

EIT Configuration Tool Kit

*Thesis submitted in partial fulfillment of the requirements for the award of degree
of*

Master of Engineering
in
Computer Science and Engineering

Submitted By
Neha Mittal
(801632030)

Under the supervision of
Mr. Sumit Miglani
Assistant Professor



COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
PATIALA – 147004

June 2018

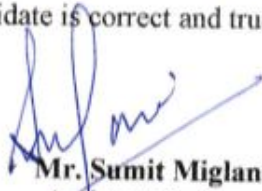
CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*EIT Configuration Tool Kit*" in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Computer Science and Engineering* submitted in Computer Science and Engineering Department of Thapar Institute of Engineering and Technology, Patiala is an authentic work carried at *Oracle India Pvt. Ltd, Bangalore* under the supervision of industrial mentor *Mr. Christopher Nihar* and faculty mentor *Mr. Sumit Miglani*, and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University


Neha Mittal

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.


Mr. Sumit Miglani
Assistant Professor
Computer Science and Engineering Department
Thapar Institute of Engineering and Technology

Acknowledgement

No volume of words is enough to express my gratitude towards my guide, **Mr. Sumit Miglani**, Assistant Professor, Computer Science and Engineering Department, Thapar University, who has been very concerned and has supervised the work presented in this dissertation report. He has helped me to explore this vast field in an organized manner and provided me with all the ideas on how to work towards a research oriented venture.

I am also thankful to Dr. Maninder Singh, Head of Department, CSE and Dr. Rupali Bhardwaj, P.G. Coordinator, for the motivation and inspiration that triggered me for the dissertation work.

I would also like to thank the staff members and my colleagues who were always there in the need of the hour and provided with all the help and facilities, which I required, for the completion of my dissertation.

Most importantly, I would like to thank my parents, friends and the almighty for showing me the direction out of the blue, to help me to stay calm in the oddest of the times and keep moving even at times when there was no hope.

Neha Mittal
801632030

Abstract

It is important in software industry to deliver the right product at the right time to customer with all the quality checks. Enterprise manual testing becomes difficult and time consuming if one tries various input combinations. Moreover, it is costly to manually analyze all results. This project is motivated by the drawbacks of manual regression testing and how to overcome those drawbacks using automated tools.

In case of manual testing, changes in source code, hardware configuration or operating system leads to repetition of whole process. These efforts can be reduced by means of an automation. Automation testing does the predefined actions written through code and it is also helpful in analyzing result. Automating this process also serves as a reporting solution which reports the success or failure of a particular test along with the reason of failure.

For doing it, we came up with an Enterprise Integration Testing [EIT] Tools and Utilities which are used by the EIT team to manage their day to day integration testing. It contains four major functionalities, EIT Dashboard, EIT Reports, EIT Tools, EIT Task Management with multiple utilities under each of them.

Table of Contents

Certificate:	i
Acknowledgement	ii
Abstract	iii
Table of Contents	iv
List of Figures	vi
1. Introduction	1
1.1 Project Definition	1
1.2 Motivation	1
1.3 Overview of Projects	1
2. Literature Survey	3
2.1 Retail Products	3
2.1.1 Retail Merchandising System (RMS)	3
2.1.2 Store Inventory Management (SIM)	3
2.1.3 Warehouse Management System (WMS)	3
2.1.4 Retail Invoice Matching (ReIM)	3
2.1.5 Retail Price Management (RPM)	4
2.1.6 RIB (Retail Integration Bus)	4
2.1.7 Allocation	4
2.2 Foundation Data	5
3. TECHNOLOGIES USED	6
3.1 Automation Tool's Survey	6
3.1.1 Open Script	6
3.1.2 Synergy	8
4. Automate Nightly Run	10
4.1 Introduction:	10
4.2 Additional Requirements:	10
4.2.1 Table Requirement:	10
4.2.2 Constraint for table	11
4.2.3 Requirement for bat file:	11
4.2.4 Other requirements and constraints:	12

