSS	VALIDATECSSVALUE VALIDATECSSVALUE VALIDATECSSVALUE VALIDATECSSVALUE VALIDATECSSVALUE VALIDATECSSVALUE	font size font size font weight font weight color color	
	ELSE IF		VerifyCSS	VALIDATECSSVALUE VALIDATECSSVALUE	font family font family	
		Custom		CHECKTEXTEXIST CHECKTEXTEXIST CHECKLINKEXIST CHECKLINKEXIST CHECKPARTIALLINKEXIST CHECKPARTIALLINKEXIST		
	ELSE IF			CLICKLINK CLICKLINK NAVIGATEBACK	Customer Assistance Cust sistance	
	ELSE			CLICKPARTIALLINK CLICKPARTIALLINK	Assistance Asance	
	END IF			CAPTURESCREEN	SanityTest	
	END IF					

One can use Nested IF-ELSE too as shown in the figure above.

5	IFNOT		Custom		LAUNCHSITE	http://www.zoho.com
		ZohoHome	Login		PAGELOADWAIT	
		Custom			CLICK	
	IF NOT	ZohoHome	UFrameChange		WAIT	5
			UVerifyEleFrame		SWITCHFRAMELOCATOR	
		Custom			VERIFYELEMENTPRESENT	
		ZohoHome	UVerifyEleFrame		SWITCHTODEFAULT	
			UVerifyEleFrame		VERIFYELEMENTNOTPRESENT	
	END IF	Custom			SWITCHFRAME	0
		ZohoHome	UVerifyEleFrame		VERIFYELEMENTPRESENT	
		Custom			SWITCHTODEFAULT	
		ZohoHome	UVerifyEleFrame		VERIFYELEMENTNOTPRESENT	
			UVerifyEleFrame		SWITCHFRAMENAME zohiam	
		Custom			VERIFYELEMENTPRESENT	
		Custom			SWITCHTODEFAULT	
		ZohoHome	ULNKEmail		WAIT	5
			ULNKEmail		CLICKNCCLICKLINK	ShowTime
					CLICKONSUBELEMENT	SHOWTIME

One can also use the negations of them as shown and various combinations of them as shown above.

5.1.2.2 LOOP

User can use the loop feature through control flow column. There are two types of loops and each loop must be explicitly have the LOOP-END.

- a. LOOP-WHILE: where the loop will continue repeat its execution as far as Action result of the step is PASSED
- b. LOOP-UNTIL: Loop-Until is just opposite to loop-while where the loop continue to repeat its execution till the Action result becomes PASSED

Between LOOP-WHILE and LOOP-END or between LOOP-UNTIL and LOOP-END there are four other control flow can be added.

- a. LOOP-SKIP ON FAIL: The steps between LOOP-SKIP ON FAIL and LOOP-END will be skipped if the Action result is failed
- b. LOOP-SKIP ON PASS: The steps between LOOP-SKIP ON PASS and LOOP-END will be skipped if the Action result is passed
- c. LOOP-BREAK ON FAIL: Loop will terminate beyond LOOP-BREAK ON FAIL skipped if the Action result is FAILED
- d. LOOP-BREAK ON PASS: Loop will terminate beyond LOOP-BREAK ON PASS skipped if the Action result is passed

5.1.2.3 FOREACH

User can user the FOREACH-START feature through control flow column, for block will start from FOREACH-START and the block will end when FOREACH-END is called in control flow.

FOREACH-START can be used for two actions IterateArray and IterateMap.

CONTROL FLOW	OR	PAGE	ELEMENT REF	ACTION	TEST DATA	EXPECTED RESULT
FOREACH-START	Custom			IterateArray	[Apple,Mango,Orange,4]	{}{charVal}}
				SetORVariable	{}{FruitName}}	{}{charVal}}

5.1.3 Object Reference

Once user updates the OR in the TCM UI it will start populating in the OR, Page and Element Ref columns. Based on the OR selected the PAGE column list values will be updated and based on the PAGE value selected ELEMENTREF column list values will be updated

A	B	C	D	E	F	G	H
TC ID	TEST CASE NAME	CONTROL FLOW	OR	PAGE	ELEMENT REF	ACTION	TEST DATA
1	LaunchHomePage	LaunchHomePage	Custom			LAUNCHSITE	\$HomePage.URL.Field1
2						VERIFYTITLE	
3						VERIFYELEMENTPRESENT	
4	VerifyHomePageElements	VerifyHomePageElements	HomePage	SignInPage	HeaderRef	VERIFYELEMENTPRESENT	
5					Logo	VERIFYELEMENTPRESENT	
6					FooterRef	VERIFYELEMENTPRESENT	
7					SearchBox	VERIFYELEMENTPRESENT	
8	SignIn	SignIn	SignInPage			WAIT	10
9					SignInRef	CLICK	
10					UserName	ENTER	\$SignInPage.UserName.Field1
11					Password	ENTER	\$SignInPage.Password.Field1
12					Remember	SELECTCHECKBOX	
13					LoginBtn	CLICK	

A pop-up will appear saying that the OR refresh is complete. When you again now click the OR drop-down in the TCM now, it will list the added new sheet. This is similar to any change in the OR. You can press the shortcut as many times as you wish to refresh.

Sometimes you might run into issues doing this, like the dropdown might say <>ORERROR<> or ask you for an OR path. In such cases make sure your OR is pointed rightly, to check where your OR is pointing currently. Click on Add-Ins > CATS > TCM UI and click on the OR and Actions tab.

Make sure the OR path is pointing rightly to the OR you are using currently. If not browse and point to the right OR. This should resolve any OR dropdown problems if they appear.

OR column can also have valid values “custom” or “Services” though they are not part of object repository. Custom is mentioned when Action doesn’t require any object reference and Services is mentioned when the test step is related to services

5.1.4 Selecting Actions

Choosing actions from the dropdown is pretty easy; the dropdown contains the list of actions that are available in CATS. The list of items displayed in the Title /Camel case are user friendly Action names which will be converted to actual action name once user selects the action

E	F	G	H
PAGE	ELEMENT REF	ACTION	TEST DATA
ShippingPage	FirstName	ENTER	\$ShippingData.FirstName.Field1
	LastName	ENTER	\$ShippingData.LastName.Field1
	Address1	Enter	\$ShippingData.Address1.Field1
	Address2	EnterPress	\$ShippingData.Address2.Field1
	City	EnterTextIntoAlert	\$ShippingData.City.Field1
		ExplicitWait	
		FluentWait	
		GetPageSource	
		GetRandomMailId	
		Hover	
ShippingPage	State		\$ShippingData.State.Field1

The action is the representation of the user's action on the browser or. Examples are clicking a button, entering the text. Each action method represents the ability of the user to interact with the browser in a way. More number of actions, more number of ways to interact with browser.

[CATS Functions Reference.html](#) contains the list of actions that are supported by the CATS tool as of now and will be updated after every release. There is a possibility that more number of actions added in the new versions.

5.1.5 Highlighting Errors

The upgraded TCM will also highlight common errors or prerequisite for a test step or if data has to be filled in a column mandatorily. Pink highlight indicates the field is mandatory and yellow highlight indicates optional fields..

CONTROL FLOW	OR	PAGE	ELEMENT REF	ACTION	TEST DATA	EXPECTED RESULT
	Custom			VerifyTitle		{{\$Data.ItemCode.<<COUNTRY>>}}
	Custom			VerifyPartTextPresent		\$Data.Author.<<COUNTRY>>
	Custom			Quit		
FOREACH-START	Custom			IterateArray	[Apple,Mango,Orange,4]	{{charVal}}}

If Element references are not correct, they will get highlighted in red.

Custom			WAIT		10
ShippingPage	ShippingPage	Zip	SELECTELEMENTBYVALUE	\$ShippingData.State.Field1	
Custom			WAIT		10
ShippingPage	ShippingPage	Zip	ENTER	\$ShippingData.Zip.Field1	
		PhoneNo	ENTER	\$ShippingData.PhoneNo.Field1	
		Country	SELECTELEMENTBYTEXT	\$ShippingData.Country.Field1	

5.1.6 Step Description

When you select an action for a step the description column in the test case sheet automatically gets prepopulated with some description. You can let it be the way it is or add your custom description as to what the step is doing. To achieve that all you have to do is write your description in the description column and make sure it does not start with colon (:) character.

DESCRIPTION
Choose Color
Wait for 2 seconds
Choose Size
Wait for 2 seconds
Enter the Quantity
Wait for 2 seconds
Add to Cart
Store the Price in {{ProductPrice}}
Store the Discount in {{ProductDiscount}}
: Wait
Click on MiniCart

5.1.7 TestData, Expected Result, Service Source

Now, test data and expected result columns in TCM sheet will also display the drop down list provided user selects the checkbox "Enable TDM/Variable List in TCM". To toggle between TDM reference and Variable reference user needs to change the radio button selection. Refer the above screenshot.

For TDM reference user must enter the numeric prefix manually if present in TDM. Sometimes the list in the TCM column might not get updated in such case you need to go to different sheet and then come back to the same sheet.

ELEMENT REF	ACTION	TEST DATA	EXPECTED RESULT
		\$SignIn2.PageTitle.FIELD \$SignIn2.PageTitle.ER \$SignIn2.Username.FIELD \$SignIn2.Username.ER \$SignIn2.Password.FIELD \$SignIn2.Password.ER \$SignIn2.Username FIELD	

5.1.8 FLAGS

The Flag column is extended to have additional values to enable the additional feature in the CATS R5.0. Flag column can have one of below value as valid.

- a. IGNORE TESTCASE
- b. IGNORE REPORT
- c. IGNORE TESTSTEP
- d. EXIT TC
- e. EXIT SCN
- f. EXIT SUITE
- g. REPORT

TEST DATA	EXPECTED RESULT	DESCRIPTION	PRIORITY	FLAGS	SERVICE
\$Verify.URL.Field1		: Launch the Site <\$Verify.URL.Field1>			
10		: Wait implicitly for <10>			
	Ralph Lauren: Shop Clothing for Men, Women, Children & Babies			IGNORE TESTCASE IGNORE REPORT IGNORE TESTSTEP EXIT TC EXIT SCN EXIT SUITE REPORT	
	Testing	Verify failed result of the action			

5.1.9 Ignore Flag

IGNORE TESTCASE: it should be defined at the test case level (not at test step level). If you provide this value against any test step of a test case, that test case will be skipped wherever it is called.

The IGNORE TESTCASE flag can be used during the progression development where the flow is known but the application is not ready hence the placeholder test case with steps is created but marked to ignore.

TC ID	ELEMENT REF	ACTION	TEST DATA	FLAGS
ChangeQuantity	Edit	CLICK		
		WAIT		5 IGNORE TESTCASE
	Quantity	ENTER	\$ShippingData.Quantity.Field2	
	Update	CLICK		
PerformCheckout		WAIT		10
	Checkout	CLICK		
		WAIT		20

Here the WAIT step in the test case ChangeQuantity is given the value IGNORE TESTCASE. Hence ChangeQuantity is skipped wherever it is called. (Use this feature very carefully!)

IGNORE REPORT: If you provide this value against any test step of a test case, the test case will be executed. The execution report will not be pass or fail instead it will report the status as Done (I) or Done (W). I stands for information where the steps executed successfully whereas W stands for warning which indicates that the step execution failed.

This flag is useful for those steps contains P3 or P4 actions or the action is just used for reference purpose only but doesn't influence the actual functionality etc.,

TC ID	OR	ACTION	TEST DATA	FLAGS
LaunchHomePage	Custom	LAUNCHSITE	\$HomePage.URL.Field1	IGNORE REPORT
		VERIFYTITLE		\$HomePage.Title.Field1

As you see in the above screen, the test case LaunchHomePage contains two test steps namely LAUNCHSITE and VERIFYTITLE. The test step LAUNCHSITE is marked as IGNORE REPORT in the FLAGS column. This informs CATS to *execute the test step* but result will be marked as done (I) if successful else Done (W) if failed

IGNORE TEST STEP: Flag set to IGNORE TESTSTEP will ignore the test step execution and it won't appear in the report as well.

This flag is useful when you have known defect or the object property is not modified from the previous execution and you don't want to run then you can set the flag to ignore test step.

5.1.10 Exit Flag

EXIT TC: This value is introduced as part of exit features, when user doesn't want to continue the further steps in the test case when a particular step fails. The "EXIT TC" should be marked against the critical steps to decide whether to execute further steps or not based on the execution status of the step.

The result set as TC TERMINATED and other test cases are skipped.

Scenario: SCENARIOS_3 Verify that Order confirmation page is opened when click on Place order in Order review Page and verify the Order number and amount						
Condition	Action Description	Action	Test Data	Expected Result	Actual Result	Result
Test Case: HOMEPAGE_1 HOMEPAGETest						
	Launch Amazon.in	LAUNCHSITE	https://www.snapdeal.com		Site launched success	PASSED
	Validate Title of the home page	VALIDATETITLE		Online Shopping: Shop Online for Mobiles, Books, Watches, Shoes and More - Amazon.in	Actual page title is : Online Shopping - Largest Online Shopping Site for Electronics, Mobile, Fashion & Home - Snapdeal	TC TERMINATED
	: Wait implicitly for <\$HomePage.ImplicitWait.FLD1>	IMPLICITWAIT			10	SKIPPED
Test Case: HOMEPAGE_14 HOMEPAGETest						
	Launch Amazon.in	SETVARIABLEVALUE	URL2	URL2	Variable "<>" is set to "URL2"	PASSED
	Validate Title of the home page	LAUNCHSITE	http://www.google.com		Site launched success	PASSED
	: Wait implicitly for <\$HomePage.ImplicitWait.FLD1>	IMPLICITWAIT	10			PASSED
	: Verify <<CURR_SUITE>> is matching with ex	VERIFYVARIABLEVALUE	ChromeDesktop	ChromeDesktop	Variable "<>" is as expected	PASSED

EXIT SCN: This value is introduced as part of exit features, when user doesn't want to continue the further steps as well as further test cases in the test scenario when a particular step fails. The "EXIT SCN" should be marked against the critical steps to decide whether to execute further steps or not based on the execution status of the step. Below is an example

TC ID	TEST CASE NAME	ACTION	TEST DATA	EXPECTED RESULT	FLAGS	REMARKS
1	TC1	LAUNCHSITE	\$Verify.URL.Field1			
		IMPLICITWAIT		10		
		VALIDATETITLE		Ralph Lauren: Shop Clothing for Men, Women, Children & Babies		
		VALIDATETITLE		Testing	ON ERROR EXIT SCENARIO	
		SCROLLPAGEDOWN				
		SCROLLPAGEUP				
		REFRESH				
		VALIDATECSSVALUE	font size	10px		
		VALIDATECSSVALUE	font size	9px		
		VALIDATECSSVALUE	font weight	400		
		VALIDATECSSVALUE	font weight	399		
		VALIDATECSSVALUE	color	rgba(144, 144, 144, 1)		
		VALIDATECSSVALUE	color	rgba(150, 144, 144, 1)		
		VALIDATECSSVALUE	font family	\"UniversLTW01-53Extended\"		
		VALIDATECSSVALUE	font family	\"UniversLT-53Extended\"		
		CHECKTEXTEXIST		Sign In Register		
		CHECKTEXTEXIST		Siln Rister		
		CHECKLINKEXIST		Change Country		
		CHECKLINKEXIST		Ch ountry		
		CHECKPARTIALLINKEXIST		Assistance		
		CHECKPARTIALLINKEXIST		Asnce		
		CLICKLINK	Customer Assistance			
		CLICKLINK	Cust sistance			

To use the feature, the user is supposed to mention the "ON ERROR EXIT SCENARIO" or "EXIT SCN" in the flags column against a test step. Suppose the above test case is part of scenario1 mentioned below

ID	SCENARIO NAME	TEST CASES
1	VerifyTestcaseRepetitionInScenarioSheet	Script_1,Script_2,Script_3,Script_5
2	VerifyLinkAndWindowActions	HomePageTest_1, HomePageTest_2

Then, if the step fails the rest of the steps of script_1 will be skipped and the test cases Script_2, Script_3, Script_5 will also be skipped. And the execution of the suite will continue with the next scenario (if it exists)

TEST SUITE	TEST CASES - SCENARIOS	CHANNEL	BROWSERS	ITERATIONS	RUN
4 VerifyAllBasicActions	Script_1,Script_2,Script_3	Desktop	Chrome	1	N
4 VerifyFrameActions	Script_4,Script_5	Desktop	Chrome	1	N
4 ExampleSuite	SCENARIOS_1,SCENARIOS_2	Desktop	Chrome	1	N
4 VerifyLinkNWindowActions	SCENARIOS_2	Desktop	Chrome	1	N

In the example suite, the scenarios1 will be skipped after the step flagged above fails. And the scenarios2 will continue execution normally as shown in the report snapshots shown below.

Scenario: SCENARIOS_1										
Test Case: Script_1			Test Case Name: TC1							
Condition	Action Description	Action	Test Data	Expected Result	Actual Result	Result	Functional Failure	Script Failure	Duration	Screenshots
	: Launch the Site <Verify.URL.field1>	LAUNCHSITE	http://www.ralphlauren.com		Site launched success	PASSED	NO	NO	0 min, 9 sec	
	: Wait implicitly for <10>	IMPLICITWAIT	10			PASSED	NO	NO	0 min, 0 sec	
		VALIDATETITLE		Ralph Lauren: Shop Clothing for Men, Women, Children & Babies	Page title as expected	PASSED	NO	NO	0 min, 0 sec	
	Verify failed result of the action	VALIDATETITLE		Testing	Actual page title is : Ralph Lauren: Shop Clothing for Men, Women, Children & Babies SCENARIO TERMINATED : Skipping Rest of Test steps and Test Cases	FAILED	YES	NO	0 min, 0 sec	Screenshots

Scenario: SCENARIOS_2										
Test Case: HomePageTest_1			Test Case Name: VerifyLogin							
Condition	Action Description	Action	Test Data	Expected Result	Actual Result	Result	Functional Failure	Script Failure	Duration	Screenshots
	: Launch the Site <http://www.qa.way2automation.com/>	LAUNCHSITE	http://www.qa.way2automation.com/		Site launched success	PASSED	NO	NO	0 min, 2 sec	
		VALIDATETITLE		Welcome to the Test Site	Page title as expected	PASSED	NO	NO	0 min, 0 sec	
	: Click Link on <Signin>	CLICKLINK	Signin			PASSED	NO	NO	0 min, 0 sec	

EXIT SUITE: This value is introduced as part of exit features, when user doesn't want to continue the further steps in the current test case, further test cases in the test scenario as well as the further scenarios in the test suite when a particular step fails. The "EXIT SUITE" should be marked against the critical step to decide whether to execute or terminate the suite execution.

Status will be marked as SUITE TERMINATED and remaining steps in the suite are Skipped.

Condition	Action Description	Action	Test Data	Expected Result	Actual Result	Result
Test Case: HOMEPAGE_1 HOMEPAGELaunchHomePage						
	Launch Amazon.in	LAUNCHSITE	https://www.snapdeal.com		Site launched success	PASSED
	Validate Title of the home page	VALIDATETITLE		Online Shopping: Shop Online for Mobiles, Books, Watches, Shoes and More - Amazon.in	Actual page title is : Online Shopping - Largest Online Shopping Site for Electronics, Mobile, Fashion & Home - Snapdeal	SUITE TERMINATED
	: Wait implicitly for <\$HOMEPAGE.ImplicitWait.FL1>	IMPLICITWAIT			10	SKIPPED
Test Case: HOMEPAGE_14 HOMEPAGETest						
	Launch Amazon.in	SETVARIALEVALUE		URL2	<>Key>	SKIPPED

5.1.11 Production Monitoring Flag

REPORT: The flag value set to "REPORT" is part of Production Monitoring feature. User needs to setup the configuration to make the execution as production monitoring. There is no change at the execution process for production monitoring. The only change is that the execution report will be based on the steps flagged as REPORT only, other failures are just ignored.

5.1.12 Single Suite on Multiple Browser:

To run same test suite for multiple browsers at a time, we can give multiple configurations for browser in summary sheet as shown below:

A	B	C	D	E	F	G
1 RELEASE	TEST SUITE	TEST CASES - SCENARIOS	ITERATIONS	RUN	CONFIG	BATCH
2 DEMO	DemoHomePage	SCENARIOS_1		N	CONFIG_LocalChrome;CONFIG_LocalFirefox;CONFIG_LocalIE	B2
3 DEMO	testScenario	SCENARIOS_3		Y	CONFIG_LocalChrome	B1
4 SauceLabs	MobileSauceLabs	MOBILE_1	1	N	CONFIG_MOBILE_1	
5 SauceLabs	ChromeSauceLabs	SCENARIOS_3	1	N	CONFIG_Chrome	
6 SauceLabs	FirefoxSaucelabs	SCENARIOS_3	1	N	CONFIG_Firefox	
7 ServicesDemo	RESTDEMO	SCENARIOS_13	1	N		
8 ServicesDemo	xmlFile Suite	SCENARIOS_20	1	N		
9 DEMO	ProductionMonitor	SCENARIOS_24		N	CONFIG_LocalChrome	
10 DEMO	testScenario1	SCENARIOS_1		N	CONFIG_SLWIN;CONFIG_SLWIN2	B1

5.2 Test Data

Test data and expected result column requires the data based on the action selected. The data can be passed to the step in below ways

- a. Hardcoded Data and Excel Formula
- b. Variables
 - a. Global
 - b. Local
 - c. OR
 - d. Random Number
 - e. Iterating variable (increment/decrement)
 - f. Suite
 - g. Reserved
- c. TDM Values
 - a. Reference
 - b. Preserved
 - c. Dynamic
 - d. TDM ref as variable

5.2.1 Hardcoded Data

User can enter the hardcoded string in the test data and expected result, the data will be used as is during the execution

Excel formula can also be used as the test data, expected result and in TDM values. The formula resulted value during data process will be taken and **not** the formula value at the time of execution.

5.2.2 Variables

User can use the variables in the test data and expected result column. Variables are of four type each one can be identified using its prefix and suffix. Single cell can have more than one variable defined and during runtime all the variables will be replaced with its value and resulted value will be used as test data and expected result.

Local variables: it is prefixed with {{and suffixed with}}, for example *{{Variable Name}}* . The scope of the variable is limited to the suite.

Global Variable: It is prefixed with {{{and suffixed with}}}, for example *{{{Variable Name}}}* . The scope of the variable is across the execution. This variable can be used either in TCM test data, expected result or in ORM dynamic properties

OR Variable: It is prefixed with the {{and suffixed with}}, for example *{{Variable Name}}* . The scope of the variable is limited to ORM workbook dynamic properties. To set and rest the OR variables you need to use the Actions SETORVARIABLE, RESETORVARIABLE for more details refer the CATS Actions reference. Even global variables can be used in the ORM dynamic properties

Iterating Variable: These variables can be used when the user want to increment or decrement the variable value every time it is called. The access to the variables are limited to the suite.

For Example:

<<++:var1>> - every call it will increase the value by 1
<<+5:var2>>- every call it will increase the value by 5
<<--:var3>>- every call it will decrease the value by 1
<<-2:var2>>- every call it will decrease the value by 2

Incase user wants to use the just the value without increment, then user needs to use the following pattern <>:<variableName>>, \$PDP.KEY.Field<>:<VariableName>>

Suite Variable: It is defined in the *Suite Variable* column in the *Summary* sheet, multiple variables are separated by semicolon and key and values are separated by equal. Suite variables are loaded before test case execution starts. For example ENV=QA;LANG=EN. These variables can be accessed using runtime by prefixing <>and suffixing with>>. For example <>ENV>>, these variables can be used in the TDM reference as well like **\$PDP.<>ENV>>.<>LANG>>**, **\$<>ENV>>.USER.Field1**, etc.,

RELEASE	TEST SUITE	TEST CASES - SCENARIOS	CHANNEL	BROWSERS	RUN	SUITE VARIABLES
DEMO	suite1	SCENARIOS_3	Desktop	Chrome	Y	Sheet=HomePage;Key=URL
DEMO	FirefoxDes	SCENARIOS_3	Desktop	Firefox	N	
SauceLabs	MobileSauc	MOBILE_1	RemoteWebDriver	N		

Reserved Variable: It is defined during the execution and access it limited to the suite level. Reserved variables too use the same syntax as Suite variable, except it has the reserved keywords.

- a. CURR_SUITE : It gives the currently executing suite
- b. CURR_SCENARIO: to get the currently executing scenario id.
- c. CURR_TCNAME: It gives the currently executing test case name
- d. CURR_BROWSER: It gives the current browser name.
- e. CURR_CHANNEL: It gives the current channel.
- f. CURR_DATAKEY: It gives the current Test Data Key it refers to

Test Case: HOMEPAGE_14 HOMEPAGETest					
	Launch Amazon.in	SETVARIABLEVALUE	URL2	URL2	Variable "<>Key>>" is set to "URL2"
	Validate Title of the home page	LAUNCHSITE	http://www.google.com		Site launched success
	: Wait implicitly for <\$HomePage.ImplicitWait.FILED1>	IMPLICITWAIT	10		
	: Verify <>CURR_SUITE >>> is matching with expected result	VERIFYVARIABLEVALUE	ChromeDesktop	ChromeDesktop	Variable "<>CURR_SUITE>>" is as expected
	: Verify <>CURR_TCNAME >>> is matching with expected result	VERIFYVARIABLEVALUE	HOMEPAGETest	HomePageTest	Variable "<>CURR_TCNAME>>" is as expected
	: Verify <>CURR_BROWSER >>> is matching with expected result	VERIFYVARIABLEVALUE	Chrome	Chrome	Variable "<>CURR_BROWSER>>" is as expected
	: Verify <>CURR_SCENARIO >>> is matching with expected result	VERIFYVARIABLEVALUE	SCENARIOS_3	SCENARIOS_3	Variable "<>CURR_SCENARIO>>" is as expected

5.2.3 TDM Reference

TDM Reference: TDM reference can be given in the test data or expected result column. The format to refer the TDM reference is dollar symbol followed by sheet name followed by dot followed by key name followed by dot followed by field name. Example **\$HomePage.UserName.Field1**, **\$Sheet.<>SuiteVar>><>ScenarioVar>>.Field<>:<IncrementVariable>>**

Now you can update the TDM reference during runtime, and updated value will be used when the same reference is used in other place. The scope of the updated variable will be limited to the

current execution. Actions to be used SetTDMRef and SetTDMRefValue for more details refer the CATS Actions reference.

TDM Dynamic: Refer the Suite Variable section

Preserved TDM Reference: When user provides the test case and scenario data for iteration, all the TDM reference used in the test cases will be overridden by the field reference provided. In case user doesn't want to change the TDM reference irrespective of when the test case data key or scenario test data key is supplied then user can simply add dollar at the end of TDM reference. For example **\$HomePage.User.Field1**, the same functionality can be attained by using TDM Ref as variable.

TDM Ref as Variable: Like local or global variable it needs to be prefixed and suffixed with curly braces. This can be used when you want to construct the test data or expected result using more than one variable and string combinations.

For example {{\$Home.FirstName.Field1}}-{{\$Home.LastName.Field1}}-{{Local Variable}}

TDM variables will be static it won't have any effect based on the test case data and scenario data passed through scenario sheet.

5.2.4 Random Number

The user can request for a random number from anywhere inside the test case generation. The random number is generated at the run time.

The user uses the notation {{RANDOM}}, to get the next random number. This notation can be appended with any other string, and variables. This keyword can be used either in test data or Expected result.

5.3 Scenario

The test scenario is created by combining more than one test case in the required sequence, it could be a user journey involved in multiple component navigation or it could be simple feature required multiple section of involvement or could be a simple flow involved in multiple reusable navigational flows.

Add First Scenario:

- a. Enter the ID, Scenario Name to create the new scenario
- b. Enter the list of test cases you want to associate to the scenario. Multiple test cases are separated by comma.
- c. Test cases are mentioned by its Sheet name followed by underscore followed by TC ID
- d. If the test data or expected result is from TDM reference then user can override the TDM reference data from one field to other by overriding the data field. Add equal followed by the Field Name after TC ID. (HomePage_1=Field8)
- e. Particular test case can be iterated through multiple

ID	SCENARIO NAME	SCENARIO DESCRIPTION	TEST CASES	TEST DATA KEY
AccountCreation	Account creation with multiple cards	Create three account with different personal details, having master and visa card with Texas address	HomePage_Launch, HomePage_Login, AccountMgmt_PersonalDetail, AccountMgmt_AddCreditCards=(Field4;Field5), AccountMgmt_AddAddress=Field2 HomePage_Logout	Field1;Field2;Field3

- f. Test Data Key can be given at case level or scenario level. Numeric digit(s) of the data key part will be replaced during run time. For example your step has TDM reference like **\$Sheet.Key1.Col1**
- If Test Data Key is 4;7;10 then it will execute for **\$Sheet.Key1.Col4**, **\$Sheet.Key1.Col7** and **\$Sheet.Key1.Col10**
 - If Test Data Key has range like 4 TO 7;10 then it will execute for **\$Sheet.Key1.Col4**, **\$Sheet.Key1.Col5**, **\$Sheet.Key1.Col6**, **\$Sheet.Key1.Col7** and **\$Sheet.Key1.Col10**
 - If Test Data Key has range with interval 4 TO 7 BY 2 ;10 then it will execute for **\$Sheet.Key1.Col4**, **\$Sheet.Key1.Col6** and **\$Sheet.Key1.Col10**
 - If you want to iterate ROW wise then you have to do a small change to your reference like **\$Sheet.Key<<CURR_DATAKEY>>.Col1\$**
 - If you miss to add \$ at the end in row iteration pattern then it will iterate diagonally example iii above will become like **\$Sheet.Key4.Col4**, **\$Sheet.Key6.Col6** and **\$Sheet.Key10.Col10**

Test Cases Explained:

- Highlighted in Blue is Test Case sheet name
- Underscore is sheet and test id separator
- Highlighted in Green is Test Case id from the sheet
- Comma is used to separate multiple test case
- Equal is a Test Case data separator
- Highlighted in Amber is Test Case level data override and an iterator, this will take higher precedence than TEST DATA KEY column. The specific test case will iterate for all the field values given here for every scenario data combinations
- Semicolon is used for multiple data key separator
- Highlighted in Red is Scenario level data override and an iterator. All the test cases will be executed in the same sequence as given in the test cases column for every test data key given here.

ID	SCENARIO NAME	TEST CASES	TEST DATA KEY	SCENARIO VARIABLES	LABELS
productIDSearch	Verifying product Search Scenario	Home_Launch, Home_Navigate, Home_Search	4;5	search=productID;country=UK	Sanity;Regression;Search;
productNameSearch	Verifying product Search Scenario	Home_Launch, Home_Navigate, Home_Search	2;4	search=productName	Regression;Search;
productBlankSearch	Verifying product Search Scenario	Home_Launch, Home_Navigate, Home_Search	2;4		Regression;Search;

Scenario Variables and Labels:

- Like suites user can also define the scenario level variables that will take higher precedence over suite variable. Syntax to access the scenario variable remains same as suite variable. For example if you want to access scenario variable in TCM then you would need to refer **<<search>>**
 - When scenario# productIDSearch is being executed **<<search>>** will return **productID**
 - When scenario# productNameSearch is being executed **<<search>>** will return **productName**
 - When scenario# productBlankSearch is being executed **<<search>>** will return the value from suite variable if defined else blank.

-
- b. Labels column is used to perform the target execution or restrict the execution of scenarios to the labels passed through command line or CATS Config.

6 CATS BDD

6.1 CATS BDD

CATS BDD automation script development allows user to create traditional feature files and step definition.

6.1.1 Installation

Install Eclipse IDE with cucumber add-ins for the faster script creation.

6.1.2 Config Changes

All the features files must be in single folder (nested folders are not supported) and all the step definition files must be in a single folder (nested folders are not supported). Update these two folder locations in `./core/config/default/CATSToolsInput.properties`

```
#BDD Configuration
FEATURES_FOLDER_PATH=../DataFiles/Features
STEPDEFS_FOLDER_PATH=../DataFiles/StepDef
```

6.1.3 Create Feature

CATS Supports the standard feature file, for more details on how to create feature files please refer [Gherkin](#) language.

6.1.4 Step Def Creation from TCM

You can convert your TCM java methods which can be mapped to step definitions. To migrate your TCM sheet to Step Definitions then navigate to Utilities in CATS UI, click on Browse TCM and then click on Generate StepDef button. Each TCM sheet is converted into a step definition file without mapping.

6.1.5 Create Step Definition

Step definition creation requires basic knowledge on Java, for better start open the sample step definition file and continue to add your step definitions. Below code snippet is for your reference, any **step definition file** **must** contain the code highlighted in sky-blue to utilize the CATS framework

```

package com.sapient.qa.cats.core.seleniumbdd.stepdef;

import com.sapient.qa.cats.core.framework.CATSCucumberConfig;
import com.sapient.qa.cats.core.utils.ThreadVariables;
import cucumber.api.Scenario;
import cucumber.api.java.AfterStep;
import cucumber.api.java.Before;
import cucumber.api.java.BeforeStep;
import cucumber.api.java.en.Given;
import cucumber.api.java.en.Then;

public class CATSSteps extends CATSCucumberConfig {

    @Before()
    public void launch(Scenario scenario) {
        catsBeforeScenario(scenario); // this must be called all the time to invoke CATS framework
        disableAutoAssertion();
    }

    @BeforeStep
    public void beforeStep(Scenario scenario) {
        catsBeforeStep(); // this must be called all the time to invoke CATS framework
    }

    @AfterStep
    public void afterStep() {
        catsAfterStep(); // this must be called all the time to invoke CATS framework
    }

    @Given("your step definition reference from feature file {argument}")
    public void method_name(String url) {
        catsAction.launchSite(url);
        catsAction.pageLoadWait();
    }

}

```

1. Open the sample step definition file in eclipse or copy the above code in a new class file
2. Add CATS/core/automatonplus-core-all.jar to your Java Build Path as external jar
3. Start defining the test steps. CATS actions are exposed through an object *catsAction*

6.1.6 Using CATS Actions

CATS actions are exposed through *catsAction* object, it exposes the wrapper actions customized with limited arguments for cucumber. However user can still pass all the arguments for any action using the generic action methos.

```

@Given("your step definition reference from feature file {argument}")
public void method_name(String url) {
    #CATS action accessing through wrapper
    catsAction.launchSite(testData);

    #Verifies menu should match "Mens" where object reference is from ORM
    catsAction.verifyElementsTextValues("Home.Header.Menu1", "", "Mens");

    #Verifies object contains "Mens" where object reference is using By.id
    catsAction.verifyElementsTextValues(By.id("menu_1"), "contains", "Mens");

    #Verifies object contains TDM Reference where object reference is using By.id
    catsAction.verifyElementsTextValues(By.id("menu_1"), "contains",
"$Home.Menu.field1");

    #Using selenium methods without using CATS Action
    getDriver().findElement(By.xpath("//img[6]")).click();

    #Explicit reporting for non-CATS actions
    boolean status;
    String fitDesc = getDriver().findElement(By.xpath("//div[@data-test='detailsTab']")).getText();
    if (fitDesc.equalsIgnoreCase(fit))
        status = catsAction.reportResultPass("Verify the Fit", "", fit, fitDesc);
    else
        status = catsAction.reportResultFail("Verify the Fit", "", fit, fitDesc);

    #Explicit reporting for non-CATS actions using true/false status
    status = fitDesc.equalsIgnoreCase(fit)
    catsAction.reportResult(status, "Verifyin the Fit attribute", "", fit, fitDesc);

    #Generic way of calling CATS Action
    catsAction.action(desc, actName, orRefName, testData, expResult, exitFlag, svcConfig,
controlFlow, servSource);

```

6.1.7 CATS Suites Setup

The summary sheet is used to define the test suites, the actual execution flow sequence will begin from here.

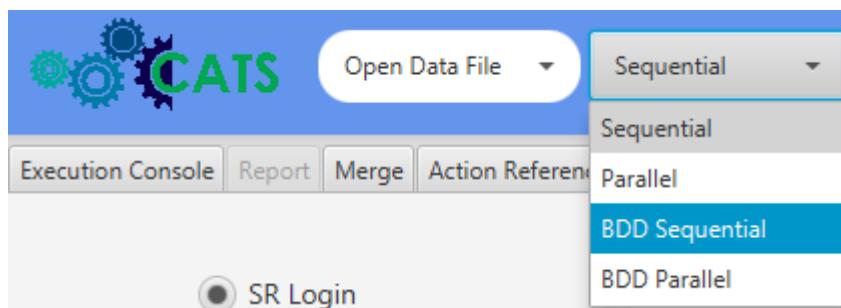
User needs to provide the suite level configuration details in this sheet.

Column Name	Description
RELEASE	This hold the release# or build # which is used to print in the execution report.
TEST SUITE	Name of the test suite which is mandatory and it should not have any special characters.
TEST CASES - SCENARIOS	This column contains reference to the FEATURE FILES. Multiple FEATURES will be separated by comma. Feature file location can be either absolute path or it can be relative path to CORE FOLDER.

RUN	Whether suite is to be executed or not. Valid value could be "Y" or "N"
CONFIG	Configuration ID must be provided which holds the channel and browser information
BATCH	Suites with the same batch name will be grouped for execution. Batch name suffixed with _SEQ will be executed in sequential mode this is mainly used when you want to execute suite one by one due to inter dependency.
OWNER	Optional. It is for user reference
SUITE VARIABLES	User will have access to some of runtime variables like CURR_SUITE, CURR_TCNAME, CURR_SCENARIO, CURR_BROWSER, CURR_ENV (if passed through cmd arg) along with the user can also define additional variables prior to execution start. KEY1=VALUE1;KEY2=VALUE2
DESCRIPTION	Optional. It is for user reference
LABELS	Valid TAG expression to runner class. Refer Click here for tag-expression

6.1.8 BDD Execution

CATS BDD can be executed either using command line or CATS UI. To run BDD from CATS UI user needs to select either BDD Sequential for single thread execution and BDD Parallel for multithread execution.



To trigger BDD execution from command line either you can use `start_cats` or `start_cats_multithread` with the argument name mode with value bdd. Please refer the section [Command Line Execution](#)

6.1.9 BDD Auto Assertion

By default auto assertion is disabled in CATS BDD which means, all the steps will continue to execute even when after step encounters failure from CATS steps. However user has a choice to enable the auto assertion by calling a `enableAutoAssertion` method inside the `@Before` annotation.

```
public class CATSSteps extends CATSCucumberConfig {
    @Before()
    public void launch(Scenario scenario) {
        catsBeforeScenario(scenario); // this must be called all the time to invoke CATS framework
        disableAutoAssertion();
        enableAutoAssertion();
    }
}
```

Enable or disable of auto assertion only controls the CATS actions, it doesn't have any control on the user defined assertions or any unhandled exception caused through the code block created by the user

6.1.10 BDD Report

CATS BDD report can be located inside output/BDDReport/Cucumber-html-reports. CATS Step level report is embedded into the cucumber HTML report and failure screenshot can be accessed through the given against each failed step.

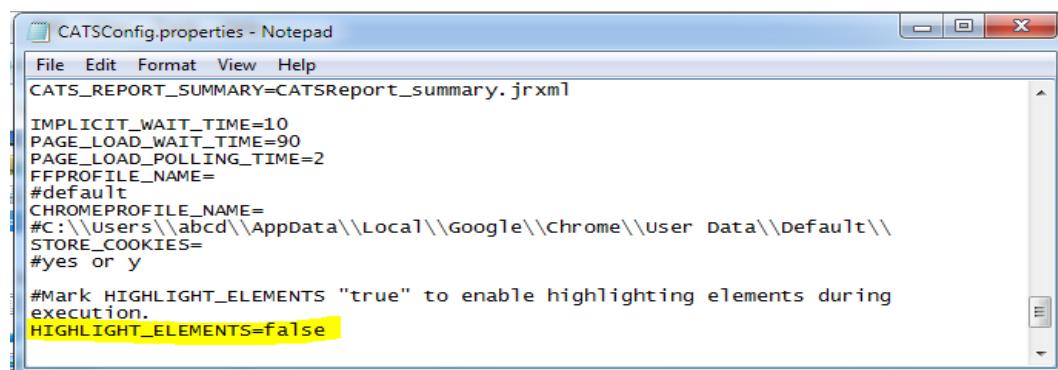
7 CATS Configurations

CATS configurations are now moved to config folder under core.

7.1 CATSConfig

7.1.1 Highlighting Elements

You can configure CATS to highlight the elements under test on the webpage before any action is performed on them. This can be turned on or off as per user requirement. In the CATS/Core folder, you will find a file called CATSConfig.properties



```
CATSConfig.properties - Notepad
File Edit Format View Help
CATS_REPORT_SUMMARY=CATSReport_summary.jrxml
IMPLICIT_WAIT_TIME=10
PAGE_LOAD_WAIT_TIME=90
PAGE_LOAD_POLLING_TIME=2
FFPROFILE_NAME=
#default
CHROMEPROFILE_NAME=
#C:\\Users\\abcd\\AppData\\Local\\Google\\chrome\\User Data\\Default\\
STORE_COOKIES=
#yes or y
#Mark HIGHLIGHT_ELEMENTS "true" to enable highlighting elements during
execution.
HIGHLIGHT_ELEMENTS=false
```

In it there is a property called HIGHLIGHT_ELEMENTS. The default value is false; you can set it to true to enable the highlighting of elements during execution.

The elements will be highlighted in a green rectangular box before any action is performed on it.

7.1.2 Direct Scenario Execution / Label restricted Execution

When user wants to perform the execution directly from Scenario sheet using CATS UI then, you have to update EXECUTE_FROM_SCENARIO_SHEET_DIRECTLY =YES and at least one browser from CONFIG sheet must be set to YES in DEFAULT column

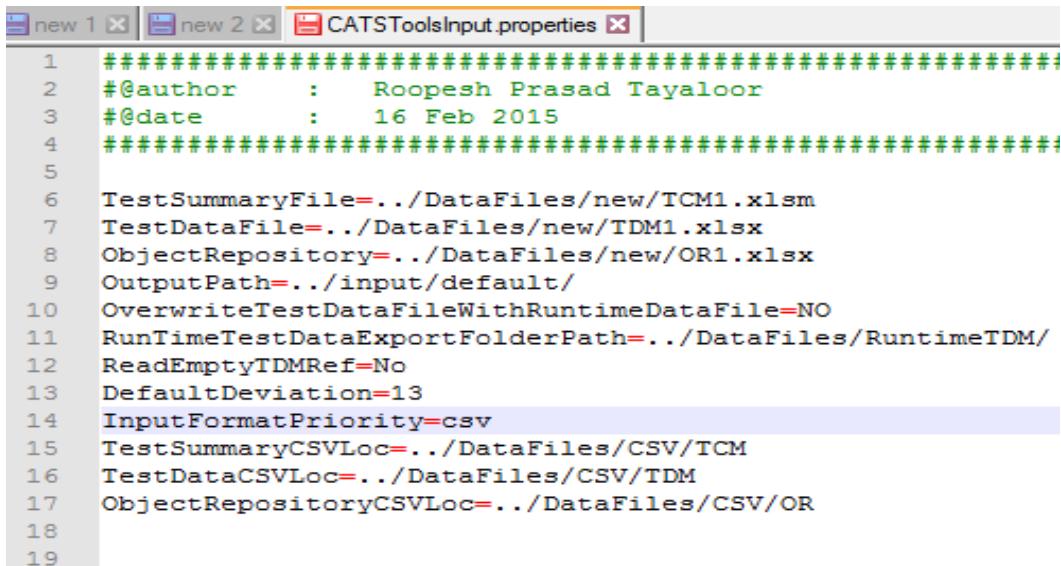
name	value
EXECUTE_FROM_SCENARIO_SHEET_DIRECTLY	NO
SUITE_LABELS	
SCENARIO_LABELS	

While execution is triggered all the Scenarios will be executed from the scenario sheet and every single scenario will act as independent suite. To execute the selective scenarios user needs to define the labels against which the execution to be performed

SECNARIO_LABELS=checkout,myaccount

Restricted suite execution can be achieved through suite labels, for example all the suites are set to run in TCM Summary sheet however user wants to execute only sanity. User needs to set the labels against **SUITE_LABELS=sanity** so that the suites matching the label only be executed.

7.2 CATSToolsInput



```
new 1 new 2 CATSToolsInput.properties
#####
#author : Roopesh Prasad Tayaloor
#date : 16 Feb 2015
#####
TestSummaryFile=../DataFiles/new/TCM1.xlsm
TestDataFile=../DataFiles/new/TDM1.xlsx
ObjectRepository=../DataFiles/new/OR1.xlsx
OutputPath=../input/default/
OverwriteTestDataFileWithRuntimeDataFile=NO
RunTimeTestDataExportFolderPath=../DataFiles/RuntimeTDM/
ReadEmptyTDMRef=No
DefaultDeviation=13
InputFormatPriority=csv
TestSummaryCSVLoc=../DataFiles/CSV/TCM
TestDataCSVLoc=../DataFiles/CSV/TDM
ObjectRepositoryCSVLoc=../DataFiles/CSV/OR
#####

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
```

7.2.1 Excel Data File

Key input for the CATS framework is Data Files TCM, TDM and OR. These data files must be in MS Excel format. TDM and OR in xlsx format whereas TCM should be in xlsm (macro enabled)

The location where these files are kept must be defined in CATSToolsInput.properties and the location can be either absolute path or relative path from inside the core folder.

Test Case Management (TCM) **excel path** should be mentioned in **TestSummaryFile**

Test Data Management (TDM) **excel path** should be mentioned in **TestDataFile**

Object Repository Management (OR) **excel path** should be mentioned in **ObjectRepository**

7.2.2 CSV Data Files

To use data files in While CATS framework supports the Data Files in excel formats, it has been extended now to support CSV files as well. However CSV files with respect to TCM, TDM and OR must be placed in separate folders.

TCM Folder must have SUMMARY.csv, CONFIG.csv, SCENARIOS.csv and Test Case csv files. Since the files are kept in .csv formats user has to more cognizant about referring the test case name and scenario names. Every other CSV file will be treated as a separate sheet in the excel workbook.

To enable CSV data files feature user needs to set the **InputFormatPriority=csv** anything other than CSV will be treated as normal excel files.

TCM **Folder location** where csv files are placed to be mentioned in **TestSummaryCSVLoc**

TDM **Folder location** where csv files are placed to be mentioned in **TestDataCSVLoc**

OR **Folder location** where csv files are placed to be mentioned in **ObjectRepositoryCSVLoc**

7.2.3 YAML or JSON Data Files

To enable JSON or YAML support TCM (Config, Summary, Scenarios, {testcass}), TDM sheets, ServiceData and ORM to respective json and yaml format. Sample structure is given in along with CATS installation for reference.

InputFormatPriority has to be changed to YAML or JSON and folder where TCM, TDM and ORM files are placed should be updated against TestSummaryDir, TestDataDir and ObjectRepositoryDir properties respectively.