4.3 Workflow:	12
4.4 Working Model:	12
4.5 Working of application:	13
5. Environment Health Check Toolkit.....	15
5.1 Introduction:	15
5.2 Main Files:.....	15
5.3 Work Flow of Environment Health Check:	16
5.4 Main Modules and their working:	18
4.5 Result and discussion:	19
6. EIT Tools and Utility Portal.....	20
6.1 Product Traceability Matrix	21
7. Result Analysis.....	23
7.1 Automate Nightly Run:	23
7.2 Environment Health Check	23
7.3 Product Traceability Matrix	26
8. Conclusion	28
8.1 Conclusion.....	28
8.2 Future Scope of work	28
9. References.....	29
10. Annexure.....	30
10.1 Algorithms/pseudo code.....	30
10.1.1 Product Traceability Matrix.....	30
11. Plagiarism Report	32

List of Figures

Figure 2.1 Retail Integration bus[5].....	4
Figure 2.2 Foundation Data	5
Figure 3.1: Working of Open Script topology	6
Figure 3.1: Synergy Plug-ins for Automation.....	8
Figure 4.1 Data in Automation config check table	10
Figure 4.2 Data in RIB.properties file	11
Figure 4.3 Batch File Sample Code	11
Figure 4.4: Working Model of Nightly Run	12
Figure 4.5 Mail generated	13
Figure 4.6 Mail if some applications are down (Specific).....	14
Figure 5.1 Working Model of Environment Health check	16
Figure 5.2 Launch pad view to user	17
Figure 5.3 Live execution of status tab	17
Figure 6.1 EIT Tools and Utilities Portal.....	20
Figure 6.2 Product traceability matrix home page.....	21
Figure 6.4 Product traceability matrix multiple sub folder selection page	22
Figure 7.1 Mail Generate I.....	23
Figure 7.2 Mail Generate II	24
Figure 7.3 Mail Generate Part III.....	25
Figure 7.4 Test cases count for each product.....	26
Figure 7.5 Product Matrix graphs	27

1. Introduction

1.1 Project Definition

Enterprise Integration Testing (EIT) Automation tool, utilities and portal that helps to reduce the manual work and it will provides the Health of the Environment so that it will improve the results of automation testing.

1.2 Motivation

The goal of our automation is to reduce the job of test engineers provide certain core functionalities. Which include monitoring, logging, functioning, reporting and finding the error, So that before delivering the product to the Customer functionality should work properly.

Now a days with growing demand of testing, capabilities and new features are added every time to time with the new releases of the software and there might be the chance of bug's so Regression testing is always preferable and it motivates me to carry out this by the Automation.

1.3 Overview of Projects

Enterprise Integration Testing (EIT) team is to test the data is flowing in different products (integration for products). That means they have to check whether data is flowing with different product correctly or not.

The initial stage is to gaining the depth and breadth knowledge and understanding of the different products at Oracle like Allocation, Retail Warehouse Management System (RWMS), Retail Invoice Matching (ReIM) and Store Inventory Management System (SIM). The next stage is to training of different type of tool and framework (Open Script and Synergy).

My first project on Environment status check task to automate the nightly run from different Desktop and Laptop. Firstly we have verify all the dependent product, database, Retail Integration bus and web services are working before running nightly scheduled scripts run so that it will improve the pass % of daily based scripts run on that automation based desktop and laptop.

Environment Health check toolkit this is my 2nd project. In this tool we provide input as wiki URL of the product, URL of the environment and list of emails id of our team so that at last the tool itself trigger the mail to the required user. Tool itself fetches the required information from wiki page and it will make the various property files as per the framework it require and then it will trigger script for environment application, Retail Integration bus, database and web services, After that it will send the health of different product, retail integration bus, database and web services via email to the required user.

Third project is to make the Product Traceability Matrix This Utility is used to know how many products are covered in particular test scenario for execution purpose. User should able to select a level1 folder for specific project after this able to select specific option from the following available options for like Sub folders, All Folders, Multiple Folders. Then we should get the table format of the results. Then generate charts (Bar chart and Pie chart) for the generated result.

2. Literature Survey

2.1 Retail Products

2.1.1 Retail Merchandising System (RMS)

Retail merchandising system (RMS), it is the main repository that stores and manages the information. The data integrity is ensured by RIB (retail Integration bus). Cost change, replenishment, item maintenance and inventory management are the functionality of RMS. Through these functionality we can provide easy access to the information. It is also able to take critical decisions and helps in achieving targets related to sale and profits. RMS is a heart of all products.