All the excel columns and rows are to be converted in to key-value pairs. Below is the mapping for reference

Excel_Fields	YAML / JSON	Excel_Fields	YAML / JSON
TCM-CONFIG		TCM-SUMMARY	
ID	id	RELEASE	release
DESCRIPTION	description	TEST_SUITE	testSuite
CHANNEL	channel	TEST_CASES_SCENARIOS	scenarios
BROWSER	browser	SCENARIOS	scenarios
VIEWPORT	viewport	TEST_CASES	testcases
DRIVER	driver	TEST_STEPS	teststeps
DEFAULT	default	RUN	run
VERSION	version	CONFIG	config
PLATFORM	platform	BATCH	batch
UDID	udid	OWNER	owner
BUNDLEID	bundleid	SUITE_VARIABLES	suiteVariables
APPPATH	apppath	LABELS	labels
PROTOCOL	protocol	TCM-SUMMARY	
SERVER	server	SCENARIO_NAME	scenariename
PORT	port	SCENARIO_DESCRIPTION	scenariodescription
CONTEXT	context	TEST_DATA_KEY	testdatakey
PKGNAME	pkgname	SCENARIO_VARIABLES	scenarioVariables
ACTIVITY	activity	TCM-TEST CASE	
REMOTEWEBDRIVER_URL	remotewebdriverurl	TC_ID	id
CAPABILITIES	capabilities	TEST_CASE_NAME	name
APPUIUMPATH	appiumpath	CONTROL_FLOW	controlFlow
ANALYTICS	analytics	OR	or
TCM-ORM		ELEMENT_REF	elementRef
PAGE	page	ACTION	action
ELEMENT_NAME	elementname	TEST_DATA	testdata
DYNAMIC_PROPERTY	dynamicproperty	EXPECTED_RESULT	expectedResult
HTMLID	htmlid	SERVICE_SOURCE	servicesource
XPATH	xpath	SERVICE_CONFIG	serviceconfig
CSS_SELECTOR	cssselector	FLAGS	flags
NAME	name	PRIORITY	priority
CLASSNAME	classname	REMARKS	remarks
LINKTEXT	linktext		
TAGNAME	tagname		

7.2.3.1 TCM-Config Sample

```

firefox:
  description: Desktop FireFox
  channel: Desktop
  browser: firefox

CMEiPhone:
  description: iPhone 6 plus device
  channel: Mobile
  browser: Chrome
  name: iPhone 6 Plus
  driver: ChromeMobileEmulatorsuites:
    - release: MultiLingual
  testSuite: MultiLingualVerification
  scenarios: SCENARIOS_MLRU

```

```

{
  "iOSSafari": {
    "description": "Safari on iOS device",
    "channel": "Mobile",
    "browser": "Safari",
    "name": "sraprism iPhone",
    "driver": "iOS",
    "verion": "10.2.1",
    "platform": "iOS",
    "udid": "87649b4f0db8728ee96bbaa646",
    "bundleid": "com.sapientnitro.in.mcd.Arch",
    "apppath": "/Users//SafariLauncher.app",
    "protocol": "http",
    "server": "0.0.0.0",
    "port": 4723,
    "context": "/wd/hub"
  }
}

```

7.2.3.2 TCM-Summary Sample

```

suites:
  - release: MultiLingual
  testSuite: MultiLingualVerification
  scenarios: SCENARIOS_MLRU
  description: Verifying the British Airways header and footer content for different language
  run: Y
  config: CONFIG_Chrome
  batch: A01
  suitelables: search

  - release: MultiLingual
  testSuite: MultiLingualVerification
  scenarios: SCENARIOS_MLCH
  description: Verifying the British Airways header and footer content for different language
  run: Y
  config: CONFIG_FF

```

```
{
"suites": [
    {"release": "Functional",
     "testSuite": "AmazonMOB",
     "scenarios": "SCENARIOS_AMZMOB",
     "description": "Amazon Add to Cart on Mobile",
     "run": "Y",
     "config": "CONFIG_Samsung",
     "suitelables": "regression"}
]
}
```

7.2.3.3 Scenario Sample

```
scenarios:
- id: MLRU
  name: BA English Label Verification
  description: Verify Header and Footer Labels for India using English language
  testcases:
    - BA_Launch
    #- BA_VerifyFooter
    #- BA_VerifyHeader
    - BA_Quit
  scenarioVariables: LANG=RU
- id: AMZMOB
  name: Amazon Add to Cart Mobile
  description: Verify user able to add item to cart and verify the Item in the cart and shipping message
  testcases: [AMZ_Launch,AMZ_SearchPrd,AMZ_SelectPrd, AMZ_AddToCartMob ,AMZ_Quit]
  scenariovariables: CHNL=MOB
```

```
{
  "scenarios": [
    {
      "id": "AMZDTP",
      "name": "Amazon Add to Cart Desktop",
      "description": " verify the Item in the cart and shipping message",
      "testcases": [
        "AMZ_Launch", "AMZ_SearchPrd",
        "AMZ_SelectPrd", "AMZ_SwitchWnd",
        "AMZ_VerifyPrdDetail", "AMZ_AddToCart",
        "AMZ_VerifyCart", "AMZ_Quit" ],
      "scenariovariables": "CHNL=DTP"
    }
  ]
}
```

7.2.3.4 Test Case Sample

```
testcases:  
- id: Launch  
  name: LaunchHomePage  
  teststeps:  
    - or: Custom  
      action: LaunchSite  
      testdata: $AMZ.URL.FIELD1  
      description: Launch Amazon.in  
    - or: Custom  
      action: VALIDATETITLE  
      expectedResult: $AMZ.Title.<<CHNL>>  
      description: Validate Title of the home page  
- id: Quit  
  name: Quit  
  teststeps:  
    - or: Custom  
      action: Quit  
      description: Quit the browser
```

```
{  
  "testcases": [  
    {  
      "id": "Launch",  
      "name": "LaunchHomePage",  
      "teststeps": [  
        {  
          "or": "Custom",  
          "action": "LaunchSite",  
          "testdata": "$AMZ.URL.FIELD1",  
          "description": "Launch Amazon.in"},  
  
        {  
          "or": "Custom",  
          "action": "VALIDATETITLE",  
          "expectedResult": "$AMZ.Title.<<CHNL>>",  
          "description": "Validate Title of the home page"}  
      ]  
    },  
    {  
      "id": "Quit",  
      "name": "Quit",  
      "teststeps": [  
        {  
          "or": "Custom",  
          "action": "Quit",  
          "description": "Quit the browser"}  
      ]  
    }  
  ]  
}
```

7.2.3.5 ORM Sample

SignIn: TXTEmail: xpath: //*[@id='ap_email'] htmlid: app_email name: appEmail TXTPassword: xpath: //*[@id='ap_password'] BTNLogin: xpath: //*[@id='signInSubmit'] Home: TXTSearchBox: xpath: //*[@id='twotabsearchtextbox'] BTNSearchGo: xpath: div/input ProductList: IMGResult1: htmlid: result_1	{ "Home": { "TXTSearchBox": { "name": "search" "tag": "div" }, "BTNSearchGo": { "xpath": "./div/input" } }, "ProductList": { "IMGResult1": { "htmlid": "result_1" }, "LNKProduct": { "linktext": "Men's Wool Jacket" } }
---	---

7.2.4 TDM EXPORT

Runtime TDM can be exported after execution ends, this features is primarily helpful when user wants to create 100s of Orders and capture the Order Numbers, or update the user details and update back the new details to TDM so that the next execution doesn't fail while using the same TDM etc.,

To enable this feature user needs to set `OverwriteTestDataFileWithRuntimeDataFile =Yes` and provide the path at `RunTimeTestDataExportFolderPath`, in case path is left blank the TDM is overwritten.

- If `OverwriteTestDataFileWithRuntimeDataFile=YES` means it 'll change the TDM values and store the same in `../DataFiles/TDM.xlsx`. Incase TDM file is already open then, the changes are made in `RunTimeTestDataExportFolderPath`
- If `OverwriteTestDataFileWithRuntimeDataFile=NO` means, changes are made to the file mentioned in `RunTimeTestDataExportFolderPath`

KEY	FIELD1	FIELD2	FIELD3	FIELD4	ER1	ER2
Title					Amazon.in: Wool Jacket	
JRL	https://www.amazon.com					Amazon.com: Leather Jacket: Clothing, Shoes & Jewelry
TITLE		AMAZON	iiio90	hjdljsd		
Search			Mobiles			iPhone6

User can use `SETTDMREF` and `SETTDMREFVALUE` actions to update the TDM values runtime

Note: While enabling this feature `ReadEmptyTDMRef=Yes` must be set to Yes, otherwise blank rows will not be read from TDM.

7.2.5 Empty TDM Cell Reading

While to enable dynamic TDM reference and TDM export CATS framework reads all the cells from the TDM including cells left blank/empty, if the first column and first row in the TDM is not maintained properly then the chances are more you leave TDM with more blank cell. Have more empty cells in the TDM will delay the execution start every time.

In case you don't want to read blank cells please change the `ReadEmptyTDMRef=No`. If you want to read selective blank cells but not all then you have to enter the cell value like `-"` instead of leaving it blank.

7.2.6 Creative Default Pixel Variation

While user is performing creative testing, project may have acceptable pixel variation like 2px or 5px etc.,

User has to define the pixel value that needs to be ignored while performing verification, for example if 2px is given as `DefaultDeviation=2` and expected element width is 10px and actual width is between 8 and 12 still considered to be valid and the verification will PASS

7.3 CATSReport

7.3.1 Executive Report

The Executive reports now contain stats about test scenarios and by default, stats based on teststeps. However the Teststeps part can be configured to be Testcase stats by changing a property value in `CATSReport.properties` file as follows.

`EXECUTIVE_REPORT_BASED_ON=TESTSTEP` //this is the default value.

`EXECUTIVE_REPORT_BASED_ON=TESTCASE` //you can change it to TestCase

Started at: Fri Mar 31 19:29:56 IST 2017
Ended at: Fri Mar 31 19:30:14 IST 2017
Total Time Taken: 0 min, 15 sec

SUITE DETAILS				SCENARIOS				TEST STEP				COMMENTS	
NAME	RELEASE	CHANNEL	BROWSER	PASSED	FAILED	SKIPPED	TOTAL	PASSED (%)	FAILED (%)	SKIPPED	TOTAL	EXECUTION %	
DRIVERTEST	DEMO	Desktop	Chrome	0 [0 %]	1 [100 %]	0 [0 %]	1	4 [80 %]	1 [20 %]	0	5	100 %	

CURRENT RUN REPORT					CONSOLIDATED RUN REPORT				
PASSED	FAILED	SKIPPED	PARTIALLY SKIPPED	NO RUN	PASSED	FAILED	SKIPPED	PARTIALLY SKIPPED	NO RUN
0 [0.0 %]	1 [100.0 %]	0 [0.0 %]	0 [0.0 %]	0 [0.0 %]	0 [0.0 %]	1 [33.333336 %]	0 [0.0 %]	0 [0.0 %]	2 [66.666667 %]

Failed!!! TESTSTEP Pass Rate : 80.0 % TESTSTEP Execution Rate : 100.0 %

Overall pass percentage and execution percentage can be configured. By default the percentage is based on the scenarios, and it can be modified to Test Case or Test Step

`EXECUTIVE_OVERALL_PASS_PERCENTAGE_BASED_ON= TESTCASE`
`EXECUTIVE_OVERALL_PASS_PERCENTAGE_BASED_ON= TESTSTEP`

7.3.2 Executive Report

The detailed reports too have been changed in their look and feel. Stats regarding the suite start time, suite end time, time difference are included. Scenario description is also included in each scenario header. Test case column headers are only introduced only per scenario to avoid unnecessary duplication of headers.

Similar to Executive Reports, Detailed reports too will now have stats based on Test Steps which is again configurable in the same way in the CATSReport.properties

DETAILED_REPORT_BASED_ON=TESTSTEP //this is by default

DETAILED_REPORT_BASED_ON=TESTCASE // can be configured to reflect test case status.

7.3.3 Report Header and Threshold

You can update the project name in the executive summary report by configuring the value against PROJECT_NAME={{Your Project Name}}

RAG (Red Green Amber) threshold can be configured using below two variables, the percentage fall between Pass and Fail is treated as Amber. The pass threshold is also used to decide the Jenkins job status.

PASS_PERCENTAGE_THRESHOLD=95

FAIL_PERCENTAGE_THRESHOLD=80

7.3.4 Manual Mapping Report

Manual mapping spread sheet will be updated with Current Run Status and Consolidated Run Status. Consolidated Run Status will reflect the status of all the test cases irrespective of whether executed in current run or not. But it will have the recent last execution status.

7.3.5 XML Report

User can generate XML report, incase users want to create their own report or to integrate with any other test management or to process the execution data.

CATS		Your Project Name						Started at: Mon Apr 03 17:45:10 IST 2017 Ended at: Mon Apr 03 17:54:26 IST 2017 Total Time Taken: 0 Hours 9 Mins 15 Secs		
EXECUTION SUMMARY REPORT FOR ProductionMonitoring Desktop Chrome										
SCENARIO		SCENARIO NAME			TOTAL	PASSED [%]	FAILED [%]	SKIPPED	EXECUTION %	
SCENARIOS_24	PM				19	5 [26%]	14 [74%]	0	100%	
SCENARIOS_25	DriverTest				16	10 [63%]	6 [38%]	0	100%	
SCENARIOS_24 Verify that Order confirmation page is opened when click on Place order in Order review Page and verify the Order number and amount										
CONDITION	DESCRIPTION	ACTION	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS	ERROR CODE	OR REFERENCE	DUR	SCREEN LINK
TEST CASE : HOMEPAGE_1 HOMEPAGE_LAUNCHHOMEPAGE										
1	CATS_SUITE_NAME	CATS_SCENARIO_ID	CATS_TC_ID	MANUAL_ID	CURRENT_RUN_STATUS	CONSOLIDATED_RUN_STATUS	EXECUTION_DATE			
2	DRIVERTEST	SCENARIOS_26	TEST_LOCAL	CATS-10	SKIPPED	SKIPPED	03/04/2017 05:54:31	F		
3						NORUN	30/03/2017 10:33:23			
4						NORUN	30/03/2017 10:33:23			

To generate xml report user needs to set the *GENERATE_XML_REPORT* to Yes in CATSReport.properties file. User can find the XML report inside the XML folder present in the OUTPUT folder.

7.3.6 Take Screenshots

User can set TAKE_SCREENSHOT_ALWAYS to YES if user wanted to capture screen shot for every other step. Make sure that you do not enable this feature unless otherwise required. As the size of the overall images will be too high.

7.4 CATSServiceTestConfig

7.5 CATS License

Users can make the license request from the CATS UI itself. There are 4 kind of license are generated, user need to navigate to License tab in CATS UI and select an appropriate license required.

1. SR License for SRF employees
2. Trial License for external people
3. Permanent License for external people
4. Server License for CI/CD environments

7.5.1 SR License

Sapient Razorfish employees / anybody who has access to DOJO JIRA can create the license without anybody's help. Provide the JIRA User ID (which you use to login to JIRA which is not necessarily your NT ID) and Password then Select the project your are generating the ID and Click on the Generate license. If you are using LL ID then your JIRA user ID will be likely

xxxx@publicisgroupe.net

One person can have maximum of 3 active licenses (for three different machines), beyond that license will not be generated, in order to generate for new machine you have to deactivate the license of other machine(s).

7.5.2 Trial License

Trial license is generated for people who want is outside SRF and wanted to explore CATS. License will be generated for 15 days without any restriction

Once license is expired, user cannot make another trial license request for the same machine. However user can continue to execute up to two test cases.

7.5.3 Permanent License

Permanent license is for clients and external people. To make permanent license request user needs to provide Name, Email and Company Name and then Sapient Razorfish contact person name and JIRA ID of that person (work with project counterpart to know this information, who can approve your license) and Submit the request.

Upon submission of the ticket user will get the ticket number in the below console like "CATS-xxxx" which needs to be noted for future activation

Once request is submitted, an email will be triggered to approver and approver needs to open the ticket and add the comment "APPROVED" or "REJECTED" based on the comment either license will be generated or rejected.

User can check Approval or Rejection status, using the reference key “CATS-xxxx”, navigate to License tab > External > Generate License and use the key and click on Activate License. Based on the approval status it will either generate the license or it will give you the actual status.

License generated will be valid for 10 years

7.5.4 Renewal of License

Navigate to License tab > External > Renewal and use the key “CATS-xxxx” and click on Request for Renewal. Once request is submitted, an email will be triggered to approver and approver needs to open the ticket and add the comment “APPROVED” or “REJECTED” based on the comment either license will be generated or rejected.

7.5.5 Server License

TBU

7.5.6 CI/CD Server License

User has to create a license request through **CATS Service Portal** and license type should be **“CI/CD Integration”**. Make a note of your ticket number and update **CATSLicenseConfig.properties** with below three information.

1. CI_CD_LIC_ID = <Should be the **JIRA ticket id** Ex. CATS-xxxx>
2. CI_CD_LIC_OWNER = JIRA ID of the user who created the ticket
3. CI_CD_PROJECT_NAME=Project name given during ticket creation

7.6 BrowserConfiguration

7.6.1 Browser Capabilities

CATS supports IE, Firefox and Chrome. The capabilities and preferences used by these browsers are stored in a properties file called **BrowserConfiguration.properties** in CATS/Core folder.

```
BrowserConfiguration.properties - Notepad
File Edit Format View Help
IE.CAPABILITY.ignoreZoomSetting(boolean, true)
IE.CAPABILITY.ignoreProtectedModeSettings(boolean, true)
IE.CAPABILITY.nativeEvents(boolean, false)
IE.CAPABILITY.ie.ensureCleanSession(boolean, false)

#####
# Firefox
#####
Firefox.CAPABILITY.acceptSSLcerts(boolean, true)
Firefox.PREFERENCE.network.proxy.type(int, 5)
Firefox.PREFERENCE.network.proxy.http(string, localhost)
Firefox.PREFERENCE.network.proxy.http_port(int, 8080)
Firefox.PREFERENCE.network.cookie.cookieBehavior(int, 0)
Firefox.PREFERENCE.browser.cache.memory.enable(boolean, true)
```

One can change the capabilities and preferences listed to their liking and also add extra in the same format described in the file.

Once you make the changes and save the file. Suppose if changes are made to the preferences and capabilities and preferences of Firefox.

Then every instance of Firefox called in the TCM now will be spawned with the described capabilities and preferences. In the figure shown above, both suite1 and suite3 will be launched with the capabilities and preferences set in the properties file.

7.6.2 Firefox User Agent

A User agent is basically a string which identifies what kind of a device is requesting access to a site. Based on it, the appropriate view is rendered to the user on the browser.

CATS 4.0 also supports passing of user agent strings to render mobile and tablet views in the normal desktop browsers. This feature is currently supported only for Firefox.

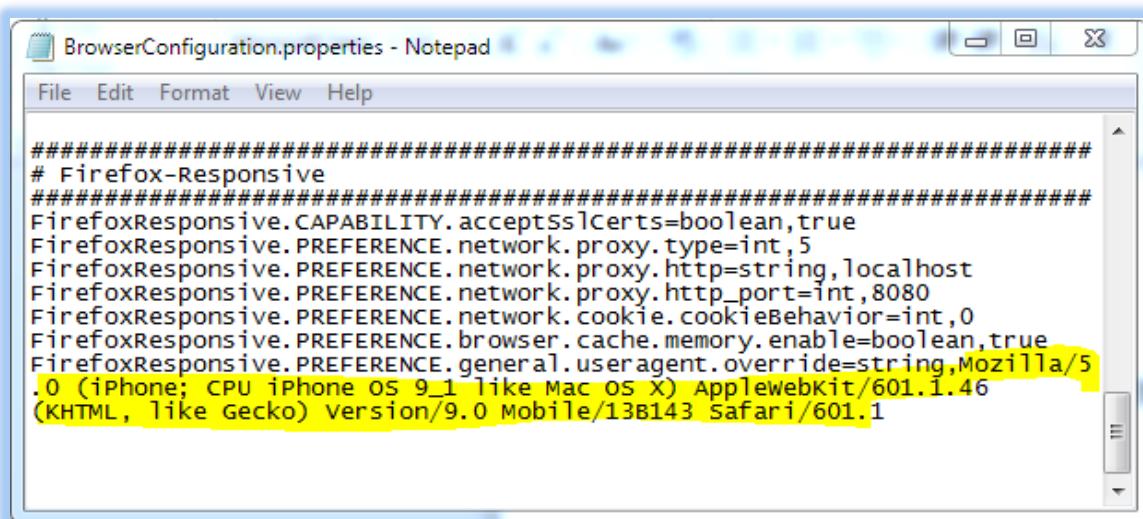
To use this feature, first you have to get a user-agent string for the device you want to change your site's view to. For ex.

"Mozilla/5.0 (iPhone; CPU iPhone OS 9_1 like Mac OS X) AppleWebKit/601.1.46 (KHTML, like Gecko) Version/9.0 Mobile/13B143 Safari/601.1"

is the user agent string for an iPhone. Similarly you can get user-agent strings for your desired device by a simple lookup on the web.

Once you get hold of the user agent string, open the `BrowserConfiguration.properties` file and in the **# Firefox-Responsive** Section, change the property value for

`FirefoxResponsive.PREFERENCE.general.useragent.override`



The highlighted part is where you have to paste the user agent string. Save the changes to the file.

RELEASE	TEST SUITE	TEST CASES - SCENARIOS	CHANNEL	BROWSERS	ITERATIONS	RUN	CONFIG
4	Suite1	Script_1,Script_2,Script_3	Desktop	Firefox-Responsive	1	Y	
4	Suite2	Script_4,Script_5	Desktop	Chrome	1	N	
4	Suite3	SCENARIOS_1	Desktop	Firefox	1	N	

In the TCM Summary Sheet, in the BROWSERS column pass the value of the browser as "Firefox-Responsive" as shown in the figure above

Now when you run CATS, Firefox will be launched and the view will be the as per the user agent string provided.

7.6.3

Browser Proxy

CATS can initialize the driver along with proxy, to enable proxy user need to pass protocol, server and port details. To add more than one protocol to browser user needs to pass semicolon separated values in each of the column. If user wants to add different protocols to proxy with same server and port details then it can be achieved without duplicate entries (refer fourth entry in below snapshot)

PROTOCOL	SERVER	PORT
http	198.121.11.1	8080
ftp	myhost.com	8011
http;ssl	localhost;test.com	90;8090
http;ssl;ftp	localhost	9090

8 CATS Execution Modes

Once you have the test suite (the set of TCM, TDM and OR files) are prepared and validated, it is time for executing the test cases. When the test suite is executed the input files are validated and test execution engine starts, browsers get opened and test scripts starts running.

To start the execution, just invoke the batch file core\Start_Cats.bat or use the UI starting LaunchCATSUI.bat which can be found inside the core folder of CATS

Here is the list of parameters that affects the execution

Properties File	Parameter	Description
CATSConfig	GRID_HUB_URL	The node in which Selenium server is run. The url of the Selenium server. This is required for run the test cases in parallel
	CATS_INPUT_JSON_PATH	The default value of the property is D:\\CATS\\input\\Inputs.json In this path where the json file is created. You can change this when you change the folder.
	HTML_REPORT_PATH	The default value of this property is D:\\CATS\\Core\\output\\reports\\ In this path the html reports are placed. You can change this when the default folder is changed.

After the execution is done, there are TWO possibilities

Execution is passed: When the execution is passed, you will see the execution summary report in the folder D:\\CATS\\core\\output\\reports (configured in HTML_REPORT_PATH). The result of each action of the test case given with the details on Description, “Test data”, “Expected Result”, “actual result”, and the result.

Execution is failed: When the inputs are given wrong, the execution will not start. The errors are listed in an excel sheet and kept in the folder D:\\CATS\\input\\Excel2JsonToolOutputxxxx.xlsx. The XXXX are replaced with the time of running in this format (YYYY-MM-DD HH-MI-SS), so that the errors list of the previous runs are still maintained.

The errors list will have enough context and the references so that fixing the errors is easy. Given below is the sample from the errors list excel file.

Sl No	Error	Reference
1	Sheet: SUMMARY, Row: 2 - Test Suite name is missing	Execution:
2	Missing Title for (LaunchHomePage)	TestCase
3	Missing Title for (Quit)	TestCase
4	Missing Title for (LaunchHomePage)	TestCase
5	Missing Title for (SelectProduct)	TestCase
6	Missing Title for (Quit)	TestCase
7	Missing Priority for (10)	Scenarios
8	Missing Title for (LaunchHomePage)	TestCase
9	Missing Title for (LaunchHomePage)	TestCase
10	Missing Title for (SelectProduct)	TestCase
11	Missing Title for (Quit)	TestCase

8.1 Parallel Execution

The parallel execution facilitates the user to execute the tests in parallel. Each suite will run in its own browser window. This will save a lot of time executing the test suites. This will be very good feature when you have lot of test suites to run.

How to run the test suites in parallel?

Instead of running the batch file Start_Cats.bat, invoke the batch file start_cats_multithread.bat.

8.2 Command Line Execution

Not all the times you want to trigger the execution from CATS UI, at times you may decide to run from the command prompt directly or call the batch file through jenkins.

CATS framework lets you pass the below named arguments.

Argument Name	Description	Usage
env	Below 5.0.3 - Allows to pick the environment from the ENV sheet Above 5.0.3 – Sets the reserved variable <>CURR_ENV>> value	Start_cats.bat –env QA
batches	Executes the selected batches	Batch Execution , Batch Sequence Execution
configPath	Overrides the configuration folder location	-configPath config_2
mode	0 to run CATS from start and 1 to run only the execution	Being not used
inputPath	Overrides the CATSToolsInput.properties	-inputPath d:/cats/test.properties
suitelabels	Suites matching the labels will be executed	-suitelabels “sanity&sprint1”
Scenariolabels	Scenario matching the labels will be executed	-scenariolabels “checkout”

mode	This argument is used to trigger the bdd execution or to perform bdd dry run to identify the missing step definitions	-mode bdd or -mode bdddryrun
tags	This argument is used to accept the tag expression which needs to be passed to the cucumber runner	-tags "@desktop and not @sprint4"

Many of times the scripts to test may remain the same the only thing the changes is the environment like it may be on **SIT, UAT or IST** etc.

KEY	SIT	UAT	RLQA2	IST
Homepage			https://www-rlqa2.sapient.com/ralphlauren/en/ralphlauren/en	https://www-rlqa2.sapient.com/ist1.ralphlauren.com/ralphlauren/en/StaticContent/ContentLeft/NavPage
BackOffice			https://www-sit.ralphlauren.com/backoffice	https://www-rlqa2.sapient.com/ist1.ralphlauren.com/backoffice
BackOfficeUser	admin	admin		admin
BackOfficePWD				
HPC	https://admin-SIT.ralphlauren.com	https://admin-rlqa2.sapient.com	https://admin-rlqa2.sapient.com	https://admin-rlqa2.sapient.com
HPCUSER	admin	admin	admin	admin
HPCPWD				
Wait	5	10	15	25
11				
12				
13				
14				
15				
16				

Keeping this in consideration, you can now specify the environment variable in your CATS command line. i.e. start_cats.bat

First and foremost, one has to keep their environment data organized in a sheet name called ENV in TDM sheet as shown in the figure above earlier.

Once you have arranged your data and suppose your test cases are written specifically to adhere to a particular environment, say like SIT in the sheet shown in figure below.

A	B	C	D	E	F	G	H
TC ID	TEST CASE NAME OR	PAGE	ELEMENT	ACTION	TEST DATA	EXPECTED RESULT	
1	Modular Test Case	Custom		LAUNCHSITE	\$ENV.HOME PAGE.SIT		
		HomePage	Home	TXTBXSearch	\$ENV.BACKOFFICE.SIT		
		Custom		VERIFYELEMENTPRESENT	{}{{variable}}		
		Custom		SETVARIABLEVALUE		\$ENV.BACKOFFICE.SIT	
		Custom		QUIT			
		Custom		LAUNCHSITE	\$HomePage.URLQA.FIELD1		
		Custom		QUIT			

And if you want to run the same test cases for a different environment say UAT, all you have to do is, when you call start_cats.bat from wherever you do whether from command line or a Jenkins job , you have to pass arguments as shown in figure below.

```
D:\CATS LATEST\CATS4.0\core>start_cats -env UAT
```

Figure 4.3

This tells CATS to override all test data and expected result columns if they begin with ENV to the value UAT. I.e. The test data of the first step LAUNCHSITE is \$ENV.HOMEPAGE.SIT, so now it will become \$ENV.HOMEPAGE.UAT.

This ensures that you are now using the same test case for different environment data values.

IS_TEST				
Modular_1		Test Case Name:	ModularTest	
Action	Test Data	Expected Result	Actual Result	Result
LAUNCHSITE	https://www-uat.ralphlauren.com/ralphlauren/en		Site launched success	PASSED
VERIFYELEMENTPRESENT	https://www-uat.ralphlauren.com/backoffice		Element doesn't exist	FAILED
SETVARIABLEVALUE	[[variable]]	https://www-uat.ralphlauren.com/backoffice	Variable "[[variable]]" is set to "https://www-uat.ralphlauren.com/backoffice"	PASSED
QUIT				PASSED
LAUNCHSITE	https://www-rqa2.sapient.com/ralphlauren/en/			NOT STARTED
QUIT				NOT STARTED

This is the report after overriding the test case in figure 4.2 with UAT values. As you can see all the test data and expected result values are replaced with UAT values.

8.3 Labeled Execution / Target Regression

Label feature is introduced in to enable to the targeted regression. More than one label can be given to a suite or scenario using semicolon separated values.

When user executing through CATS UI, they can provide suite labels and scenario labels in *SUITE_LABELS*, *SCENARIO_LABELS* of *CATSCfg.properties* file respectively . If execution is performed through command prompt, user can provide suite labels using the syntax -suitelabels and scenario labels using -scenariolabels. For further clarification, refer the screenshots below:-

RELEASE	TEST SUITE	RUN	CONFIG	BATCH	OWNER	SUITE VARIABLES	LABELS
R6.0	LoopTesting	Y	CONFIG_Chrome				a
R6.0	LoopTesting1	Y	CONFIG_Chrome				a;b;c
R6.0	LoopTesting2	Y	CONFIG_Chrome				bce;cfg

ID	SCENARIO NAME	SCENARIO DESCRIPTION	TEST CASES	TEST DATA KEY	SCENARIO VARIABLES	LABELS
1	New Scenario to test Loop Implementation	Loops	Loop_1			s;b
2	MyNew Test		TEST_1=1;TEST_2			i;j;k;g
3	My Another Test		TEST_2			asc

```
CATS\core>start_cats.bat -suitelabels a,b
```

Here only those suites will execute which have either *a* or *b* or both values present in label column of Summary Sheet

```
CATS\core>start_cats.bat -scenariolabels s,asc
```

Here only those scenarios will execute which have either *s* or *asc* or both values present in label column of SCENARIOS Sheet

```
CATS\core>start_cats.bat -suitelabels a,"a&b" -scenariolabels s,asc
```

Here only those suites will execute which have either *a* or both *a* and *b* value present in label column of Summary Sheet and only those scenarios which have either *s* or *asc* or both values present in label column of SCENARIOS Sheet. **Remember quotes are necessary when & symbol is used in command prompt.** It is not required when user run through CATS UI.

8.4 Batch Execution

Suites now can be tagged and run in batches with the latest Batch Execution feature in CATS 4.0

RELEASE	TEST SUITE	TEST CASES - SCENARIOS	CHANNEL	BROWSER	ITEI	RUN	BATCH
REL3	PreExplicitPro	SCENARIOS_SetNgetXdollaroff	Desktop	Firefox	1	N	
REL3	PreNavigation	SCENARIOS_EmptyCart	Desktop	Firefox	1	N	A01_SEQ
REL3	PreVerifyFindi	SCENARIOS_PreVerifyFindinSt	Desktop	Firefox	1	N	A01_SEQ
REL3	PreEXplicitPro	SCENARIOS_SetNgetXPerOnN	Desktop	Firefox	1	N	B01
REL3	PreShippingRestrictionForDelivery	SCENARIOS_PreVerifyShippingRestrictionForDelawareState	Desktop	Firefox	1	N	B01
REL3	PreMultiShip	SCENARIOS_MultiShipngItem	Desktop	Firefox	1	N	C01
REL3	PreShippingConditions	SCENARIOS_MultipleShipping	Desktop	Firefox	1	N	C01
REL3	PreVerifyGiftCardBalanceInPounds	SCENARIOS_VerifyGiftCardDetails	Desktop	Firefox	1	N	

Sometimes, the suites will be interdependent and in that case you would want to run it sequentially in order for them to run correctly. But you would also want to run the other suites to run in parallel in order to reduce execution time. This can be achieved using the batch execution feature.

As seen in the figure above, in the summary sheet of the TCM. In the column BATCH (you can create one if it doesn't exist), you can specify the batch name for a suite. The suites with the different batch names will never be executed simultaneously (even if you are running it in parallel). This can be used to efficiently run the test suites that will resolve the interdependency issues and also run in parallel.

The batch execution feature can only be exploited with the CATS parallel run. In the sequential run, the only benefit is that you can change which suites to be run by just changing the batch name in the command line. I.e. instead of going to TCM and changing “RUN” value from Y to N or vice versa.

```
Administrator: C:\windows\system32\cmd.exe
D:\CATS\core>start_cats.bat -batches A01
```

You can select the batch to be run, in that case it will run the suites with selected batch name (the RUN value still have to Y though)

```
Administrator: C:\windows\system32\cmd.exe
D:\CATS\core>start_cats.bat -batches A01,B01
```

To run multiple batches you can pass comma separated values like in the figure shown above.

To run them in parallel you can call start_cats_multithread.bat along with the batch names

```
Administrator: C:\windows\system32\cmd.exe
D:\CATS\core>start_cats_multithread.bat -batches A01,B01
```

8.5 Batch Sequence Execution

If you want the suites in a batch to run sequentially (because the suites are interdependent or whatever might be the reason) you can configure the suites to run in sequential just by appending “_SEQ” to the batch name.

RELEASE	TEST SUITE	TEST CASES - SCENARIO	CHANNEL	BROWSER	ITEI	RUN	BATCH
REL3	Suite1	SCENARIOS_SetNgetXdollar offNByMinHMC	Desktop	Firefox	1	Y	
REL3	Suite2	SCENARIOS_EmptyCart	Desktop	Firefox	1	Y	A01
REL3	Suite3	SCENARIOS_PreVerifyFindin StoreforPDP	Desktop	Firefox	1	Y	A01
REL3	Suite4	SCENARIOS_SetNgetXPerOn NByMinHMC	Desktop	Firefox	1	Y	B01_SEQ
REL3	Suite5	SCENARIOS_PreVerifyShippi ngRestrictionForDelawareSt ateAddress	Desktop	Firefox	1	Y	B01_SEQ
REL3	Suite6	SCENARIOS_MultiShipngItmAddrRestriction	Desktop	Firefox	1	Y	C01
REL3	Suite7	SCENARIOS_MultipleShippin gAddress	Desktop	Firefox	1	Y	C01
REL3	Suite8	SCENARIOS_VerifyGiftCardD etails	Desktop	Firefox	1	Y	

```

Administrator: C:\windows\system32\cmd.exe
D:\CATS\core>start_cats_multithread.bat -batches A01,B01_SEQ

```

The suites with batch name B01_SEQ will run in sequential manner among themselves.

Let us see some case examples to understand the feature better

RELEASE	TEST SUITE	TEST CASES - SCENARIO	CHANNEL	BROWSER	ITE	RUN	BATCH
REL3	Suite1	SCENARIOS_SetNgetXdollar offNByMinHMC	Desktop	Firefox	1	Y	
REL3	Suite2	SCENARIOS_EmptyCart	Desktop	Firefox	1	Y	A01
REL3	Suite3	SCENARIOS_PreVerifyFindin StoreforPDP	Desktop	Firefox	1	Y	A01
REL3	Suite4	SCENARIOS_SetNgetXPerOn NByMinHMC	Desktop	Firefox	1	Y	B01
REL3	Suite5	SCENARIOS_PreVerifyShippi ngRestrictionForDelawareSt ateAddress	Desktop	Firefox	1	Y	B01
REL3	Suite6	SCENARIOS_MultiShipngite mAddrRestriction	Desktop	Firefox	1	Y	C01
REL3	Suite7	SCENARIOS_MultipleShippin gAddress	Desktop	Firefox	1	Y	C01
REL3	Suite8	SCENARIOS_VerifyGiftCardD etails	Desktop	Firefox	1	Y	

When in parallel run, Suite2 and Suite3 are clubbed into batch A01, Suite4 and Suite5 in B01 batch, Suite6 and Suite7 in C01 batch. Now only one batch will start for execution first, let us it is A01: A01 has two suites suite2 and suite3 which will run in parallel to completion. After both complete, the batch A01 is done. Now batch B01 starts, here suite4 and suite5 will run in parallel to completion and then similarly batch C01 also get executed in the same fashion.

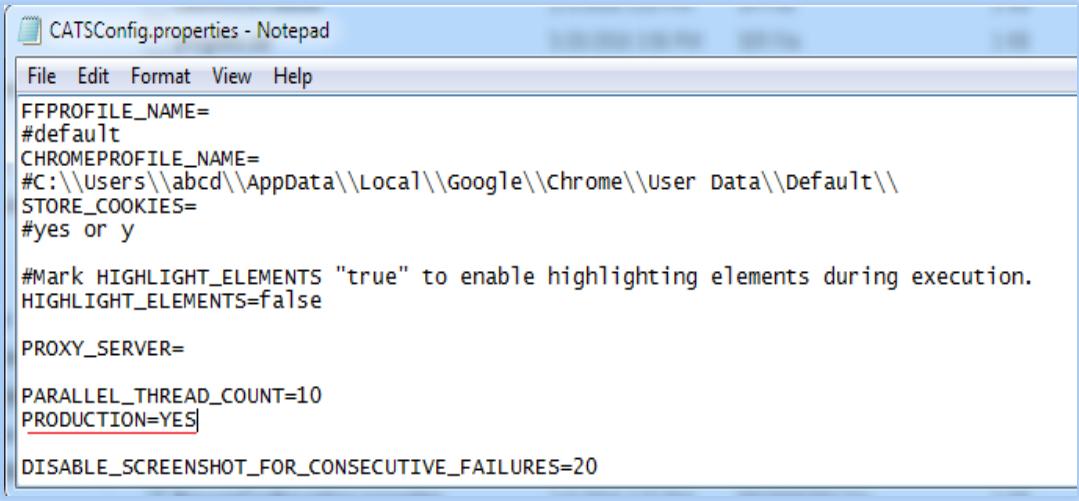
Now let us see “_SEQ” part

RELEASE	TEST SUITE	TEST CASES - SCENARIO	CHANNEL	BROWSER	ITE	RUN	BATCH
REL3	Suite1	SCENARIOS_SetNgetXdollar offNByMinHMC	Desktop	Firefox	1	Y	
REL3	Suite2	SCENARIOS_EmptyCart	Desktop	Firefox	1	Y	A01
REL3	Suite3	SCENARIOS_PreVerifyFindin StoreforPDP	Desktop	Firefox	1	Y	A01
REL3	Suite4	SCENARIOS_SetNgetXPerOn NByMinHMC	Desktop	Firefox	1	Y	B01_SEQ
REL3	Suite5	SCENARIOS_PreVerifyShippi ngRestrictionForDelawareSt ateAddress	Desktop	Firefox	1	Y	B01_SEQ
REL3	Suite6	SCENARIOS_MultiShipngite mAddrRestriction	Desktop	Firefox	1	Y	C01
REL3	Suite7	SCENARIOS_MultipleShippin gAddress	Desktop	Firefox	1	Y	C01
REL3	Suite8	SCENARIOS_VerifyGiftCardD etails	Desktop	Firefox	1	Y	

When this above suite is run in parallel, A01 batch gets executed and when B01 batch starts, suite4 and suite5 in the B01 batch won't run in parallel to completion because we have appended "_SEQ" which will instruct them to run sequentially (i.e. one by one).

8.6 Production Monitoring Execution

It is a simple feature in CATS that lets you track the failure of important steps in a test case. An additional report will be generated along with the normal reports generated. To use the production monitoring feature and let the additional reports to get generated, it must be made sure that the value of PRODUCTION in CATSConfig.properties (in the CATS/Core folder) is set to YES.



```
CATSConfig.properties - Notepad
File Edit Format View Help
FFPROFILE_NAME=
#default
CHROMEPROFILE_NAME=
#C:\\\\users\\\\abcd\\\\AppData\\\\Local\\\\Google\\\\Chrome\\\\user Data\\\\Default\\\\
STORE_COOKIES=
#yes or y
#Mark HIGHLIGHT_ELEMENTS "true" to enable highlighting elements during execution.
HIGHLIGHT_ELEMENTS=false
PROXY_SERVER=
PARALLEL_THREAD_COUNT=10
PRODUCTION=YES
DISABLE_SCREENSHOT_FOR_CONSECUTIVE_FAILURES=20
```

Once the reports are enabled, you can mark any test step in your test case to be included for production monitoring reports. This can be done by marking the test step as “REPORT” in the flags column as shown in the figure below.

TC ID	ACTION	EXPECTED RESULT	FLAGS
LaunchHomePage	LAUNCHSITE	\$HomePage.URL.Field1	REPORT
	VERIFYTITLE	wrong title	

Once you mark a step a test step as “REPORT”. The additional reports generated will contain this step **if and only if it fails**.

So the production monitoring reports will contain only the steps that were marked “REPORT” and failed during execution.

The additional reports also get generated in the same default folder as the normal reports get generated. If you go to the CATS/core/output/report folder.

 Images	5/24/2016 2:20 PM	File folder
 ExecutiveSummaryReport.html	5/24/2016 2:21 PM	Firefox HTML Doc... 9 KB
 ProductionExecutiveSummaryReport.html	5/24/2016 2:21 PM	Firefox HTML Doc... 9 KB
 SanitySuite_Execution_Report.html	5/24/2016 2:20 PM	Firefox HTML Doc... 19 KB
 SanitySuite_Production_Execution_Report.html	5/24/2016 2:21 PM	Firefox HTML Doc... 17 KB

You will see extra reports get generated, in them, open ProductionExecutiveSummaryReport.html file. In it you can see your test suites listed the same way as before.

CATS

EXECUTIVE DASHBOARD

Click on the suite to open the detailed report.

DETAILED TEST EXECUTION REPORT								
Suite : SanitySuite								
Scenario	Scenario Name			Total Testcases	Passed	Failed	Skipped	Execution %
SCENARIOS_GENERATED_1	SCENARIOS_GENERATED_1			1	0	1	0	100.0%
Scenario: SCENARIOS GENERATED 1								
Test Case:	HomePage_LaunchHomePage		Test Case Name:	LaunchHomePage				
Action Description	Action	Test Data	Expected Result	Actual Result			Result	
Verifying the Title of the page	VERIFYTITLE		wrong title	Actual page title is : Not Found Ralph Lauren Site US			FAILED	

If you observe the detailed report only contains the VERIFYTITLE step which was marked “REPORT” (i.e. important) and failed!

Make sure to remember enabling the production report generation in the CATSConfig.properties (by setting PRODUCTION=YES), otherwise you will see no reports getting generated in spite of you marking steps as important!

8.7 Sauce Labs Execution

Sauce Labs is a cloud-hosted, web and mobile application automated testing platform. Executing CATS tests on sauce labs is as follows.

Visit <https://saucelabs.com/> and make an account. A free Trial or a paid one according to your convenience. After you login, you are presented with a 'Dashboard'. Under left hand corner open 'My Account'.

Refer below image, copy your username and access key from here. You will need this data to configure it in CATS (i.e. in the 'CONFIG' sheet in TCM workbook).

Here, in the below image 'pragun1993' is the username.

The screenshot shows the SauceLabs dashboard for the user 'pragun1993'. The top navigation bar has a message: 'Manual sessions are limited to 10 minutes for free accounts. This limit will be removed when you subscribe to a monthly plan.' The main area displays the user's name 'pragun1993' and full name 'Pragun Sharma'. It also shows 'CONCURRENCY LIMIT' at 0 VMS and usage statistics. A red box highlights the user name 'pragun1993'.

Scroll down here to see your access key

The screenshot shows the SauceLabs dashboard with the 'Access Key' section highlighted. It features two line charts: 'MINUTE CONSUMPTION BY DAY' and 'TEST RUN BY DAYS'. Below the charts, there is a large redacted area for the access key, with a 'Show' button. The sidebar on the left includes 'Team Management', 'My Account', 'User Settings', 'Support', and 'Sign Out'. The bottom navigation bar shows the user 'Pragun Sharma'.

1. Make the following configuration changes in the TCM.

First in the SUMMARY SHEET,

A	B	C	D	E	F	G	H	J	
RELEASE	TEST SUITE	TEST CASES - SCENARIOS	CHANNEL	BROWSERS	ITERA	RUN	Y	BATC	CONFIG
151 REL3	My Account Mobile Android	SCENARIOS_MyAccountMobile	RemoteWebDriver		1	Y	C01		CONFIG_4
152 REL3	Product Grid Sorting Mobile Android	SCENARIOS_ProductGridSortingMobile	RemoteWebDriver		1	Y	C01		CONFIG_7
153	MobileFilterTestCases Android	SCENARIOS_SearchStepByStepFilter001,SCENARIOS_MobileProductGridRefine045And046,SCENARIOS_ProductGridPriceRefineMobile,SCENARIOS_Search027	RemoteWebDriver		1	Y	C01		CONFIG_7

The Suite (Highlighted yellow) is designated to run on SauceLabs. 'CHANNEL' is 'RemoteWebDriver'. Provide a corresponding Configid in 'CONFIG' column.
Here it is CONFIG_4

In the CONFIG sheet,

ID	NAME	VERSION	DRIVER	PLATFORM	UDID	REMOTEWEBDRIVER_URL	CAPABILITIES
1	GoogleNexus5	5.1	AndroidSimulator	Android			
2	GoogleNexus6	5.1.1	AndroidDevice	Android			
3	iPhone 6		SauceLabsRemoteWebDriver				platformVersion:9.2,\r\ndeviceName:iPhone 6,\r\nplatformName:iOS,\r\nplatform:MAC,\r\nbrowserName:Safari,\r\ndeviceOrientation:portrait
4	Android		SauceLabsRemoteWebDriver			<a href="http://<username>:<access-key>@ondemand.saucelabs.com:80/wd/hub">http://<username>:<access-key>@ondemand.saucelabs.com:80/wd/hub	browserName:android,\r\nplatform:Linux,\r\nversion:4.4,\r\ndeviceName:Samsung Galaxy S4 Emulator,\r\ndeviceOrientation:portrait
5	AndroidAppium		SauceLabsRemoteWebDriver				browserName:Browser,\r\nappiumVersion:1.4.16,\r\ndeviceName:Google Nexus 7

Add columns REMOTEWEBDRIVER_URL and CAPABILITIES to the sheet if not present. Here 'NAME' can be any text. 'DRIVER' field needs to be 'SauceLabsRemoteWebDriver'

Under 'REMOTEWEBDRIVER_URL' copy the user name and access key from step 3 and paste it here. It should look like this

<http://<username>:<access-key>@ondemand.saucelabs.com:80/wd/hub>

'CAPABILITIES' determine the device/emulator/OS version etc. on which SauceLabs is going to run execution in their VM. Visit <https://wiki.saucelabs.com/display/DOCS/Platform+Configurator#/> to get the right set of capabilities depending on your needs. The format under 'CAPABILITIES' reads like this

<attribute>:<attribute-value>,\r\n<next-attribute>:<next-attribute-value>,....

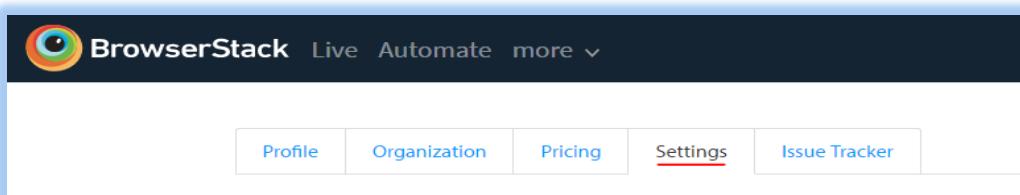
Note: In case of execution on Android 'browserName' should be 'Android' and not 'Browser'. This is for CATS engine internal purpose.

8.8 Browser Stack Execution

Browser Stack is a cloud-based cross-browser testing tool that enables developers to test their websites across various browsers on different operating systems and mobile devices, without requiring users to install virtual machines, devices or emulators.

Visit <https://www.browserstack.com/> and make an account. A free Trial or a paid one according to your convenience. After you login, go to account --> profile → Settings

Refer below image, copy your username and access key from here. You will need this data to configure it in CATS (i.e. in the 'CONFIG' sheet in TCM workbook).



Scroll down here to see your access key

Automate

View your private and unique BrowserStack Automate information and keys.

Username
shishirverma

Access Key
9VaquNMcaE...Phz4Fp

1. Make the following configuration changes in the TCM.

First in the SUMMARY SHEET,

A	B	C	D	E	F	G	H	J	
RELEASE	TEST SUITE	TEST CASES - SCENARIOS	CHANNEL	BROWSERS	ITERA	RUN	L	BATC	CONFIG
151	REL3	My Account Mobile Android	SCENARIOS_MyAccountMobile	RemoteWebDriver	1	Y	C01	CONFIG_4	
152	REL3	Product Grid Sorting Mobile Android	SCENARIOS_ProductGridSortingMobile	RemoteWebDriver	1	Y	C01	CONFIG_7	
153	REL3	MobileFilterTestCases Android	SCENARIOS_SearchStepByStepFilter001,SCENARIOS_MobileProductGridRefine045And046,SCENARIOS_ProductGridPriceRefineMobile,SCENARIOS_Search027	RemoteWebDriver	1	Y	C01	CONFIG_7	

The Suite (Highlighted yellow) is designated to run on BrowserStack. 'CHANNEL' is 'RemoteWebDriver'. Provide a corresponding Configid in 'CONFIG' column. It is CONFIG_4

In the CONFIG sheet,

ID	NAME	VERSION	DRIVER	PLATFORM	BROWSER	REMOTEWEBDRIVER_URL	CAPABILITIES
1	GoogleNexus5	5.1	AndroidSimulator	Android	Chrome		
2	GoogleNexus6	5.1.1	AndroidDevice	Android	Chrome		platformVersion:9.2,\ndeviceName:iPhone6,\nplatformName:iOS,\nplatform:MAC,\nbrowserName:Safari,\ndeviceOrientation:portrait,\nacceptSslCert:true,\njavascriptEnabled:true
	iPhone6Safari	iPhone 6	SauceLabsRemoteWebDriver				
Test	Test						device:HTC One M9,\nplatform:ANDROID,\nbrowserName:android,\nversion:4.4,\nplatformName:Android,\nbrowserstack.debug:true
4			BrowserStackRemoteWebDriver			http://USERNAME:ACCESSKEY@hub.browserstack.com/wd/hub	

Add columns REMOTEWEBDRIVER_URL and CAPABILITIES to the sheet if not present. Here 'NAME' can be any text. 'DRIVER' field needs to be 'BrowserStackRemoteWebDriver'

Under 'REMOTEWEBDRIVER_URL' copy the user name and access key from step 3 and paste it here. It should look like this <https://username:accesskey@hub-cloud.browserstack.com/wd/hub>

'CAPABILITIES' determine the device/emulator/OS version etc. on which BrowserStack is going to run execution in their VM. The format under 'CAPABILITIES' reads like this

<attribute>:<attribute-value>,\r\n<next-attribute>:<next-attribute-value>,....