2.1.2 Store Inventory Management (SIM)

Store Inventory Management (SIM), it is used to track the inventory of the store, it will track the Inventory Management, ship of the item and receive the items accurately and efficiently. 3 tier architecture is used in this i.e. Server, client and Data. The Database Oracle 11g are consisted in Data tier, and the server tier is in J2EE application inside Web logic application server. This is a standalone application for hand held devices [3].

2.1.3 Warehouse Management System (WMS)

WMS manage the warehouse related transaction in the organization. It defines the warehouse types. Like picking area, block storage, high-rack storage etc. It will manage the information of goods at storage bin level. We can associate the data with products then we have to assign warehouse number to each storage [6].

2.1.4 Retail Invoice Matching (ReIM)

Retail Invoice Matching (ReIM) is responsible for creating and matching the invoices for the Return to Vendor (RTV) and purchase order (PO). It will verify the cost of the item and quantity of Items. If the quantity or cost of the item is not matching, then it can be review through ReIM [3].

2.1.5 Retail Price Management (RPM)

Oracle Retail Price Management (RPM), It is used for strategy based pricing for any item. It will assist and suggest you with pricing decisions. With RPM, you are able to control pricing strategy across the board and it also gives more predictable and profitable outcome [3].

2.1.6 RIB (Retail Integration Bus)

Retail Integration Bus (RIB), It supports the message transfer to other product with guaranteed delivery and also it transfer the message sequentially without any error. It has automatic retry, ability to fix message and also retry messages. It supports multi-threading and fast delivery of items SOH (Stock on hand) to the other product [3].

2.1.7 Allocation

Allocation can allocate items from many sources such as Warehouse, transfers and purchase order. Types of the allocation, such as What if, Standard and Scheduled Allocations. Through What if Allocation we can allocate infinite number of items to the different location. We can schedule the allocation through Scheduled allocation so that it will automatically re-create the allocation [3].

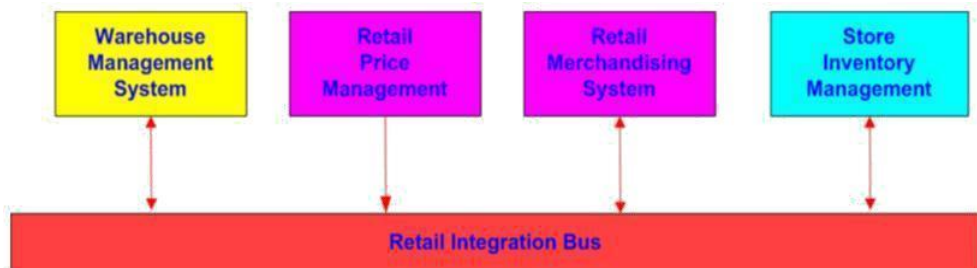


Figure 2.1 Retail Integration bus[5]

2.2 Foundation Data

Information about Organization hierarchy as well as merchandising hierarchy is required. Such as Warehouse (WH), Stores (ST), Items and Suppliers (SUPP) are required data for Our Customer. Details of Price of items, SOH(Stock On hand) of Item, Sales per day(SPD), Purchase Order (PO) detail, Transfers (S2S(Store to Store) or W2W(Warehouse to warehouse),W2S(Warehouse to store)) should be stored in the Product table with correct data[2].



Figure 2.2 Foundation Data

3. TECHNOLOGIES USED

3.1 Automation Tool's Survey

3.1.1 Open Script

Open Script provides an Eclipse-based environment that provides the development of automated test scripts. Using Java, we can develop our script. We can also use any Java API in Open Script [1].

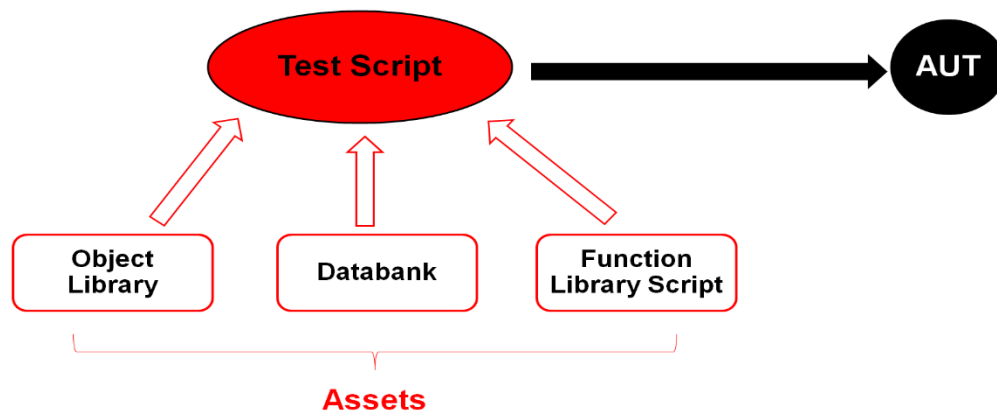


Figure 3.1: Working of Open Script topology

Assets Description:

1. Object Library:- These are .properties files. In this file we are storing Xpaths of the objects that is used to identify the object & perform operations on the object.
2. Databank:- Here we can add database as Databank or Comma separated value file (CSV). In Databanks, we are used to save the information like DB details.
3. Function Library:- We are writing functions in Function Library. In the test script we can add these functional library through the assets so that we can use the function inside the functional library the test script.

Guidelines:

1. We cannot delete or change any files is generated by Open Script when we are creating test script. For example manifest.mf, build.properties, script.properties, module.properties etc.

2. To run the script we need to create a test script. I.e. we cannot run from functional library script.
3. Try to use minimum number of assets in the individual script.
4. Avoid the cyclic reference of the functional library.
5. For saving of script we have to use save from tool options and "Save path relative to current script" should not be used.
6. Avoid move/copy scripts using file system because sometime it will create issue in the script.
7. If user wants to change the name the scripts, so they should change via Tool only.

Using Command Line:

We can run a script from command prompt using .bat file

In Command:

Run_Script.bat "ScriptPath/Script.jwg [ExtraArg]"

- a. Script Path is folder path where script is there. We have to give full path containing a forward slash (/) or double back slashes (\\).
- b. User can pass the extra argument also. For Example Iteration, Iteration Delay etc.

From Open Script Tool, We can export the setting also and after that we can pass the setting file path location as an argument.

Command to pass a property file:

run.bat "Script Path/Script.jwg properties file Path" "C:/playback.properties"

3.1.2 Synergy

Synergy tool, which is used to write the script for java based application.