Valid capabilities can be identified from <https://www.browserstack.com/automate/capabilities>

Note: In case of execution on Android 'browserName' should be 'Android' and not 'Browser'. This is for CATS engine internal purpose.

8.9 Grid Execution

The Grid execution is an extension to the Parallel Execution. Grid setup will facilitate running the tests in parallel across the systems, instead of running it in one system.

This needs setting up the Selenium Hub and Selenium node. The selenium Hub needs to be setup in one machine. This need not be the same machine where the test execution is kicked in.

Setting up the Selenium Hub

Copy the “selenium Server” folder, which is included in the distributable package of CATS. Configure the parameters and start the batch file “Start Selenium Hub.bat”. Here is the list of parameters to configure in the hubconfig.json, which is present in the same folder.

“host” - IP Address of the computer, on which you would like to start Selenium server

“port” - The port on which the selenium server is run.

Setting up the Selenium Node

Copy the “selenium Node” folder, which is included in the distributable package of CATS. Configure the parameters and start the batch file “Start Selenium Node.bat”. Here is the list of parameters to configure in the nodeconfig.json, which is present in the same folder.

hubHost: This is the url of the register service of the Selenium server, Example:

<http://10.209.5.79:4447/grid/register>

hub: IP Address of the Selenium Server computer, on which you would like to start Selenium server

hubport: port of the Selenium Server computer, on which you would like to start Selenium server

port: Port on which the selenium node is run.

In addition to the above parameters you need to configure the capabilities of the each node. This is very useful while testing the cross browser testing.

You can assign multiple capabilities to a particular node. In this way the nodeconfig.json might differ for each node.

Here is the configuration of each capability.

seleniumProtocol - The value is always WebDriver

platform - The value will be Windows on a windows machine

browserName - Browser name (Firefox, IE, Chrome, Safari)

maxInstances - Number of instances of the selected browser can run

version - The browser version

Here is the sample configuration for Firefox version 33.

```
{  
    "seleniumProtocol": "WebDriver",  
    "platform": "WINDOWS",  
    "browserName": "firefox",  
    "maxInstances": 10,  
    "version": "33"  
},
```

8.10 Cross browser testing

The Cross browser testing can be accomplished using the CATS framework, it is a very easy and hassle-free way compared to traditional non-automated method.

Here are the steps to accomplish the cross browser testing.

1. Set up the machines
 - a. Set up the each machine with a specific browser and version
 - b. Set up the selenium node
2. Set up the test scripts
 - a. Configure each test suite to run in using a specific browser and specific version
3. Run the Test scripts in Grid execution fashion to run across all the selenium nodes
4. Inspect the report for checking the cross browser compatibility.

8.11 Jenkins Execution

8.11.1 Jenkins Job Setup

CATS can be configured to run as Jenkins Job, either it can be scheduled to run on specific time or can be called as a subsequent job as part of CI build process.

Basic information about jenkins job configuration is given below, however for more information on Jenkins setup you may have to connect with the Project infra team members.

The screenshot shows the Jenkins dashboard at localhost:8080/jenkins/. On the left sidebar, there are links for 'New Item', 'People', 'Build History', 'Manage Jenkins', and 'Credentials'. In the main area, there is a table titled 'All' showing two items: 'CATS' (status red, icon sun) and 'D:\23 August\CATS\core' (status grey, icon cloud). Below the table, there is a link 'Icon: S M L'. At the bottom of the page, there is a footer bar with the number '76' and the CATS logo.

Now Click on New Item link

The screenshot shows the Jenkins dashboard at localhost:8080. On the left, there's a sidebar with links for New Item, People, Build History, Manage Jenkins, and Credentials. The main area displays a table with columns S, W, Name (sorted by name), Last Success, Last Failure, and Last Duration. A single row is present for 'ProjectName'. Below the table, there are sections for Build Queue (No builds in the queue) and Build Executor Status (1 Idle, 2 Idle).

Enter Job Name and click ok button

The screenshot shows the 'New Item' creation page in Jenkins. The 'Item name' field contains 'All'. Below it, there are five options: 'Freestyle project' (selected), 'Maven project', 'External Job', 'Multi-configuration project', and 'Copy existing item'. The 'Freestyle project' description states: '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.' The 'OK' button is visible at the bottom.

Click on Advanced Button

The screenshot shows the 'Configure' page for the 'ProjectName' job in Jenkins. The left sidebar includes links for Back to Dashboard, Status, Changes, Workspace, Build Now, Delete Project, Configure (which is selected), and Email Template Testing. The main form has a 'Project name' field set to 'ProjectName'. Under 'Advanced Project Options', there are checkboxes for Discard Old Builds, This build is parameterized, Disable Build, and Execute concurrent builds if necessary. Under 'Source Code Management', 'None' is selected. Under 'Build Triggers', 'Build after other projects are built' is checked. At the bottom are 'Save' and 'Apply' buttons.

Enter the path of the CATS core folder

Advanced Project Options

- Quiet period
- Retry Count
- Block build when upstream project is building
- Block build when downstream project is building
- Use custom workspace

Directory: E:\Test\CATS\core
Display Name:

Keep the build logs of dependencies

Source Code Management

- None
- CVS
- CVS Projectset
- Subversion

Build Triggers

Build periodically
Schedule: 0 7 * * *

[Save](#) [Apply](#)

Enter the frequency of every job. The below schedule represents to run the project every day at 7 AM

To trigger the CATS you have to add the build step “Execute Windows batch command” for windows or “Execute Shell” for Linux.

Add the batch/shell commands to make the CATS core directory as current directory then call the “start_cats_multithread.bat” for windows or “start_cats_multithread.sh” for linux

Source Code Management

- None
- CVS
- CVS Projectset
- Subversion

Build Triggers

Build periodically
Schedule: 0 7 * * *

Build

Execute Windows batch command
Command: Start_CATS

[Save](#) [Apply](#)

Provide Email id to which email notification has to be sent as a post build activity. Install **Email Extension Plugin** from manage plugins in manage Jenkins in the dashboard

Save

SriLaxmi Priya...

localhost:8080/job/ProjectName/configure

Jenkins > ProjectName > configuration

Post-build Actions

Editable Email Notification

Disable Extended Email Publisher

Project Recipient List: jsapient.com

Project Reply-To List:

Content Type: HTML (text/html)

Default Subject: CI - Execution # \$BUILD_NUMBER - Automation Test Report - \$BUILD_URL

Default Content: \${FILE,path="output/reports/HTML/ExecutiveSummaryReport.html"}

Attachments: output/reports/HTML/*.html

Can use wildcards like 'module/dist/*.zip'. See the [@includes of Ant tasks](#) for the exact format. The base directory is [the workspace](#).

Attach Build Log: Do Not Attach build Log

Content Token Reference:

Advanced...

Save Apply

This screenshot shows the 'Post-build Actions' configuration for a Jenkins job named 'ProjectName'. It includes an 'Editable Email Notification' section with fields for recipient lists, content type (HTML), subject, and default content. There's also an 'Attachments' section for build logs and a 'Content Token Reference' section.

SriLaxmi Priya...

localhost:8080/job/ProjectName/configure

Jenkins > ProjectName > configuration

Additional groovy classpath

Add

Save to Workspace

Triggers

Always

Send To: Recipient List

Add Advanced... Remove Trigger

Add Trigger Delete

Add post-build action

Save Apply

This screenshot shows the 'Triggers' configuration for the same Jenkins job. It includes a 'Triggers' section with an 'Always' trigger and a 'Recipient List' for notifications. There are buttons for adding triggers, advanced settings, and removing triggers. A 'Save' and 'Apply' button are at the bottom.

Help us localize this page

Page generated: Dec 9, 2015 1:48:35 PM REST API Jenkins ver. 1.640

Start Lync Vaibhav G... Downloads Inbox - sk... how to ru... Medtronic... CATS_Act... Java clien... ProjectN... Java - au... Document... Administr... 1:49 PM

This screenshot shows a browser toolbar with various links and icons, including 'Start', 'Lync', 'Vaibhav G...', 'Downloads', 'Inbox - sk...', 'how to ru...', 'Medtronic...', 'CATS_Act...', 'Java clien...', 'ProjectN...', 'Java - au...', 'Document...', 'Administr...', and a timestamp of '1:49 PM'.

If want to trigger manually go to home page click on project and build.

The screenshot shows the Jenkins dashboard with a context menu open over a project entry. The menu options are:

- Changes
- Workspace
- Build Now** (highlighted)
- Delete Project
- Configure
- Email Template Testing

Build progress will be shown

The screenshot shows the Jenkins dashboard with the same project entry as before, but the icon now indicates that a build is in progress.

Click on the progressing icon

The screenshot shows the Jenkins Build History page for the project 'ProjectName'. It lists the following builds:

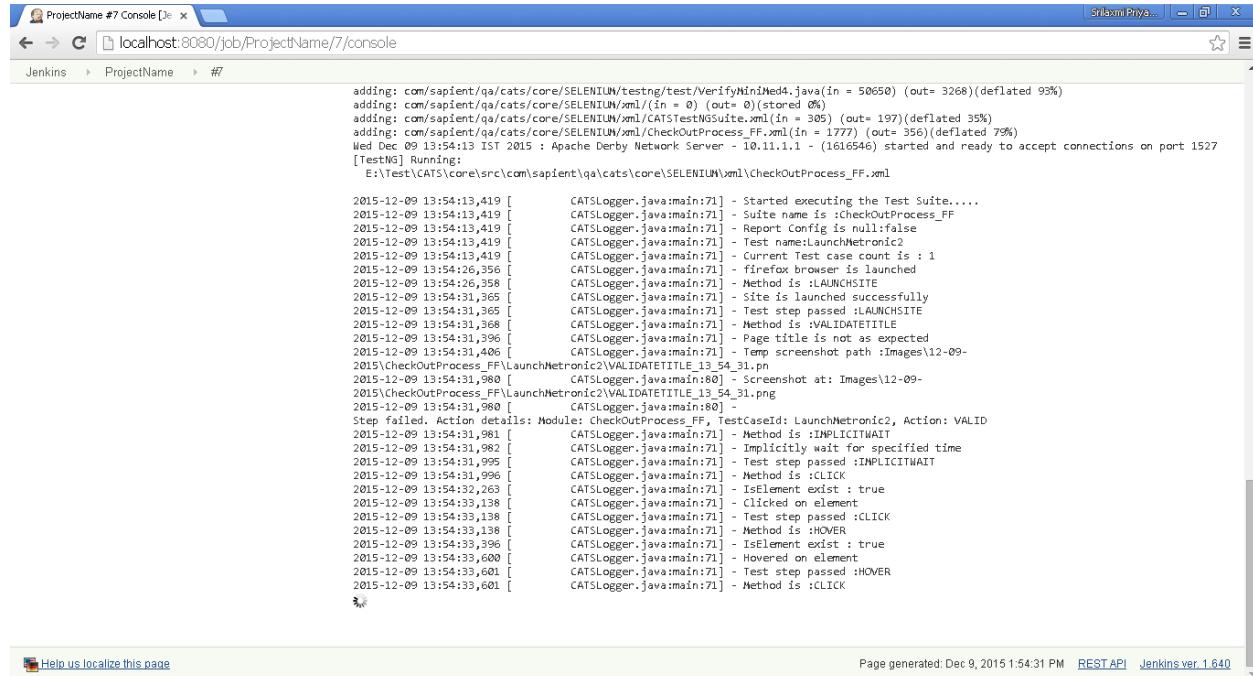
- #7 Dec 9, 2015 1:54 PM
- #6 Dec 9, 2015 9:24 AM
- #5 Dec 8, 2015 8:51 PM
- #4 Dec 8, 2015 8:47 PM
- #3 Dec 8, 2015 7:53 PM
- #2 Dec 8, 2015 7:52 PM
- #1 Dec 8, 2015 7:52 PM

At the bottom of the page are links for RSS feeds: RSS for all, RSS for failures, and RSS for just latest builds.

Click on the progressing icon

The screenshot shows the Jenkins Job page for build #7 of the project 'ProjectName'. The page title is 'Build #7 (Dec 9, 2015 1:54:03 PM)'. The progress bar shows 'Started 21 sec ago' and 'Build has been executing for 21 sec'. The sidebar includes links for Back to Project, Status, Changes, Console Output, Edit Build Information, and Previous Build. The main content area displays a note: 'No changes.' and 'Started by anonymous user'.

Click on console output link to see the progress.



The screenshot shows the Jenkins console output for a job named 'ProjectName'. The log displays various test steps and their outcomes, including assertions and browser launches. Key log entries include:

```
adding: com/sapient/qa/cats/core/SELENIUM/testng/test/VerifyNiniMed4.java(in = 50650) (out= 3268) (deflated 95%)
adding: com/sapient/qa/cats/core/SELENIUM/xml/in = 0) (out= 0)(stored 0%)
adding: com/sapient/qa/cats/core/SELENIUM/xml/CATSTestNGSuite.xml(in = 305) (out= 197)(deflated 35%)
adding: com/sapient/qa/cats/core/SELENIUM/xml/checkOutProcess_FF.xml(in = 1777) (out= 356)(deflated 7%)
Wed Dec 09 13:54:11 IST 2015 : Apache Derby Network Server - 10.11.1.1 (161656) started and ready to accept connections on port 1527
[TestNG] Running:
E:\test\CATS\core\src\com\sapient\qa\cats\core\SELENIUM\xml\checkOutProcess_FF.xml
```

2015-12-09 13:54:13,419 [CATSLocator.java:main:71] - Started executing the Test Suite.....
2015-12-09 13:54:13,419 [CATSLocator.java:main:71] - Suite name is :CheckOutProcess_FF
2015-12-09 13:54:13,419 [CATSLocator.java:main:71] - Report Config is null:false
2015-12-09 13:54:13,419 [CATSLocator.java:main:71] - Test name:LaunchMetronic2
2015-12-09 13:54:13,419 [CATSLocator.java:main:71] - Current Test case count is : 1
2015-12-09 13:54:26,356 [CATSLocator.java:main:71] - firefox browser is launched
2015-12-09 13:54:26,358 [CATSLocator.java:main:71] - Method is :LAUNCHSITE
2015-12-09 13:54:31,365 [CATSLocator.java:main:71] - Site is launched successfully
2015-12-09 13:54:31,365 [CATSLocator.java:main:71] - Test step passed :LAUNCHSITE
2015-12-09 13:54:31,368 [CATSLocator.java:main:71] - Method is :VALIDATETITLE
2015-12-09 13:54:31,396 [CATSLocator.java:main:71] - Page title is not as expected
2015-12-09 13:54:31,406 [CATSLocator.java:main:71] - Temp screenshot path :Images\12-09-
2015\CheckOutProcess_FFLaunchMetronic2\VALIDATETITLE_13_54_31.png
2015-12-09 13:54:31,988 [CATSLocator.java:main:80] - Screenshot at: Images\12-09-
2015\CheckOutProcess_FFLaunchMetronic2\VALIDATETITLE_13_54_31.png
2015-12-09 13:54:31,988 [CATSLocator.java:main:80]

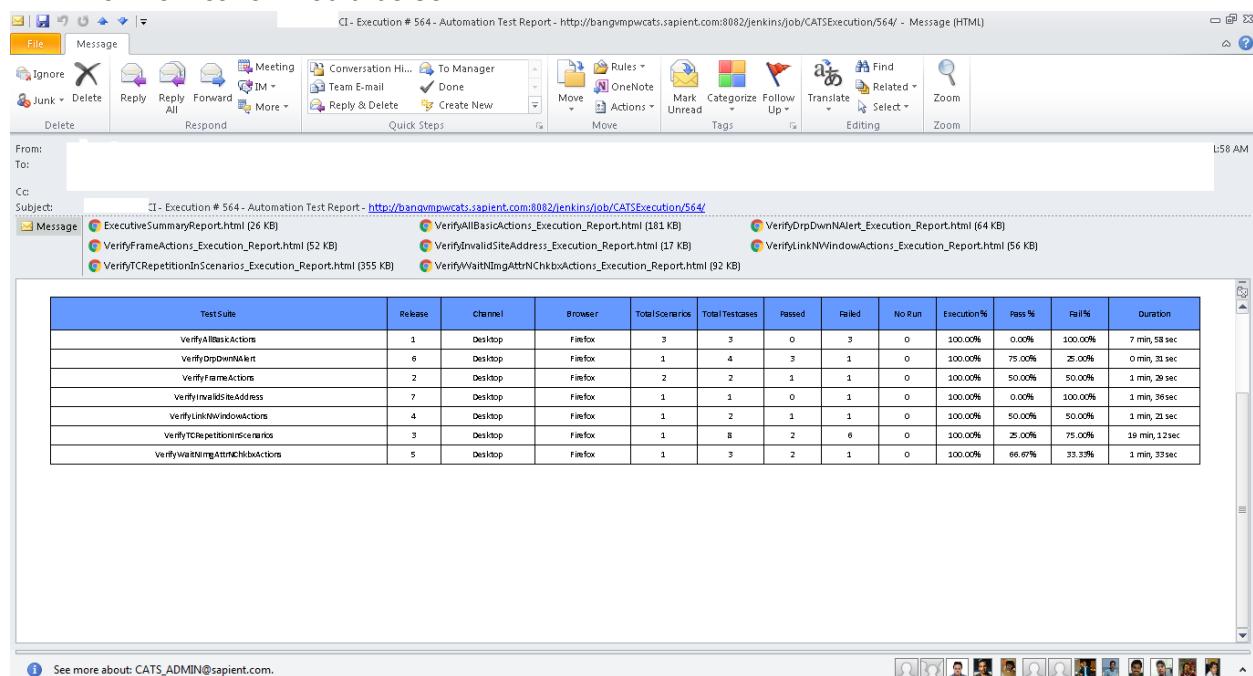
Step failed. Action details: Module: checkOutProcess_F, TestCaseId: LaunchMetronic2, Action: VALIDATETITLE_13_54_31

2015-12-09 13:54:31,981 [CATSLocator.java:main:71] - Method is :IMPLICITWAIT
2015-12-09 13:54:31,982 [CATSLocator.java:main:71] - Implicitly wait for specified time
2015-12-09 13:54:31,995 [CATSLocator.java:main:71] - Test step passed :IMPLICITWAIT
2015-12-09 13:54:31,996 [CATSLocator.java:main:71] - Method is :CLICK
2015-12-09 13:54:32,263 [CATSLocator.java:main:71] - IsElement exist : true
2015-12-09 13:54:33,138 [CATSLocator.java:main:71] - Clicked on element
2015-12-09 13:54:33,138 [CATSLocator.java:main:71] - Test step passed :CLICK
2015-12-09 13:54:33,138 [CATSLocator.java:main:71] - Method is :HOVER
2015-12-09 13:54:33,396 [CATSLocator.java:main:71] - IsElement exist : true
2015-12-09 13:54:33,600 [CATSLocator.java:main:71] - Hovered on element
2015-12-09 13:54:33,601 [CATSLocator.java:main:71] - Test step passed :HOVER
2015-12-09 13:54:33,601 [CATSLocator.java:main:71] - Method is :CLICK

Once all the suites are executed, then the JenkinsExecutionThresholdCheck will be called which will allow the Job to be marked as successful if the overall pass percentage meets the PASS_PERCENTAGE_THRESHOLD else the exception will be thrown to mark the build failed.

Note: Since the exception is thrown to mark the build as failed, the jenkins will not process any other command post the failure status. Hence in case you have any commands written to copy / move the artifacts after the execution you have have to move those commands under Post-build Action.

1. Email notification would be sent



The screenshot shows an email message from Jenkins titled 'CI - Execution # 564 - Automation Test Report - http://bangvmpwcats.sapient.com:8082/jenkins/job/CATSExecution/564/ - Message (HTML)'. The message body contains a summary of the test execution results:

From: [redacted] To: [redacted] L58 AM

Cc: [redacted]

Subject: CI - Execution # 564 - Automation Test Report - <http://bangvmpwcats.sapient.com:8082/jenkins/job/CATSExecution/564/>

Message attachments listed:

- ExecutiveSummaryReport.html (26 KB)
- VerifyAllBasicActions_Execution_Report.html (181 KB)
- VerifyDpDownNAler_Execution_Report.html (64 KB)
- VerifyFrameActions_Execution_Report.html (52 KB)
- VerifyInvalidSiteAddress_Execution_Report.html (17 KB)
- VerifyLinkNWindowActions_Execution_Report.html (56 KB)
- VerifyTCRepetitionInScenarios_Execution_Report.html (355 KB)
- VerifyWaitingAttnChkboxActions_Execution_Report.html (92 KB)

A detailed table of test results is shown below:

Test Suite	Release	Channel	Browser	Total Scenarios	Total Testcases	Passed	Failed	No Run	Execution %	Pass %	Fail %	Duration
VerifyAllBasicActions	1	Desktop	Firefox	3	3	0	3	0	100.00%	0.00%	100.00%	7 min, 58 sec
VerifyDpDownNAler	6	Desktop	Firefox	1	4	3	1	0	100.00%	75.00%	25.00%	0 min, 31 sec
VerifyFrameActions	2	Desktop	Firefox	2	2	1	1	0	100.00%	50.00%	50.00%	3 min, 39 sec
VerifyInvalidSiteAddress	7	Desktop	Firefox	1	1	0	1	0	100.00%	0.00%	100.00%	1 min, 36 sec
VerifyLinkNWindowActions	4	Desktop	Firefox	1	2	1	1	0	100.00%	50.00%	50.00%	1 min, 21 sec
VerifyTCRepetitionInScenarios	3	Desktop	Firefox	1	8	2	6	0	100.00%	25.00%	75.00%	19 min, 12 sec
VerifyWaitingAttnChkboxActions	5	Desktop	Firefox	1	3	2	1	0	100.00%	66.67%	33.33%	1 min, 33 sec

CATS can be configured to run as Jenkins Job, either it can be scheduled to run on specific time or can be called as a subsequent job as part of CI build process. Through jenkins CATS execution can be triggered at jenkins server itself or it can be triggered on slave machine.

Below documentation provides the configuration for standalone jenkins installation and execution on the same machine.

For pipeline job setup, slave configuration etc., please connect with your project infra person.

To get the execution on real browser through Jenkins , install Jenkins as slave on windows.

Note: When jenkins is executed as service the CATS execution will be performed on headless browser which means you can't see the execution. In order to see the execution on actual browser either you have to run the jenkins as standalone server or slave node must be configured to "*Launch Agent via Java Web Start*"

8.11.2 Jenkins Slave Setup

1. On your master machine go to **Manage Jenkins > Manage Nodes**.

The screenshot shows the Jenkins 'Manage Nodes' interface. At the top, there's a navigation bar with 'Hudson' and 'nodes'. Below it is a toolbar with 'Back to Dashboard', 'New Node' (which is highlighted with a red box), and 'Configure'. A sidebar on the left contains sections for 'Build Queue' (empty) and 'Build Executor Status' (two entries: '1 Idle' and '2 Idle'). The main content area displays a table with one row for the 'master' node. The columns are labeled 'Name', 'Response Time', 'Free Swap Space', and 'Free Disk Sp'. The 'Name' column shows an icon of a computer monitor and the text 'master'. The other three columns show 'N/A'.

2. New Node --> Enter Node Name.
3. Select Dumb Slave --> Press OK

This is a configuration dialog for adding a new Jenkins node. It has a 'Node name' field containing 'Jenkins Slave'. Below it is a radio button group: 'Dumb Slave' (selected, with a description about it being a plain dumb slave) and 'Copy Existing Node' (with a 'Copy from' field). At the bottom are 'OK' and 'Cancel' buttons.