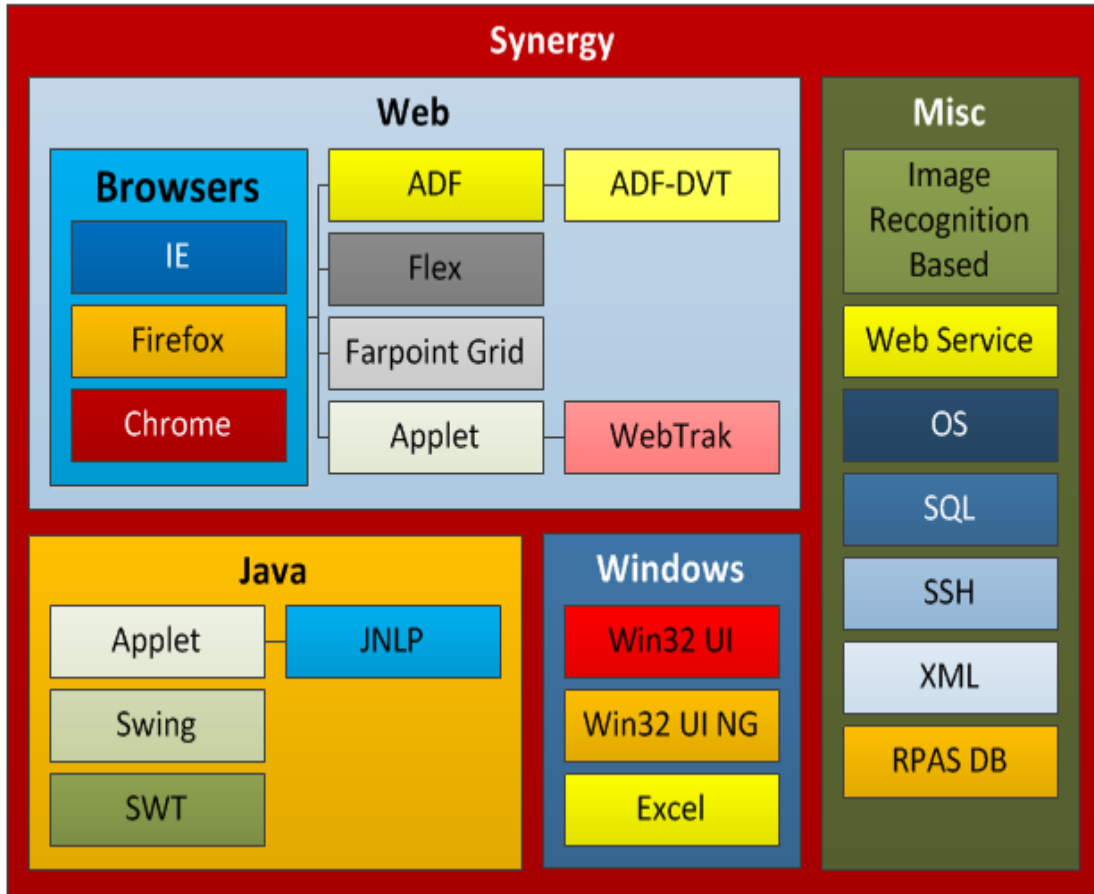


Figure 3.1: Synergy Plug-ins for Automation

We are writing the script in java script as well as Tabular format. For test case writing we can use only tabular format. For the keywords we can use tabular as well as java script.

Syntax of Tabular Format:

1. 1st column is a variable assignment.
2. If we are using 1st column as a variable assignment, then the 2nd column will be keyword invocation.
3. If we are using 1st column is a keyword invocation, then the 2nd column will be argument of keyword.
4. Rest all the columns are used for number of argument.
5. If we are using 1st column is a directive then the rest of the column are used for arguments for directive.

In Synergy, Firstly we have to create Test Suite which may contain the following things:

1. TestCases: Steps of code.
2. User Keywords:- like function or methods.
3. Resources: Collection the keywords (function or methods).
4. Variable_files: Test cases under test suite are used this variable files.
5. Global_Variables: Used to define the global variable inside test suite.
6. Configuration: Like adding libraries such as Excel, JavaScript, String Support Library, SQL etc.

Important Libraries:

SQL Library:

For connection to the database, we are importing SQL library, so that it will create one SQL instance and give connection.

Some keywords:

1. Execute: used for Insert and update.
2. Query: For select query

SSH Library:

For connection to any UNIX system and perform any commands we need to import SSH library, It uses remote machine's prompt for execution of the command.

Some keywords:

1. Open Connection: Open connection for individual host.
2. Login: We need to give username/password.
3. Write: Executed on the particular host
4. Close: Closing of connection

4. Automate Nightly Run

4.1 Introduction:

I have design to run nightly scheduled test scripts from individual remote desktop. If applications on which those scheduled scripts are dependent are up and running.

4.2 Additional Requirements:

4.2.1 Table Requirement:

EIT Database having “Automation_config_check” table with these many fields:

- **ID** –This is a Unique ID
- **Environment** – like 16.0 QA7, 16.0 QA8 and EIT 3.
- **HOST** – Laptop and Desktop where we are running automation.
- **MODULE** – List of product i.e. RMS- RWMS-SIM-RPM.
- **RUN_DATE** – Vdate from Period table.
- **RMS, RPM, RWMS, RI, SIM, ALLOCATION, RDE, RMS_WEB, RESA, RIB, IGS_WEB, REIM** – Products Health flag (Working or Not).

ID	ENVIRONMENT	HOST	MODULE	RMS	RWMS	SIM	RPM	RUN_DATE	ALLOCATION	RI	RDE	RESA	RMS_WEB	IGS_WEB	SIM_WEB	RIB	REIM	NIGHTLY_BATCH
86	EIT2	Blr2223131	(null)	DOWN	(null)	(null)	DOWN	11-NOV-17	DOWN	UP	UP	DOWN	UP	UP	UP	UP	DOWN	DOWN
85	EIT1	EIT_Auto_Prod-1	(null)	UP	UP	UP	UP	11-NOV-17	UP	UP	UP	UP	UP	UP	(null)	UP	UP	UP
84	EIT2	Blr2223131	(null)	DOWN	(null)	(null)	DOWN	10-NOV-17	DOWN	UP	UP	DOWN	UP	UP	UP	UP	DOWN	DOWN
83	EIT1	EIT_Auto_Prod-1	(null)	UP	UP	UP	UP	10-NOV-17	DOWN	UP	UP	UP	UP	UP	(null)	UP	UP	UP
82	EIT2	Blr2223131	(null)	DOWN	(null)	(null)	DOWN	09-NOV-17	DOWN	UP	UP	DOWN	UP	UP	UP	UP	DOWN	DOWN
81	EIT1	Blr2223131	(null)	UP	UP	UP	UP	09-NOV-17	DOWN	UP	UP	UP	UP	UP	(null)	UP	UP	UP
80	EIT2	Blr2223131	(null)	DOWN	(null)	(null)	DOWN	08-NOV-17	DOWN	UP	UP	DOWN	UP	UP	UP	UP	DOWN	DOWN
79	EIT1	EIT_Auto_Prod-1	(null)	UP	UP	UP	UP	08-NOV-17	DOWN	UP	UP	UP	UP	UP	(null)	UP	UP	(null)
78	EIT2	Blr2223131	null	DOWN	(null)	(null)	DOWN	07-NOV-17	DOWN	UP	UP	DOWN	UP	UP	UP	UP	DOWN	DOWN
77	EIT1	Blr2223131	null	UP	UP	UP	UP	07-NOV-17	UP	UP	UP	UP	UP	UP	(null)	UP	UP	UP

Figure 4.1 Data in Automation config check table

4.2.2 Constraint for table:-

If the INTEGRATION FLAG are set YES in RIB. Properties file then we check only to “Automation_config_check” table. Only one entry for individual environment and individual product per day should be there. If we are running multiple times then it would just updates the status.

```
Version=SAAS
env=eit1
integration_flg=No
module=RMS_RWMS
```

Figure 4.2 Data in RIB.properties file

4.2.3 Requirement for bat file:

We have design the bat file in such a way that it should run first the configuration script then it will call “checkNightlyRun.java” file instead of (Automation_executor.java) file, which make the required status check, after that it will call “Automation_Executor.java” file if status of all dependent products are working, it will sent mail with reason for not carrying out planned run.

```
IF NOT EXIST D:\EIT_Automation_Workspace\report\Execution_Report.xlsx (
xcopy /s /y D:\EIT_Automation_Workspace\Gold_EIT1_config*.xlsx D:\EIT_Automation_Workspace\master\
xcopy /s /y D:\EIT_Automation_Workspace\Gold_EIT1_config\input.properties D:\EIT_Automation_Workspace\master\

xcopy /y D:\EIT_Automation_Workspace\Gold_EIT1_config\RMS.properties D:\SVN_16_0\RMS\Environments\
xcopy /y D:\EIT_Automation_Workspace\Gold_EIT1_config\allocation.properties D:\SVN_16_0\Allocation\Environments\