4. Fill out the following:

This part of the configuration dialog shows several fields:
- Name: Jenkins Slave
- Description: (empty)
- # of executors: 2
- Remote FS root: C:\Jenkins\
- Labels: (empty)
- Usage: Leave this machine for tied jobs only
- Launch method: Launch slave agents via Java Web Start

Availability: Keep this slave on-line as much as possible

Node Properties

- Environment variables
 Prepare jobs environment
 Tool Locations

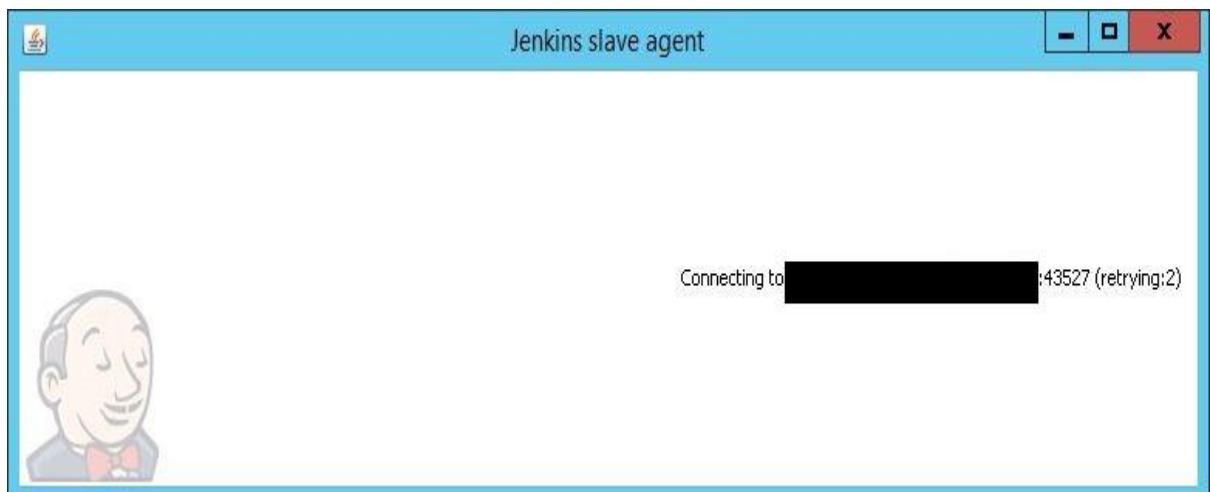
Save

- I. Set a number of executors (one or more) as needed.
- II. Set a Remote FS Root, a home directory for the master on the slave machine
 - a. For a Windows slave, use something like: "C:\Jenkins\"
 - b. TODO: add details.
- III. Select the appropriate **Usage** setting
 - a. For an additional worker: Utilize this slave as much as possible

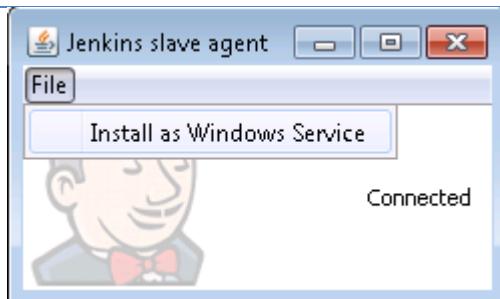
- b. For specialized jobs: Leave this machine for tied jobs only
- IV. Launch Method
- An easy way to control a Windows slave is by using Launch slave agents via Java Web Start (Recommended for Windows)
 - TODO: add steps for other methods
- V. Availability --> Keep this slave online as much as possible
- TODO: add details for each option.
- VI. Press OK
- VII.
5. Now you need to connect your slave machine to the master using the following steps



- Open a browser on the **slave machine** and go to the **Jenkins master server url** (<http://yourjenkinsmaster:8080>).
- Go to **Manage Jenkins > Manage Nodes**, Click on the newly created slave machine. You will need to login as someone that has the "Connect" Slave permission if you have configured global security.
- Click on the **Launch** button to launch agent from browser on slave, it will download the slave-agent jar.
- Run the slave-agent jar. If you encounter connection issue, then you could enlarge the popup windows to see the master port used and check your network configuration (firewall, port forward, ...)



- V. Now you should see the Slave machine connected under **Nodes**.
6. If you want the service to run on start-up of the slave machine do the following (Windows only directions):



- I. In the Slave agent program running on your slave machine,
- II. Click File --> Install as Windows Service. Note that this feature requires ".Net Framework 3.5"
- III. Start, type Services and Select the Services program.
- IV. Find Jenkins Slave in the list, Double click to open.
- V. Select Startup type --> Automatic.
- VI. Go to the Log On tab, change the Log on as to a user of your choice (Special user account Jenkins recommended).
- VII. Make sure that auto login is set for the slave machine for the user account, then the VM (or physical computer) should connect and be available when needed.

8.12 Bamboo Execution

To setup CATS on Bamboo for continuous integration, new plan and user agent should be created. If bamboo is sapient provided then reach out to DOJO support. Once the plan is created user can see the plan on bamboo dashboard.

Project	Plan	Build	Completed	Tests	Reason
▼ CATS	CATS	! #30	2 days ago	No tests found	Changes by catsuser <catsuser@BLRUBUNTU32619>

- Install CATS on your remote machine and set the JDK path. For more information on CATS installation refer to the CATS User Guide.
- Plan Configuration : Click on the Edit button on the right side of your plan(highlighted on red above) for setting up your plan configuration



Plan Configuration

▶ Stages & jobs	1
▶ Branches	0

Plan details

You can update the names and description of the current project and plan.

Project name *

Plan name *

Plan description

Plan enabled

Save **Cancel**

- Adding a job to the plan : Click on Stages and Add a Job

Automation Test Solution

Plan details **Stages**

Plan contents

Each stage within a plan represents a single unit of execution, such as a stage for compilation jobs, for running tests, or for generating artifacts.

Default Stage

- Default Job
- + [Add job](#)

CATS Stage

- + [Add job](#)

Add a new job

A job defines a single unit of execution within a stage, including what gets built and what executable to use, such as Maven, Ant or a custom script.

Create a new job

Create a new job and specify its source repository and executable.

Clone an existing job

Make a copy of a job and its entire configuration.

Close

- Select the Script task and make configurations for your script as below

Default Stage

Tasks

A task is a piece of work that is being executed as part of the build. The execution of a script, a shell command, an Ant Task or a Maven goal are only few examples of Tasks. [Learn more about tasks.](#)

You can use [runtime](#), [plan](#) and [global variables](#) to parameterize your tasks.

1 agent has the [capabilities](#) to run this job

Source Code Checkout
CatsRepoTrigger

Script
start_cats_multithread

Final tasks Are always executed even if a previous task fails

Drag tasks here to make them final

Add task

Script configuration

Task description: start_cats_multithread

Disable this task

Interpreter: /bin/sh or cmd.exe

Run your script with /bin/sh or cmd.exe

Script location: Inline

Script body:

```

1 E:
2 cd /CATS/CATS/core
3 "start_cats_multithread.bat"

```

Argument:

1. Add a description to your task
2. Interpreter should be **/bin/sh** or **cmd.exe**.
3. Script location should be **Inline**.
4. On the Script body
 - a. Change directory to CATS/core folder then
 - b. Enter “start_cats.bat” for sequential and “start_cats_multithread.bat”
5. In the Environment variables you can give your System path as system.Path

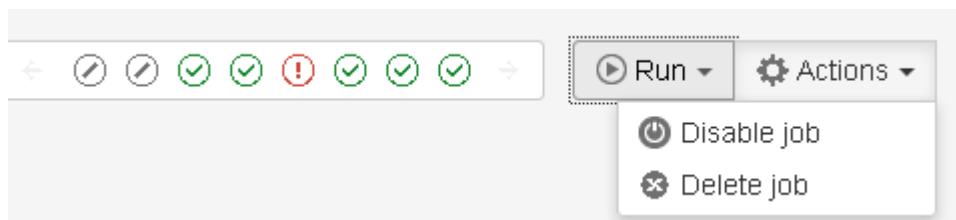
Argument

Environment variables

Working sub directory

Save [Cancel](#)

- After you save the above configuration you can run your plan



- This job can be added to development pipeline job with the help of intra team to enable execution on continuous integration.

8.13 Test Link Integration

Now CATS can update the execution status to the TestLink (Test Management Tool), the execution status will be triggered once all the suites are executed and the CATS report is generated.

To perform the test link update CATS will make use of Manual Mapping Sheet where user has provided. Every Single manual Test Case must be mapped to Suite and Scenario and Test Case (OR) Suite and Scenario (OR) Suite.

Test Link URL, Dev Key and Field Mapping information must be configured in the CATSReport.properties.

UPDATE_TEST_MANAGEMENT=YES #This must be set to YES to update the Test Link
=NO #This must be set to NO to update the Test Link Manually

TEST_MANAGEMENT=Testlink
TM_PROJECT_NAME=Test Automation Solution (CATS) #Testlink project name
TM_URL=http://10.150.6.153:81/testlink/index.php #Testlink URL
MANUAL_MAPPING_SHEET_PATH=../DataFiles/ManualMapping.xlsx
DEVKEY=32f552de37580cd29cba9c7bfe431f24 #User Dev Key, status will be updated using this user

CREATE_NEW_RUN_EVERY_EXECUTION= YES #This will create new build corresponding to all the valid configs in every row of Manual Mapping sheet .

TL_PLAN_NAME= #Column Name from Manual Mapping Sheet referring the plan name
TL_BUILD= #Column Name from Mapping Sheet referring the build name
TL_TC_ID_OR_NAME=MANUAL_ID #Column Name from Mapping Sheet referring the ID
TL_NOTES= #Column Name from Manual Mapping Sheet referring the notes (optional)
TM_BUG_ID= #Column Name from Manual Mapping Sheet referring the Bug ID (optional)

To Update the TestLink manually give the following command to cmd-
java -cp automatonplus-core-all.jar com.sapient.qa.cats.core.tms.UpdateExecutionInTms

8.14 Test Rail Integration

Now CATS can update the execution status to the TestRail (Test Management Tool), the execution status will be triggered once all the suites are executed and the CATS report is generated.

To perform the test link update CATS will make use of Manual Mapping Sheet where user has provided. Every Single manual Test Case must be mapped to Suite and Scenario and Test Case (OR) Suite and Scenario (OR) Suite.

Test Rail URL, API Key and Field Mapping information must be configured in the CATSReport.properties

UPDATE_TEST_MANAGEMENT=YES #This must be set to YES to update the Test Rail
=NO #This must be set to NO to update the Test Rail Manually

TEST_MANAGEMENT=Testrail

TM_PROJECT_NAME= #TestRail project name

TM_URL= #TestRail URL

MANUAL_MAPPING_SHEET_PATH=../DataFiles/ManualMapping.xlsx

TR_USERNAME= #User name, status will be updated using this user

TR_APIKEY= #User API Key, status will be updated using this user

CREATE_NEW_PLAN_EVERY_EXECUTION= YES #This will create new plan corresponding to every execution in CATS .

TR_TEST_PLAN_ID= #Column Name from Manual Mapping Sheet referring the plan Id

TR_TEST_CASE_ID= #Column Name from Manual Mapping Sheet referring the testCase Id

TM_BUG_ID= #Column Name from Manual Mapping Sheet referring the Bug ID (optional)

To Update the TestRail manually give the following command to cmd-

java -cp automatonplus-core-all.jar com.sapient.qa.cats.core.tms.UpdateExecutionInTms

8.15 Kanoah Integration

Now CATS can update the execution status to the Kanoah (Test Management Tool), the execution status will be triggered once all the suites are executed and the CATS report is generated.

To perform the test link update CATS will make use of Manual Mapping Sheet where user has provided. Every Single manual Test Case must be mapped to Suite and Scenario and Test Case (OR) Suite and Scenario (OR) Suite.

Kanoah URL, API Key and Field Mapping information must be configured in the CATSReport.properties

UPDATE_TEST_MANAGEMENT=YES #This must be set to YES to update the Kanoah
=NO #This must be set to NO to update the Kanoahl Manually

TEST_MANAGEMENT=Kanoah

TM_PROJECT_NAME= #Kanoah project name

TM_URL= <https://qa.tools.sapient.com/jira/rest/atm/1.0> # Kanoah URL
MANUAL_MAPPING_SHEET_PATH=../DataFiles/ManualMapping.xlsx

ATM_USERNAME= #User name, status will be updated using this user

ATM_PASSWORD= #UserPassword, status will be updated using this user

CREATE_NEW_RUN_EVERY_EXECUTION= YES #This will create new run corresponding to all the valid configs in every row of Manual Mapping sheet

ATM_TEST_PLAN_ID= #Column Name from Manual Mapping Sheet referring the plan Id

ATM_TEST_CASE_ID= #Column Name from Manual Mapping Sheet referring the testCase Id

To Update the Kanoah manually give the following command to cmd-

java -cp automatonplus-core-all.jar com.sapient.qa.cats.core.tms.UpdateExecutionInTms

8.16 Step by Step Execution

User can perform the step by step execution through CATS UI, where user needs to click on the debug button once debug is enabled, execution will halt after every step is executed. Now user needs to click on resume to execute next step.

During step by step execution mode user would be able to see the active values details with respect to the step on the console.

8.17 Multiple Instance Execution

With CATS 6.0 we have enabled multiple instance of CATS from the single location through command line. One must be careful while enabling this capability. Follow the below steps to enable this feature.

1. Duplicate the default folder located in CATS/core/Config/**default** and rename the folder to something that you want to refer to eg: uat, brand1, env1, **newconfig**, etc.,
2. Hibernate configuration must be updated. Open the CATS /core /Config /**newconfig** /**Hibernate.cfg.xml** and update the port and folder name highlighted below both should be different from one that is used in other configs.
 - <property name="connection.url">
jdbc:derby://localhost:**1529**/config/**newconfig**/reportdbg;create=true
</property>
3. Open the CATSConfig.properties and update the CATS_INPUT_JSON_PATH to reflect the new config path
 - CATS_INPUT_JSON_PATH = ..//input/**newconfig**/Inputs.json
 - DERBY_SERVER_PORT=1529
4. Open the CATSToolsInput.properties and update the OutputPath
 - OutputPath=..//input/**newconfig** /
5. Open the CATSReport.properties and update the output folder location to avoid the report conflicts and report overwrite.
 - HTML_REPORT_PATH=..//output/**newconfig** /HTML/
 - PDF_REPORT_PATH=..//output/**newconfig** /PDF/
 - REPORT_ARCHIVE_PATH=..//output/**newconfig** /History/
6. Finally while triggering the CATS execution you need to use command prompt /shell and named argument -configPath should be used and the path should be either absolute or relative path with respective path from inside the core folder. For example
 - start_cats_multithread.bat -configPath ..//config/**newconfig**/

9 Analytics Testing

Currently CATS support Adobe and Google Analytics testing on http sites. To enable Analytics testing user needs to associate the configuration to suite where Analytics is set to Yes.

This configuration enables selenium driver to pass the request and responses through a specified proxy which user can convert them into HAR file in turn the content in that file will be used to verify the TAGs generated.

Using CATS user can start the response capturing prior to performing some task then stop the response capturing post the task. The response captured between the transactions will be used for tag verification.

ANALYTICSSTART: Kicks off the request-response capturing

ANALYTICSEND: Stops the request-response capturing process and stores the information either in a user defined variable or default.

ANALYTICSVERIFYTAGFROM: Verifies the tag and corresponding values from the saved responses.

ANALYTICSVERIFYTAG: Verifies the tag and corresponding values from recently generated responses.

10 Service testing

10.1 Introduction

CATS supports a total of 8 services, below is the introduction to Service testing using CATS:

1. IO - local input/output disk operations
2. NFS - Network file system IO operations
3. XML - Search, replace and validation operations on XML file/Strings
4. JSON - Search, replace and validation operations on JOSN file/Strings
5. FTP - file/folder operations on remote FTP servers
6. sFTP - file/folder operations on remote sFTP servers
7. REST - rest services for put, post, get and remove operations
8. IBM MQ - Queue search retrieval and validation operations of XML messages

Service testing uses TCM and TDM excel workbooks to create test cases and store test data and details respectively.

- TCM - will hold the test case information to perform actions.
 - TCM test step is tied to a ServiceData entry by its key referenced in the Service Config column of TDM
- TDM - will hold the below information related to each service in the sheets below:
 - **ServiceData** - Will hold any user parameters required to be processed by the server. Ex, filenames, query, input and output parameters. Service data should always be linked to a serviceConfig entry. *QueryParam, InputParam and OuputParam columns support the variables (local, global and TDM)*

10.2 IO

IO services allow CATS to read the local file/folder content and perform actions on the file and folders. All the action related to IO operation begins with io, for more details on the actions and usage please refer the CATS Action Reference html.

10.3 NFS

NFS services allow CATS to read the intra network remote file/folder content and perform actions on the file and folders. All the action related to NFS operation begins with nfs, for more details on the actions and usage please refer the CATS Action Reference html.

10.4 FTP

FTP services allow CATS to read the remote file/folder content hosted on a FTP server and perform actions on the files and folders. All the action related to FTP operation begins with ftp, for more details on the actions and usage please refer the CATS Action Reference html.

10.5 SFTP

SFTP services allow CATS to read the remote file/folder content hosted on a SFTP server and perform actions on the files and folders. All the action related to SFTP operation begins with sftp, for more details on the actions and usage please refer the CATS Action Reference html.

10.6 XML

XML services allow CATS to read the local file/folder content and perform XML actions on the file. . All the action related to XML operation begins with xml, for more details on the actions and usage please refer the CATS Action Reference html.

10.7 IBM MQ

IBM MQ services allow CATS to read the remote IBM MQ service queue and perform search and retrieve operations on the queues.

- OR column will have Service as its entry if the test step is service related.
- The IBM MQ Actions will require the **Expected Results** containing the local variable reference to store the output. The format will be {{variableName}}.
- **SERVICE CONFIG** column will point to the ServiceData entry and should be in the format: \$ServiceData.KEY where key is the matching entry in the TDM ServiceData sheet.

TEST CASE NAME	CONTROLLER	PAGE	ELEMENT	ACTION	TEST DATA	EXPECTED RESULT	DESCRIPTION PRIORITY	FLAGS	SERVICE CONFIG	RE
MQDemo	Service			mq.CheckQueueExists		{}{{mqQueueExists}}			\$ServiceData.PRODUCT4	

ServiceData Sheet

- Service Config should have the **KEY** to match what was provided in the **SERVICE CONFIG** column above.
- **QUERY_PARAM** should contain the Node name in case of **CheckValueInMQMessage** and **GetMQMessageWithXMLNodeValue** actions and empty otherwise.
- **INPUT_PARAM** should contain the node value in case of **CheckValueInMQMessage** and **GetMQMessageWithXMLNodeValue** actions and empty otherwise.

KEY	SERVICE_CODE	QUERY_PARAM	INPUT_PARAM	OUTPUT_PARAM	STATUS_CODE
PRODUCT4	MQVerifyValue	OrderNumber	MV0000741		

ServiceConfig Sheet

- SERVICE_CODE in ServiceData sheet should match an entry in the ServiceConfigSheet
- SERVICE column should contain the **IBM MQ Queue name**.
- Each Service should be connected to a SERVER_ID
- MEDIA_TYPE should contain the expected format ex: "application/xml"

- Current supported messages in IBM MQ are XML format only.
- Future enhancements will add support for other formats.

SERVICE_CODE	METHOD	SERVICE	BASEJSON	SERVER_ID_SECURED	MEDIA_TYPE
MQVerifyValue	Verify	RLDS.RLUS.ORDEREXPORT.DIB.OMS		7	application/xml

ServerConfig Sheet

- SERVER_ID should have a corresponding entry in the Server Config sheet.
- BASESITEID should hold the Queue Manager information
- SERVER_URL should have the IBM MQ host URL Ex:- IP address or UniqueName.domainName.com
- REST_END_POINT_PORT should hold the IBM MQ Port value
- REST_MIME_TYPE should contain the application/xml as value as CATS supports only XML format for IBM MQ
- USERNAME should be the user id allowed on the system
- PASSWORD should be the correct password to be tested on the FTP server.
- AUTH_SERVICE_NAME - will hold the Channel information

SERVER_ID	BASESITEID	SERVER_URL	REST_VERSION	REST_END_POINT_PORT	REST_MIME_TYPE	REST_PROTOCOL	CLIENT_ID	CLIENT_SECRET	GRANT_TYPE	USERNAME	PASSWORD	AUTH_SERVICE_NAME
7 QAQMGR	del2rl333.sapient.com		63000	application/xml	MQ		esbsvc			SYSTEM.ADMIN.SV		

10.8 JSON

JSON services allow CATS to read the local file content and perform JSON actions on the file. All the action related to JSON operation begins with json, for more details on the actions and usage please refer the CATS Action Reference html.

10.9 REST

REST services allow CATS to connect to remote REST Servers and perform PUT, POST, GET and DELETE operations. All the action related to REST operation begins with rest, for more details on the actions and usage please refer the CATS Action Reference html.

10.10 SOAP

SOAP services allow CATS to connect to SAOP request and get the response. All the action related to SOAP operation begins with soap, for more details on the actions and usage please refer the CATS Action Reference html.

11 Mobile Testing

11.1 Windows

11.1.1 Android Emulator (Genymotion) Setup:

- Refer section header [Android Emulator Installation](#)
- While accessing GenyMotion if you get an error message "The virtual device got no IP address"
 - Go to BIOS settings -> Go to System Configuration -> Enable Virtualization Technology -> Save and Exit from BIOS settings
 - Open Oracle VM VirtualBox -> Select your simulator -> Go to settings of simulator -> Select General tab -> Select Basic sub tab -> Select your OS type -> Select your Version of your OS -> Save and Exit

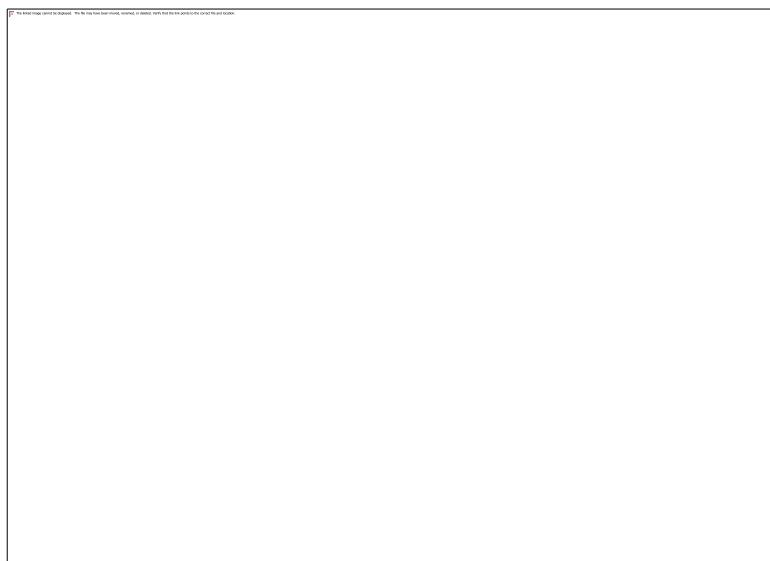
-
3. Restart your GenyMotion and then select device.
 - c. Add devices to GenyMotion through “Virtual device Create” wizard
 - d. In case you don’t see the device UI then you need to update the Graphics Drivers
 - e. Start the AVD (Android Virtual Device) Manager, In the program group Android SDK tools. Select the tab “Android Virtual Device”, click “Create” button to create the android simulator for the selected device. Provide all the details.
 - I. AVD Name: the name of your choice
 - II. Device: Select the device name from combo
 - III. Target: Select the API, which is downloaded
 - IV. Select all other options as needed/required
 - V. Click on “Create” button to create the device

11.1.2 Execution

- a. Start the Appium application
- b. Change the property “GENYMOTION_PATH” to the path of GenyMotion player in CATSConfig.properties
- c. Provide the configuration in the summary sheet, for the suites that need to be run on mobiles, using the Config_N (where N is the serial number). Refer section CONFIG Sheet
- d. Run Start_CATS.bat

11.1.3 Actual Android Device Setup :

1. Install Appium for Windows [here](#).
2. Install Node js for Windows [NodeJS 64-bit](#) / [NodeJS 32-bit](#)
3. Install Android SDK from [here](#).
 - 3.1. Once you install Android SDK, search SDK manager from the start Menu and open it.



Make sure you have installed the three highlighted packages.

And also if you scroll down in the above SDK manager you will find *Google USB Driver*

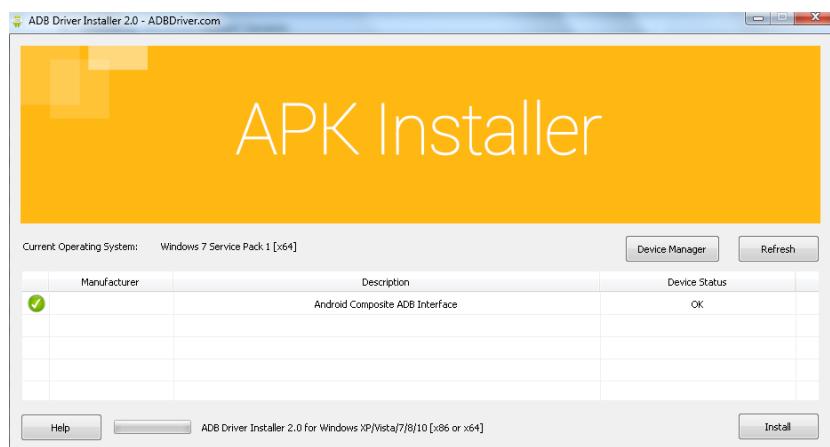


4. Setting up the Environment Variable:

- Add new environment variable **ANDROID_HOME** and value should be SDK location pointing to the folder to **(drive location)\Android\android-sdk**
- Append the environment PATH **%ANDROID_HOME%\platform-tools**

5. Turn on **USB Debugging** on your real device and connect to your system.

- Connect it through the MID mode or Camera(PTP) mode .
- The Adb driver will get installed on your system automatically .
- On the command line check for **adb devices** .
- If the Adb driver is not installed download the Universal AdbDriver from [here](#).