xcopy /y D:\EIT_Automation_Workspace\Gold_EIT1_config\SIM.py D:\Users\cnihar\Synergy\Workspace\SIM\Environments\
xcopy /y D:\EIT_Automation_Workspace\Gold_EIT1_config\RMSweb.properties D:\SVN_16_0\RMS\Environments\
xcopy /y D:\EIT_Automation_Workspace\Gold_EIT1_config\IGSweb.properties D:\SVN_16_0\RMS\Environments\
xcopy /y D:\EIT_Automation_Workspace\Gold_EIT1_config\RFWeb.properties D:\SVN_16_0\RFM\Environments\
xcopy /y D:\EIT_Automation_Workspace\Gold_EIT1_config\RIB.properties D:\SVN_16_0\RMS\Environments\
xcopy /y D:\EIT_Automation_Workspace\Gold_EIT1_config\REIM.properties D:\SVN_16_0\ReIM\Environments\
xcopy /y D:\EIT_Automation_Workspace\Gold_EIT1_config\ReSA.properties D:\SVN_16_0\ReSA\Environments\
xcopy /y D:\EIT_Automation_Workspace\Gold_EIT1_config\RFM.properties D:\SVN_16_0\RFM\Environments\
xcopy /y D:\EIT_Automation_Workspace\Gold_EIT1_config\RWMS.properties D:\SVN_16_0\RWMS\Environments\
xcopy /y D:\EIT_Automation_Workspace\Gold_EIT1_config\SIM.properties D:\SVN_16_0\SIM\Environments\

javac -classpath *.;\Jars\commons-codec-1.10.jar;\Jars\commons-collections4-4.1.jar;\Jars\ojdbc6.jar;\Jars\poi-3.15.jar;
\Jars\poi-ooxml-3.15.jar;\Jars\poi-ooxml-schemas-3.15.jar;\Jars\xmlbeans-2.6.0.jar;\Jars\activation.jar;\Jars\mail.jar" Automation_Executor_V3.java
java -classpath *.;\Jars\commons-codec-1.10.jar;\Jars\commons-collections4-4.1.jar;\Jars\ojdbc6.jar;\Jars\poi-3.15.jar;
.\Jars\poi-ooxml-3.15.jar;\Jars\poi-ooxml-schemas-3.15.jar;\Jars\xmlbeans-2.6.0.jar;\Jars\activation.jar;\Jars\mail.jar" Automation_Executor_V3 standalone
)

IF NOT EXIST D:\EIT_Automation_Workspace\report\Execution_Report.xlsx (
del D:\EIT_Automation_Workspace\master\*.xlsx
del D:\EIT_Automation_Workspace\master\*.properties
xcopy /s D:\EIT_Automation_Workspace\gold*.xlsx D:\EIT_Automation_Workspace\master\
xcopy /s D:\EIT_Automation_Workspace\gold*.properties D:\EIT_Automation_Workspace\master\

javac -classpath *.;\Jars\** Automation_Reporter.java
javac -classpath *.;\Jars\commons-codec-1.10.jar;\Jars\commons-collections4-4.1.jar;\Jars\ojdbc6.jar;\Jars\poi-3.15.jar;
.\Jars\poi-ooxml-3.15.jar;\Jars\poi-ooxml-schemas-3.15.jar;\Jars\xmlbeans-2.6.0.jar;\Jars\activation.jar;\Jars\mail.jar;\Jars\ojdbc6.jar" CheckNightlyRun.java
java -classpath *.;\Jars\commons-codec-1.10.jar;\Jars\commons-collections4-4.1.jar;\Jars\ojdbc6.jar;\Jars\poi-3.15.jar;
.\Jars\poi-ooxml-3.15.jar;\Jars\poi-ooxml-schemas-3.15.jar;\Jars\xmlbeans-2.6.0.jar;\Jars\activation.jar;\Jars\mail.jar;\Jars\ojdbc6.jar" CheckNightlyRun %1
)
exit
```