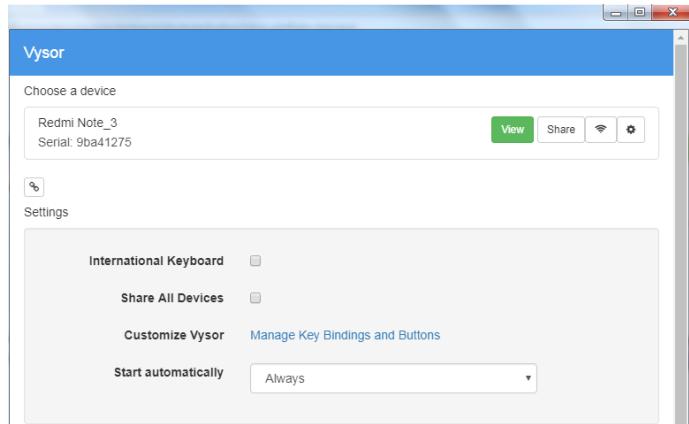
- Install the driver and check for adb devices again on you command line as depicted in the below screen shot.

```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

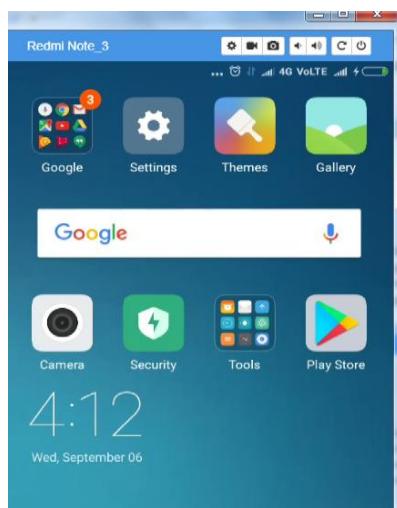
C:\Users\MIMRA2>adb devices
List of devices attached
9ba41275      device
```

6. Cast your Real Device on your System with Vysor

- a. Install the **Vysor** extension from the Chrome WebStore [Vysor](#) and launch the app.
- b. Provide permission on your device to install vysor
- c. Click on View .

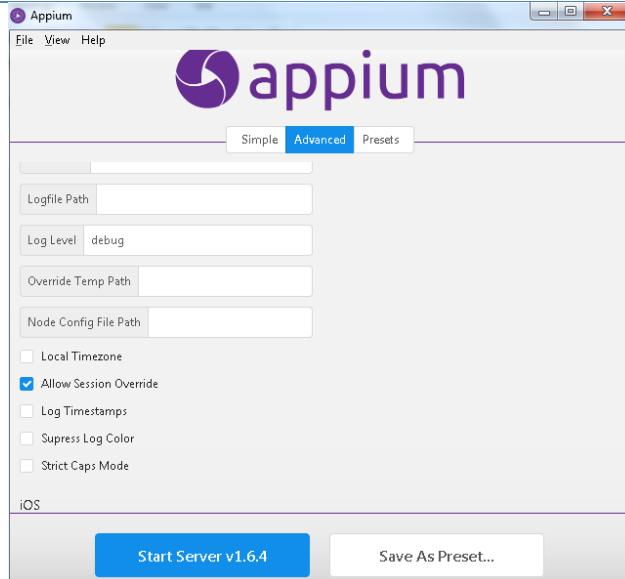


- d. The Real device will be casted on your system .



7. Open Appium on your system :

- a. Get the Host and Port from the UI for example : Host : 127.0.0.1
- b. Port :4723 . These host and port is used in the TCM Config..
- c. Note: The Host and Port can be changed by user from Advanced option on the appium UI.
- d. Click on Advanced Tab and check Allow Session Override and start the Server.



8. TCM Configurations : Update your TCM Config as depicted below.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
ID	DESCRIPTION	CHANNEL	BROWSEVIEW	NAME	DRIVER	DEFAULT VERSION	PLATFORM	UUID	BUNDLEID	APP PATH	PROTOCOL	SERVER	PORT	CONTEXT	PKGNAME	ACTIVITY		
1	Android Real Device	Mobile	Chrome	Redmi	AndroidDevice	5.1.1	Android			http		127.0.0.1	4723	/wd/hub				

A	B	C	D	E	F	G	H	I	J	K
RELEASE	TEST SUITE	TEST CASES	DESCRIPTION	RUN	CONFIG	BATCH	OWNER	SUITE VARIABLES	LABELS	ITERATIONS
Test23	Demo23		SCENARIOS_29	Y	CONFIG_1					

9. After the appium server is started and TCM Config is updated , you are ready to run your suites on Android Real Device through Cats.

11.2 Mac

11.2.1 Machine Setup

- a. Refer section header IOS Simulator Installation

11.2.2 Execution

1. Start the Appium application
2. Open the terminal, navigate to /Users/catsuser/Desktop/Build/CATS/core
3. Run "sh start_cats.sh"

11.2.3 Execution on iOS Real Device on Safari Browser

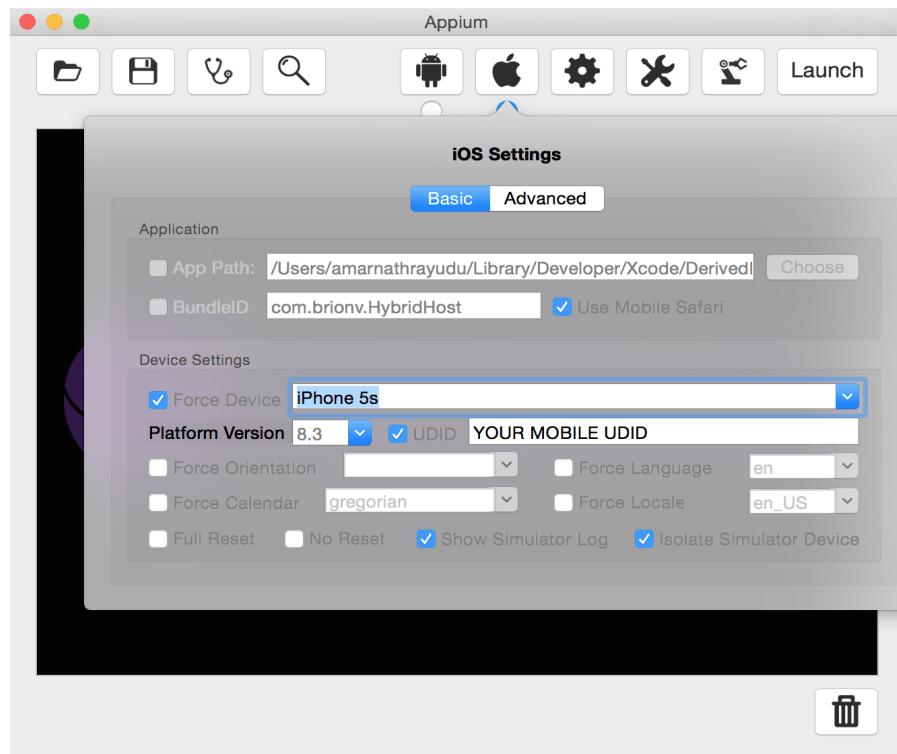
1. **Pre-requisites:**
 - a. Xcode installed on Mac OS X.
 - b. JDK 8 should be installed and JAVA_HOME and BIN path should be set
 - c. Node.js should be installed
 - d. Device added to Apple developer program
 - e. Get SafariLauncher.app(<https://github.com/budhash/SafariLauncher1.5k>)
 - f. Launch the SafariLauncher.app on connected device from Xcode. Import the SafariLauncher in Xcode and Run it. Once the build is successful
 - f. Download appium from github (<https://github.com/appium/appium293>)
 - g. Install ios-webkit-debug-proxy ... use below command from terminal.
 - h. *Install brew install ios-webkit-debug-proxy*

NOTE : If brew is not working you need to install Home Brew . Run this command on the terminal

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

2. Connect the device to your mac and verify the device is shown under Xcode
3. Launch the app on device from xcode
4. Open appium IDE, add your device details as mentioned in the below screen.

In the App path : give the safarilauncher.app path



Note : FOR - appium 1.6 (without GUI) and XCODE 8

Install Appium - `npm install -g appium@1.6`

Download XCODE 8

External Dependencies -

1. Install ideviceinstaller - `brew install ideviceinstaller`
2. Install carthage -`brew install carthage`
3. Install ios-deploy - `npm install -g ios-deploy`
4. Install device console - `npm install -g deviceconsole`
5. Install xcpretty - `gem install xcpretty`
6. Install libimobiledevice - `brew install libimobiledevice --HEAD`
7. Install appium-doctor - `npm install -g appium-doctor`

WebDriver Agent Configuration : go to the path on your terminal

```
/usr/local/lib/node_modules/appium/node_modules/appium_xcuitest_driver/WebDriverAgent/
sudo mkdir -p Resources/WebDriverAgent.bundle
sudo sh ./Scripts/bootstrap.sh -d
```

For Safari on iOS Real Device , you need to run the webkit proxy server as:

Open a terminal , type : ios_webkit_debug_proxy -c <UDID>:27753 -d

For Permission :

*Sudo chmod -R 777 .appium-xcuitest-driver/**

Build the WebDriverAgent project with XCODE pointing to your device or Simulator.

Build Command: xcodebuild -project WebDriverAgent.xcodeproj -scheme WebDriverAgentRunner -destination 'id=<UDID>' test

NOTE: NO GUI for Appium 1.6 above version

1. Launch the appium from IDE or from the appium server
Command: appium
2. TCM Configurations : In the TCM CONFIG sheet :

ID	NAME	VERSION	DRIVER	PLATFORM	UDID	BUNDLEID
8	iPhone 6s	10.2	iOSDevice	ios	4ebf1585a05f83cef3752ee2b4c746f509a08a8e	com.sapientnitro.india.ARCH

APPPATH	BROWSER	PROTOCOL	SERVER	PORT	CONTEXT
/Users/user/Library/Developer/Xcode/DerivedData/SafariLauncher-hddfykhiwgxzwedanrfcaigmunf/Build/Products/Debug-iphoneos/SafariLauncher.app	Safari	http	0.0.0.0	4723	/wd/hub

In the Summary Sheet :

RELEASE	TESTSUITE	TEST CASES - SCENARIOS	CHANNEL	BROWSERS	RUN	CONFIG	BATC
Mobile	RoomBookingMobile	SCENARIOS_87	Mobile	Safari	Y	CONFIG_8	

Make sure that appium and ios_debug_webkit_proxy is running

Run CATS.

11.3 SeeTest Mobile Lab Execution

11.3.1 Android Device

Pre-requisite : Appium , Node JS , Android SDK , Vysor(Chrome App) for casting on the screen.

For installation of all the above softwares refer to the Mobile Testing section of the CATS User Guide.

TCM configuration:

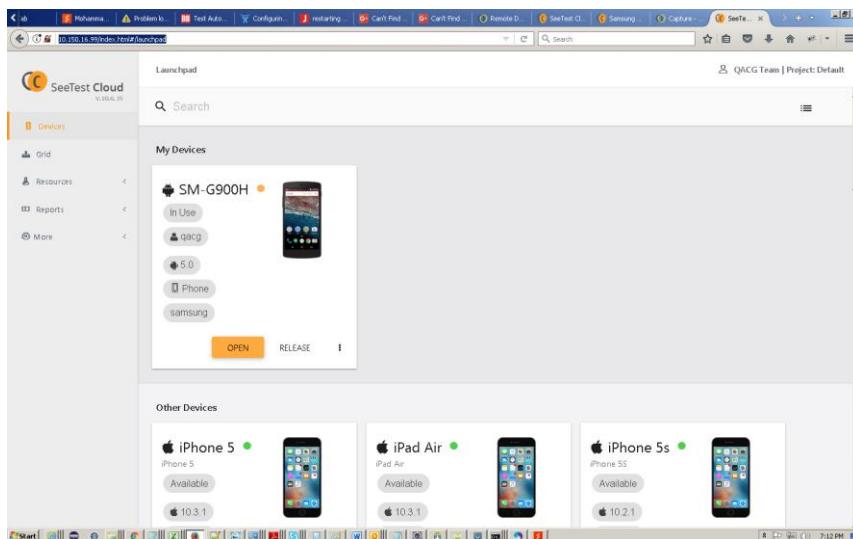
Config Sheet Configuration:

ID	NAME	VERSION	DRIVER	PLATFORM	UDID	BUNDLEID	APPPATH	BROWSER	PROTOCOL	SERVER	PORT	CONTEXT	PKGNAME	ACTIVITY	DESCRIPTION
1	SamsungGalaxyS5	4.4.4	AndroidDevice	Android				Chrome	http	127.0.0.1	4723	/wd/hub			config details for android

Summary Sheet Configuration:

RELEASE	TEST SUITE	TEST CASES - SCENARIOS	CHANNEL	BROWSERS	RUN	CONFIG
PROD	IN_VERIFY_CART_CHROME_MOBILE	MOBILE_1	Mobile		Y	CONFIG_1

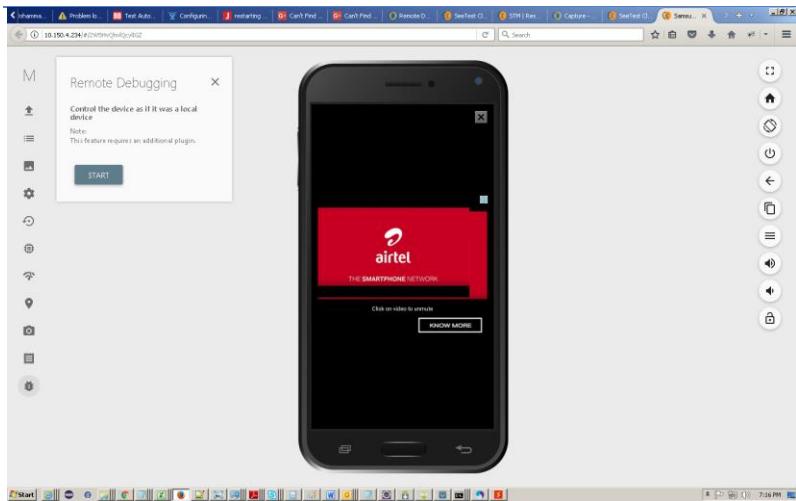
1. We will launch this url in the browser <http://10.150.16.99>
2. Login with your Credentials.
3. See Test Launch Pad page will be launched .



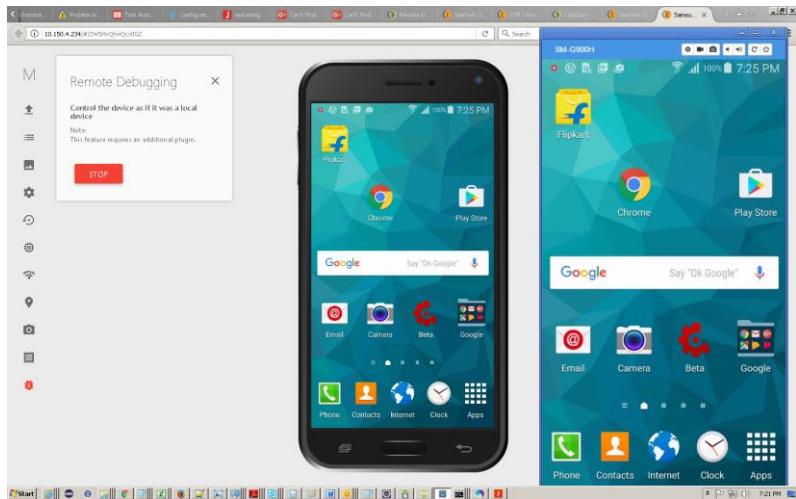
4. Select a device and click on Open



5. Click on Remote Debugging Option.



6. Click on Start
7. For the first time it will prompt you to install the remoteDebuggingDaemon pluggin.
8. Install the pluggin in your system .
9. Experitest folder will be installed in your system at location [C:\Program Files \(x86\)\Experitest](C:\Program Files (x86)\Experitest)
10. Keep a back up of your adb.exe in the [C:\Program Files \(x86\)\Android\android-sdk\platform-tools](C:\Program Files (x86)\Android\android-sdk\platform-tools)
11. Replace the adb.exe with the adb.exe of the adb folder of Experitest [C:\Program Files \(x86\)\Experitest\SeeTestRemoteDebugging\bin\adb](C:\Program Files (x86)\Experitest\SeeTestRemoteDebugging\bin\adb)
12. After installing your pluggin your device will start and the device will also screen cast on Vysor .

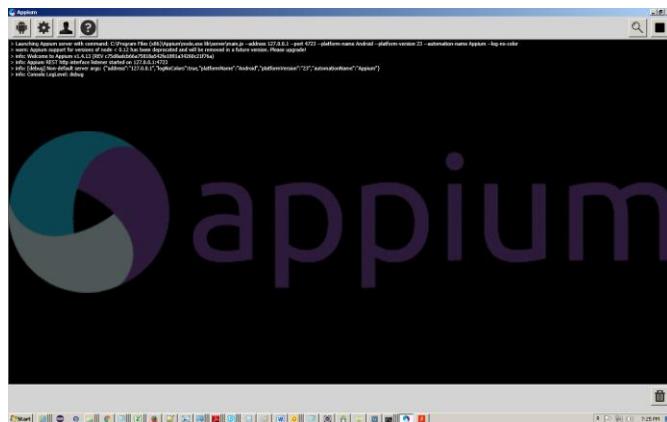


13. Go your Command Line and check for devices attached :

```
Administrator: C:\Windows\system32\cmd.exe
C:\Users\MIMRA2>adb devices
List of devices attached
4d007c754dca412f      device

C:\Users\MIMRA2>
```

14. Start Appium.



15. Run your Cats Mobile Scripts.

11.3.2 iOS Device

Prerequisite: Follow the setup guide for real iOS device execution through CATS in the user guide above

1. Login to your SeeTest Cloud with the URL <http://10.150.16.99/index.html#/login>

← → C | ① 10.150.16.99/index.html#/launchpad

The screenshot shows the SeeTest Cloud Launchpad interface. On the left, a sidebar lists navigation options: Devices, Grid, Resources, Reports, and More. The main area is titled 'Launchpad' and contains a search bar labeled 'Search'. Below it, there are two sections: 'My Devices' and 'Other Devices'. Under 'Other Devices', an iPhone 5 is listed with the following details: Status 'Available' (green dot), OS '10.3.1', and categories 'Phone' and 'Apple'. At the bottom of this card are buttons for 'MANUAL' (highlighted in orange), 'AUTOMATION', 'RELEASE', and a more options menu (three dots). To the right of the card is a small image of the iPhone 5.

2. Select your iOS device to be reserved and click on MANUAL and click on OPEN ANYWAY

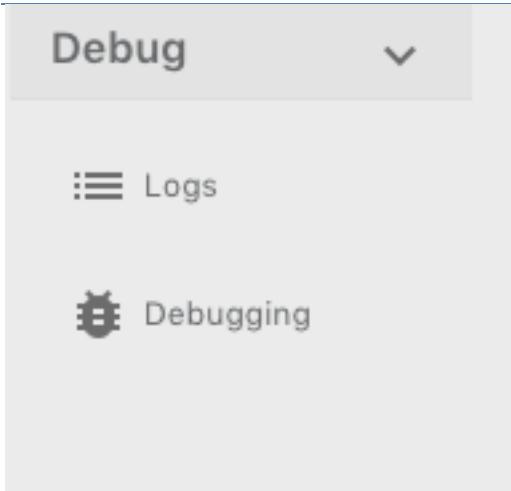
New Tab Is Blocked By The Browser

The browser has blocked this site from opening a new tab.

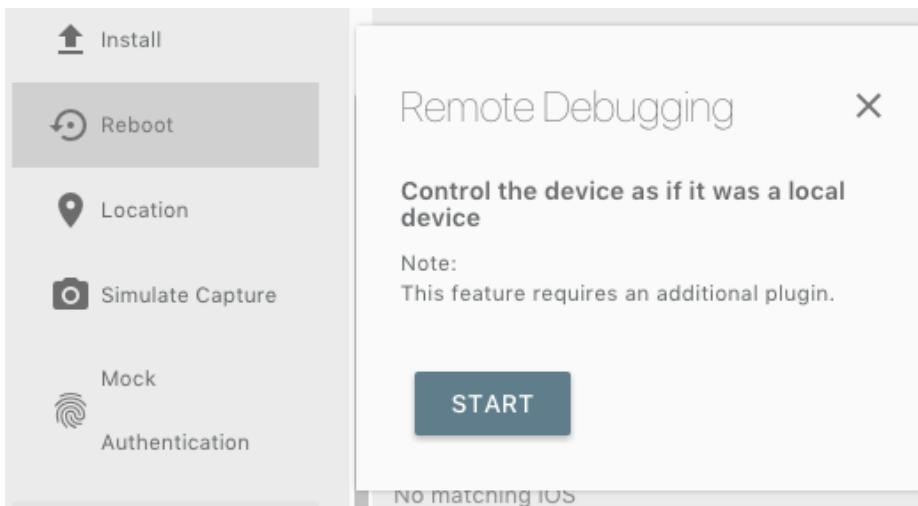
[OPEN ANYWAY](#)

Learn how to allow pop-ups in your browser.

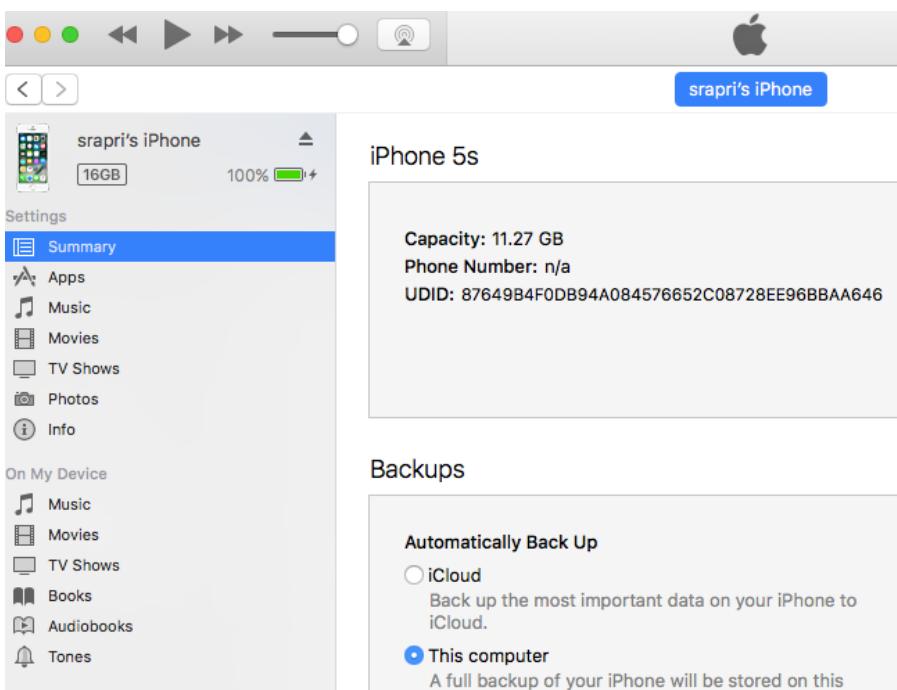
3. In the bottom of the page Click on Debugging



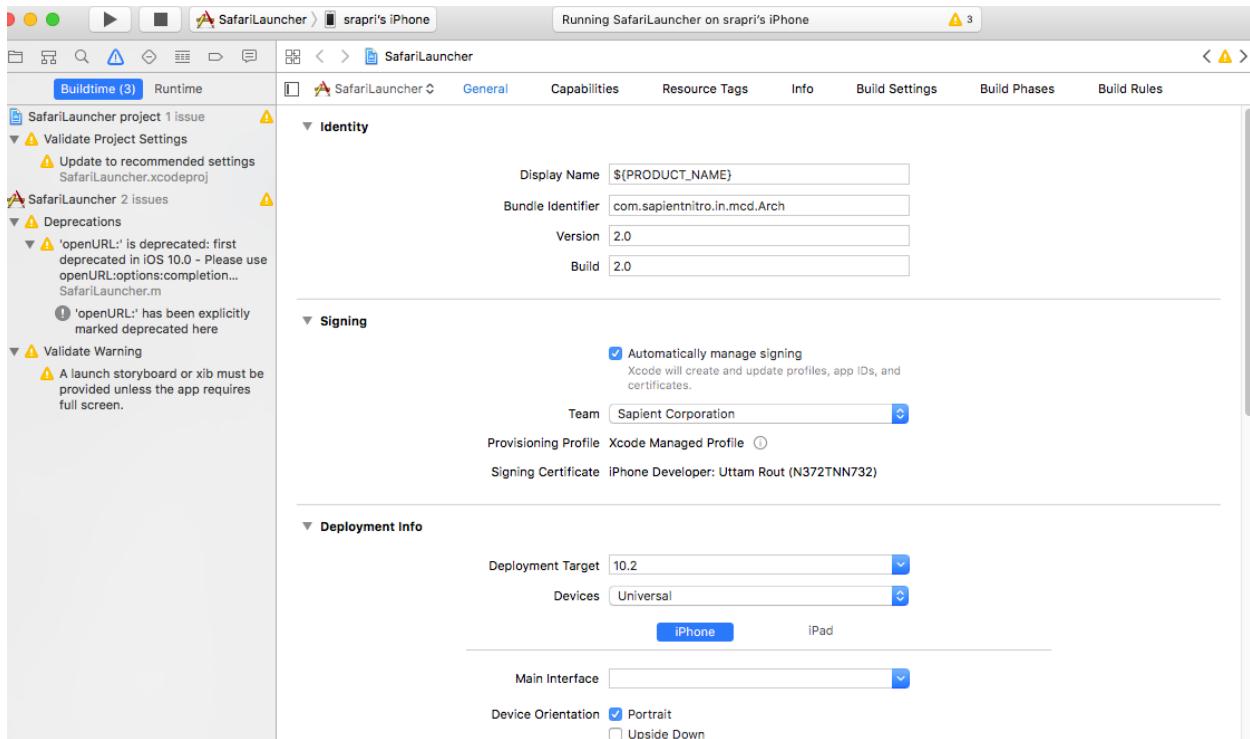
4. After you click on Debugging you will get a Pop-Up to Start



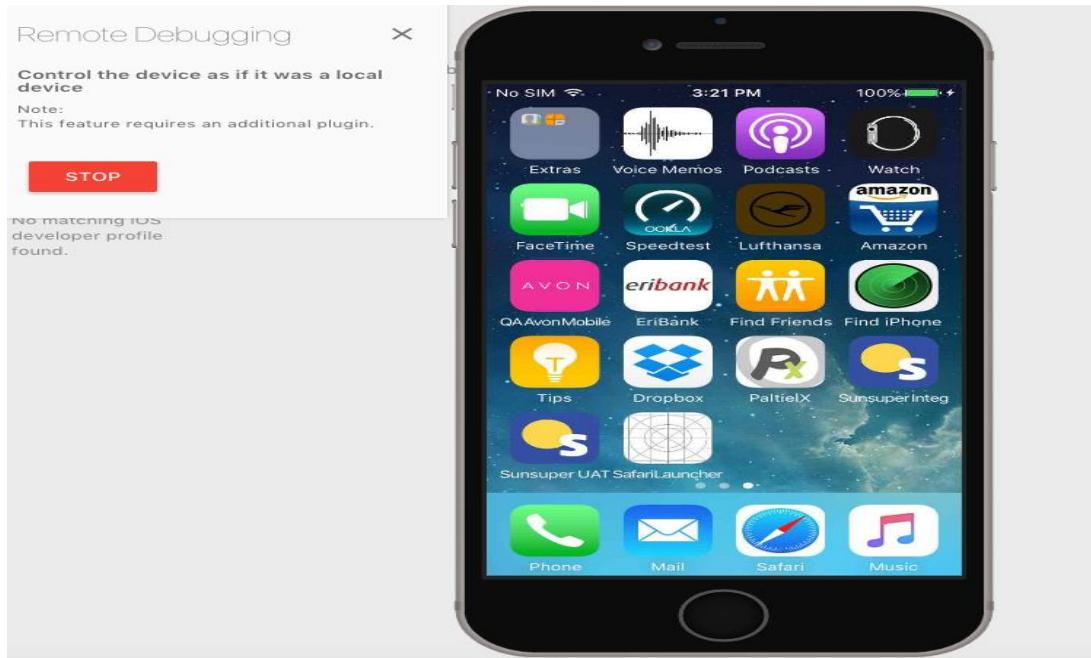
5. Open iTunes and ensure that your iOS device is connected. Copy the device name and the UDID. For example Device Name : srapi's iPhone and UDID of the Device : 87649b4f0db94a084576652c08728ee96bbaa646



6. The device name and UDID is to be updated in the TCM CONFIG sheet against NAME and UDID column respectively.
7. Open X-code and Build the SafariLauncher App on your iOS Device



8. Get the Safari Launcher Path from XCODE>File>Project Settings. The Safari Launcher Path is to be used in the App Path of the CONFIG sheet in the TCM. Once build is done safari launcher appears on the device



9. Run Appium from your Terminal

```

user — node /usr/local/bin/appium — 80x24
[BLRMMITHU11943:~ user$ appium
[Appium] Welcome to Appium v1.6.5
[Appium] Appium REST http interface listener started on 0.0.0.0:4723

```

10. Run `ios_webkit_debug_proxy` and ensure that the Syntax of `iOS_Webkit_Debug_Proxy` is mentioned in the Real Device setup

```

user — bash — 80x24
BLRMMITHU11943:~ user$ ios_webkit_debug_proxy -c 72a984e0a58b4c848c0ebad2131b3d4 6fe15a4dd:27753 -d

```

11. Update the device details in the CONFIG sheet and now you are ready to execute CATS script on SeeTest Mobile Lab

ID	NAME	VERSION	DRIVER	PLATFORM	UDID	BUNDLEID	APPATH	BROWSER	PROTOCOL	SERVER	PORT	CONTEXT
30	sapri's iPhone	10.3	iOSDevice	iOS	87649b4f0db	com.sapientn	/Users/user/U	Safari	http	0.0.0.0	4723	/wd/hub

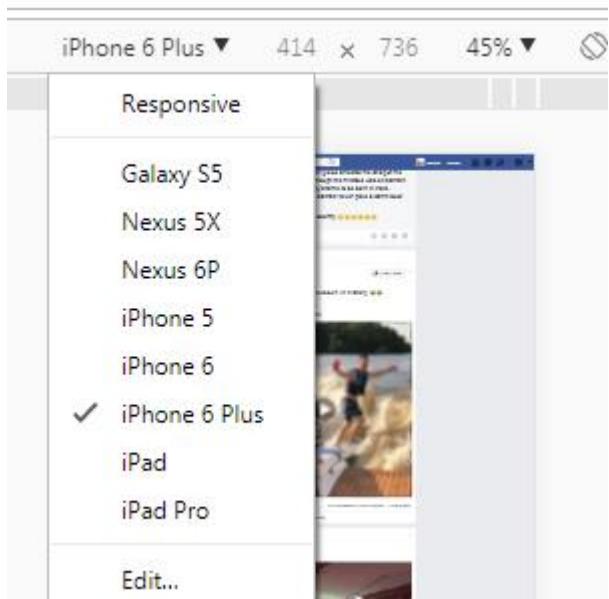
11.4 Chrome Emulator

11.4.1 Responsive Testing

As your mobile audience grows, responsive mobile-friendly web design becomes all the more important. Web content needs to look and feel great across a wide variety of devices and network conditions. But testing the quality of your mobile experiences takes longer and makes debugging more complex. Device mode brings the insights of mobile testing to your browser tab through the power of mobile emulation.

11.4.2 Supported Devices

It helps user to select device by providing the name of the device in TCM and helps to test touch actions. List of devices supported by chrome can be found from the chrome browser in device mode



11.4.3 Configuration Setup

To use this feature, make the following configuration changes in the TCM.

1. First make the changes in CONFIG sheet in TCM as per the below snapshot.

Set the following columns as

- ID as any value (number or text)
- NAME as device name as per above list. (Ex. Apple iPhone 5, Nokia N9)
- DRIVER as ChromeMobileEmulator

ID	NAME	VERSION	DRIVER
3	Apple iPhone 5		ChromeMobileEmulator

1. Then refer the configuration from CONFIG column in the summary sheet, for the suites that need to be run mobile emulation, using the CONFIG_N (where N is the ID used in CONFIG sheet) as per below snapshot.

Set the following columns as

- CHANNEL as Mobile
- BROWSERS as Default
- CONFIG refer the CONFIG_ID. (Ex CONFIG_3).

RELEASE	TEST SUITE	TEST CASES - SCENARIOS	CHANNEL	BROWSERS	ITERATIONS	RUN	CONFIG
REL3	AccountReg	SCENARIOS_AccountReg	Mobile	Default	1	Y	CONFIG_3

To test the above in Selenium grid setup:

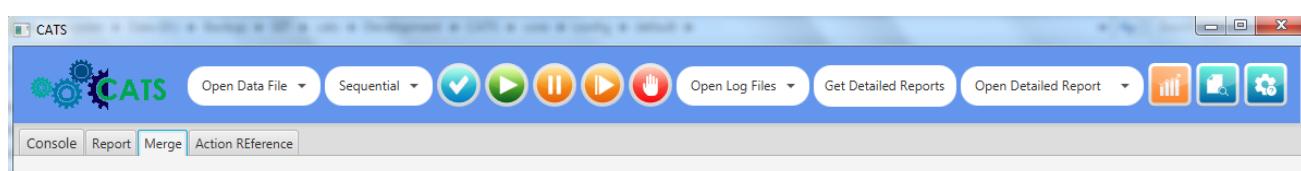
In the Config sheet, mention driver as **ChromeMobileEmulator _versionno**, testi happens in nodes available matching version number in grid.

ID	NAME	VERSION	DRIVER
3	Apple iPhone 5		ChromeMobileEmulator_43

12 CATS UI

12.1 Toolbar

CATS tool bar section has five sections 1) Data Files 2) Execution Control 3) Log Files 4) Reports 5) User guides. Refer the screenshot below



Data Files: Using open data file menu user can access Data files associated for current instance.

Execution Control: Contains execution mode, validate, execute, debug, resume and stop buttons.

Log Files: From this drop down, user can access execution log file and excel error log file

Report: Once execution is started user can see the execution report, by clicking on get detailed report button followed by the suite name from the drop down list. Once complete execution is done then user can open summary report by click on the report icon.

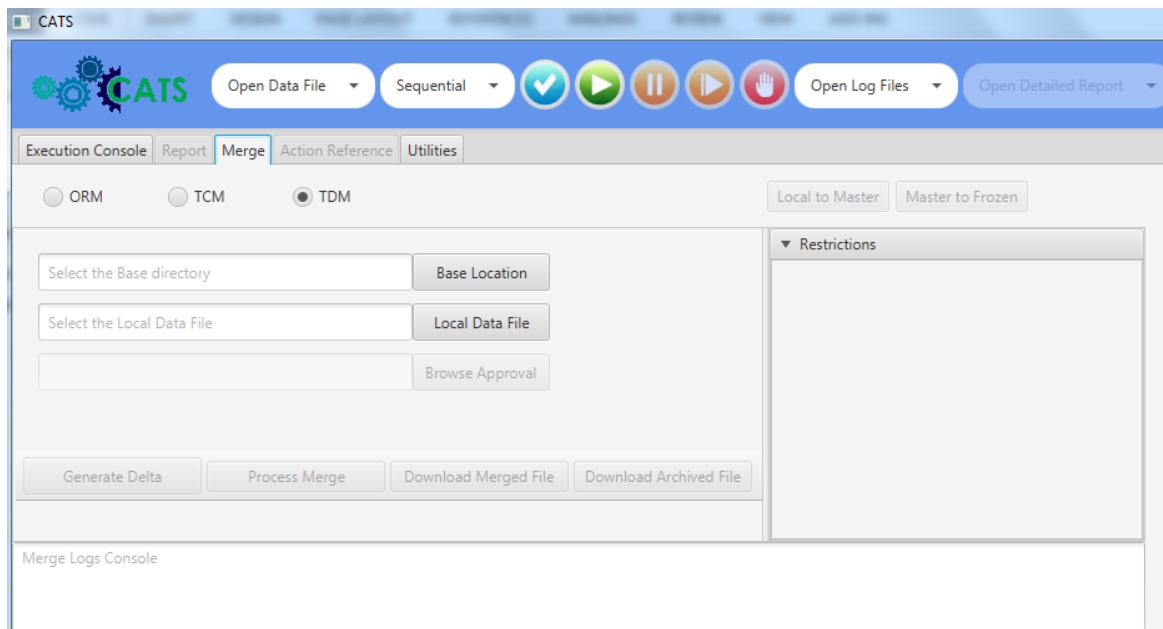
User Guides: Action Reference Guide and User guide link

12.2 Merge Tool

CATS UI provides the data files merge feature. With 6.0 CATS user can merge TCM and OR files, this feature helps when more than one user is working on the same data files and wanted to update the data files to keep everyone's data file up to date.

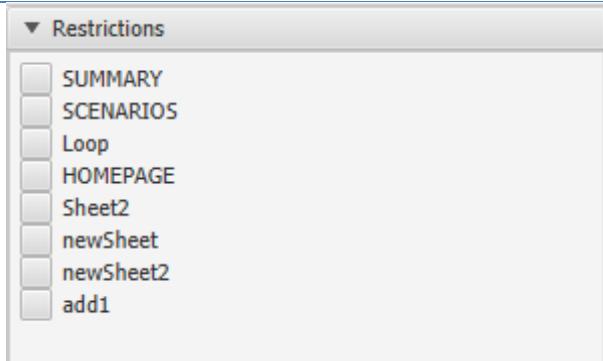
12.2.1 Baseline Data File

1. Project team has to decide centralized location where the Frozen(Base) and Master DataFile to be kept
2. First time project lead needs to select the base location and the TCM or OR file path and click on **Local to Master** followed by **Master to Frozen**. Now Frozen/Base version is set



12.2.2 Merge Data File

1. Browse base location where the frozen Data Files are placed
2. Browse your local file
3. From the Restriction section select the sheets which needs to be merged
4. The click on Generate Delta Changes on Local. This action will compare the local data file to master data file and master data file to frozen data file
5. Once the delta file is generated it will be in JSON format, now you have to click on Show Changes to be Approved this will generate the excel file which will list out the changes made.
6. User has to review the changes and **USERACTION** column must be set to **YES** to accept the changes. In case of new modification and new addition it will appear YES by default, however in case of remove or conflicts (where somebody has already modified the content compared to frozen version) user must explicitly have to mark it as YES after review



7. Save the approval file and browse the file location then click on process merge.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
Sheet	TC ID	TEST CASE	CONTROL	OR	PAGE	ELEMENT	ACTION	TEST DATA	EXPECTED	DESCRIPTI	PRIORITY	FLAGS	SERVICE C	REMARKS	ROW_NU	SEQ	STEP SEQ	STATUS	USERACTI
1	newSheet	newTCM	newTCM																
2																		ADDEDTE	YES
3																			
4																			
5																			

8. Once merge is done you can get the Master Data File and continue to use.

13 Custom Actions

CATS framework lets user create their own action using the predefined template and plug into the framework and continue to refer them as action.

There are two ways user can add their custom action to the CATS framework 1. Custom Action Jar 2. Runtime Custom Action

Custom Action JAR: User has to get the custom action project and create add their actions/class (using predefined format) and compile it and name it as **customaction.jar** and place this jar file inside the CATS core folder.

Runtime Custom Action: Users will update the predefined custom action java template with their code and place the file .java files inside the **customaction** folder, during execution all the java files inside the folders will be compiled and customactions-runtime.jar will be created. User need not to worry about the compilation and jar creation. This feature is enabled for anyone within the team to be able to look at the custom action code and modify if required instead of depending on one person who maintains the custom action project.

14 Help

14.1 Getting help

Please go through the User Guide, CATS Action Reference html and the Video of specific section to make yourself more comfortable with CATS.

Please contact the Core team member of CATS team, to get any help, clarification or discuss any new feature.

For CATS Support : Raise a support ticket in jira [Cats Support](#)

CATS [Download](#)

[CATS Reference](#) Guide and Videos

14.2 Trouble shooting

There are multiple ways to trouble shoot the CATS engine. Since the tool runs on the console, console provides the first level of error reporting. In addition to the console, here are the other forms of getting the feedback.

Console

Errors list on Excel sheet

Error log - This is mostly used by the CATS technical team. The viewer gets the first information on the errors.

14.3 Best Practices

CATS team suggest the below for the optimal experience with the CATS tool

1. Always try to install the CATS tool under D:\CATS folder, this require least amount of configuration.
2. Test Case name and Id must be brief meaningful words
3. Disable java updates.

14.4 List of “Don’t do”

Similar to Best practices, here is the list of things you don’t want to do for the smooth running of CATS.

2. Using the Underscore (_) in any of the sheet names. The underscore has a special meaning, it is used for separating the sheet name and the number of Test case, scenario.

14.5 FAQ

Here is the list of frequently asked questions on this tool.

Question: I'm not able to run the tool, getting the error "javac is not a program"

Answer: The tool requires the JDK 1.8, not just the JRE. Look at the pre-requisites and install them.

Question: Execution started and browser gets opened and does nothing.

Answer: Your browser version might not be compatible with the driver supported version, check the version and downgrade it if required.

Question: When I run the test after compilation I get Test Complete done but there is no execution happened.

Answer: Run the TCM validation and fix the errors, the possibility could be you have referred the scenario or test case which doesn't exist or it has trailing or leading line break or space.

Question: While running the test I get the exception during class compilations

Answer: Check if Suite, Scenario and Test Case Name and ID has invalid character

Question: Object exist but doesn't perform any action like Select, Click etc.

Answer: You have to re-look at the xpath, if the event is tied to different element it won't work, it is not a CATS issue. You have to learn the XPath better

Question: While running CATS I get the message on the console as CATS license status invalid, only 2 test cases will be processed.

Answer: Seems your license is expired, Copy your userMacGen.gen from the CATS License folder and send a mail to the Cats Licensing Team<catslicensingteam@sapient.com> attaching your userMacGen.gen, requesting a New License.

Question: CATS fails to start execution when we run it from the UI, we get the error message “No java files found to compile!”

```
15:17:18 12606 - > Data file parsing completed. You can find excel errors in: ./input/Excel2JsonToolOutput2017-01-09  
15:17:19 13337 - No java files found to compile!  
15:17:19 13363 - Running TestNG...  
15:17:21 15656 - Done! Tests completed. Reports can be found in core/output/ directory.
```

Answer-1: System environment variable is either not configured for Java path or it is pointing to C:\ProgramData\Oracle\Java\javapath., update the java path follow the installation steps in section

Answer-2: Java files were not generated from the scenario and test case files, this could be due to

1. The Scenario ID, Test Case Name or Test Sheet Name contains invalid name.
2. Java File contain invalid character refer the command prompt console to know the java file from where the error is prompted

Question: Sometimes when I am running the Script I have noticed that, script immediately gets completed without any generated report or any error. I am guessing the input file somewhere I have error but it is not got caught, hence it is not showing any nor it is running the script.

Answer: Remove the Scenario rows, create them again freshly and try running it again.

Question: I am facing issues executing my scripts on CATS. Only my chrome browser gets initialized again and again. None of the script gets executed.

Answer: Check the version of your chrome browser, download the chrome driver compatible to your version and replace it with the chrome driver in your CATS Drivers folder.

Question: When I try to run my scripts it opens an empty instance of Firefox and fails to launch the site.

Answer: Check for the Mozilla Firefox browser version, if it has upgraded to a version above 46, downgrade it to version 46 and below as CATS doesn't support Firefox version above 46.Keep the automatic update off for your browser

Question: When I run CATS, getting the error message: CATS_INPUT_JSON_PATH is set incorrectly or CATSConfig.properties file not found - CATSConfig.properties and the scripts are not executed.

Answer: Check the SCENARIOS sheet in your TCM. The cells under the TEST CASES or TEST DATA KEY might be formatted by an improper text, space or any character. Remove space or character if appended and try running CATS again.

Question: I am facing an issue in clicking a check box in my application web page. Actually it's hidden. I have tried all possible combinations of xpaths from my end and it's not working. It is getting located successfully but when I run the test its shows “Element doesn't exist” in the report.

Answer: Selenium determines an element is visible or not by the following criteria visibility != hidden, display != none (is also checked against every parent element), opacity != 0 (this is not checked for clicking an element), height and width are both > 0, for an input, the attribute type !=

hidden.Your element might be matching one of those criteria where opacity = 0 .So is the reason , the action fails to select the element.

Question: I have tried all options like Hover, DoubleClick, ClickAt etc., but it has not failed with ‘Element doesn't exist’ error.

Answer: We developed a work around ClickJS using javascriptexecutor which is very helpful in this scenario as it directly plays with DOM. However we haven't pushed this action to the core actions. You can request the java file from the CATS team and include it in your customactions.jar.

Question: ENTER action fails to enter data into the respective field.

Answer:Check your xpath .The xpath should point to the exact element .For e.g. if we have an input field embedded in a div, it should be written for that input tag. If the action fails still you can use ENTERAPPEND replacing ENTER action.

Question: I'm getting Suite Error/Scenario Error irrespective of the correct scenarios and test case info

Answer: It could be due to space between scenarios or test case separator. Please remove all the leading and trailing space and line breaks. In both Suites and Scenarios.

Question :While running CATS , I am getting the error “javac is not recognized as an internal or external command,operable program or batch file”.

Answer:Check your system variable path , jdk \bin path might be missing or it might be pointing to jre instead of jdk..Replace it with your jdk path.

Question :My xpath is correct still CATS is unable to locate the element.

Answer :In the OR sheet under a page there cannot be two or more elements with the same ELEMENTNAME.

Question : CATS is unable to pick the correct data from the TDM .

Answer : When adding FIELDS or ERs in your TDM make sure that the FIELD name or the ER name is not repeated (TDM cannot have the same FIELD or ER name).

Question : Element Locator gets identified in the firepath but CATS fails to locate the element .

Answer : Check for iFrames , the element must be inside an iframe . You need to switch to the frame first then use actions accordingly.For switch to frame related actions follow the CATS Action Reference Guide.

Question :In every click on Get Detailed Reports on the Cats UI I get , “Font 'Perpetua Titling MT' is not available to the JVM”.This error message also occurs in the CatsLog.log file .Any Solutions?

Question : In my xpath I am using the correct text to locate the element which I am setting in the OR sheet using SETORVARIABLE at runtime.But after the SETORVARIABLE step , the next step fails to execute.

Answer : Check the text that you are trying to use in the xpath.If it contains more than one sentence for eg. “Cats is an automation Framework. It's easy to use .” you can take only one sentence of it as a text in your contains function of xpath.xpath doesnot allow two or more sentences with respect to text function.

Question : PDF reports are coming blank whereas HTML reports are fine

Answer: Install the latest Acrobat PDF reader and open the PDF files again .

Question : I am not able to enter the data which contains double quotes through Test Data column with the ENTER action . Eg - "CATS is a "Sapient" automation framework"

Answer: Use Escape Sequence using characters : Backspace is replaced with \b, Newline is replaced with \n , Tab is replaced with \t ,Carriage return is replaced with \r,Form feed is replaced with \f,Double quote is replaced with \" ,Backslash is replaced with \\ and must be properly escaped to be used within Strings.The above Data will look like this : \"CATS is a \"Sapient\" automation framework\"

Question : While testing on Real Device or Emulator through CATS , the chrome browser opens and closes and the website is not launched.

Answer: Go to this path on your windows system : C:\Program Files (x86)\Appium\node_modules\appium\node_modules\appium-chromedriver\chromedriver\win and replace the chromedriver 2.18 with the latest chromedriver downloaded from the link : [latestChromeDriver](#) , which will make it work.

Question : License Folder is not generating after running start_cats.bat or LaunchCATSUI.bat

Answer: Please check your jdk version . It should be jdk 1.8 and not jdk 9 .

Question: Whenever automation is triggered on Safari Browser , we get the error message on the Browser as

Safari can't open the page “https://www.britishairways.com/travel/home/public/en_in” . The error is: “WebKit encountered an internal error” (WebKitErrorDomain:300)

Answer: Please clear the cache . on Safari go to Develop > Clear Cache . Also clear the history from the Safari Preferences .

Question: I am getting an error : “Please reinstall autoconf. Sorry :(

Answer : Run the command: brew install automake , this will install all the automake dependencies .