Figure 4.3 Batch File Sample Code

4.2.4 Other requirements and constraints:

We have placed all the scripts in the same machine within the same folder containing all the properties file and commander are needed to be placed in the desktop to run our Environment health check. Every script either update and insert query, for status of product in “Automation_config_check” table in EIT Database. Product status check task is very dependent on flag INTEGRATION FLAG if Yes then only Database entry and scheduled check will occur.

4.3 Workflow:

Firstly, we need to run configuration scripts to check the products, retail integration bus and web services should make the entry to “Automation_config_check” table if INTEGRATION FLAG is set to YES. Then only we should run “checkNightlyRun.java” file will be called, which checks the status of only those products whether it is ON and running on which planned scripts depend. It will check by comparing product mentioned in module column of the table to their status in their respective product status. If all dependent product, retail integration bus and web services were working then only our planned run would start. If status are DOWN or not working then an email will be triggered with list of product, retail integration bus and web services, which are DOWN.

4.4 Working Model:

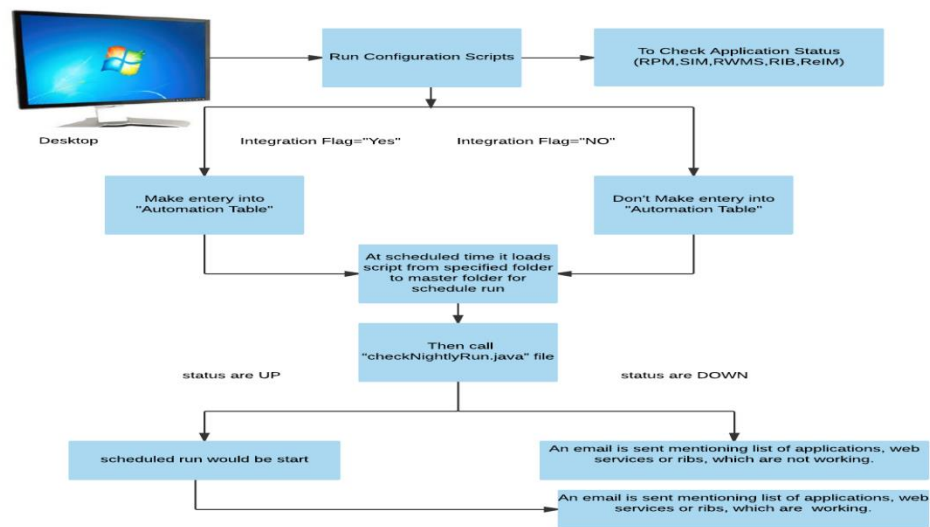


Figure 4.4: Working Model of Nightly Run

4.5 Working of application:

Firstly, the code will run the configuration test scripts for the individual environment, it will make entry into “Automation_config_check” table for today’s run status and then it will send mail for current status of products, web services and retail integration bus. Below the mail is shown with the module, which tells program which products, web services and retail integration bus are required to carry our planned run on that particular laptop or desktop. This will shows only some products, web services on which test script of that automation desktop actually depends. If any of module products, web services and retail integration bus are down then mail is sent to user notifies that due this products, retail integration bus and web services that day planned run will not occur and then user will verify issue manually and then they raise SR for fixing that issue to DSS(Development Support Services)team. Mail format is below.

Status	Execution Count
Passed	11
Failed	3
Timed Out	1
Not Started	0
Total Scripts	15

Property	Value
Run Started:	11-10-2017 06:30:08
Run Ended:	11-10-2017 08:46:39
Total Run Time:	0 days: 2 h: 16 m: 31 s
Automation Host:	EITAuto_Prod_1
Environment:	EIT1
Overall Execution Status	Completed

Functionality	Number of Stages	Manual Test Case Name	Overall Status	Duration(HH:MM:SS)	Stage 1 Results
RMS	1	RMS URL and DB Check	Timed Out	0 days: 1 h: 14 m: 53 s	Timed Out
RPM	1	RPM URL and DB Check	Pass	0 days: 0 h: 14 m: 46 s	Pass
ReIM - Invoice Match	1	ReIM URL and DB Check	Pass	0 days: 0 h: 2 m: 55 s	Pass
RWMS	1	RWMS URL and DB Check	Fail	0 days: 0 h: 14 m: 29 s	Fail
ReSA	1	ReSA URL and DB Check	Pass	0 days: 0 h: 1 m: 36 s	Pass
SIM	1	DB Check	Pass	0 days: 0 h: 0 m: 12 s	Pass
RIB	1	RIB_Check	Fail	0 days: 0 h: 5 m: 2 s	Fail
JOS_Process_Flow	1	URL_Check	Pass	0 days: 0 h: 0 m: 57 s	Pass
JOS_Scheduler	1	Jos_Scheduler_URL_Check	Pass	0 days: 0 h: 1 m: 20 s	Pass
POAM_Admin	1	POAM_URL_Check	Pass	0 days: 0 h: 1 m: 19 s	Pass
RDE	1	RDE_URL_Check	Pass	0 days: 0 h: 0 m: 16 s	Pass
RI	1	RI_URL_Check	Pass	0 days: 0 h: 0 m: 23 s	Pass
Allocation	1	Alloc_URL_Check	Fail	0 days: 0 h: 4 m: 34 s	Fail
RMS_Web_Services	1	RMS_Web_Service_check	Pass	0 days: 0 h: 8 m: 20 s	Pass
IGS_Web_Services	1	IGS_Web_Service_check	Pass	0 days: 0 h: 5 m: 28 s	Pass
List of Down RIBs					
rib-rms.Vendor_pub_1					
rib-rms.Season_pub_1					
rib-rms.Order_pub_1					

Figure 4.5 Mail generated

Test suite initiation failed due to following apps are down, RIBS, please raise SR and resolve it.

Property	Value
Server	msp52458
Environment	EIT1
Down Application	RIBS,
RMS	UP
RWMS	UP
SIM	UP
RPM	UP
ALLOCATION	DOWN
RI	UP
RDE	UP
RESA	UP
RMS_WEBSERVICE	UP
IGS_WEBSERVICE	UP
SIM_WEBSERVICE	null
RIB	DOWN
REIM	UP
RMS SCHEDULER(VDATE)	UP

Figure 4.6 Mail if some applications are down (Specific)

5. Environment Health Check Toolkit

5.1 Introduction:

This is an UI interface where we can manually give the wiki URL link for specified environment that will automatically fetch the required data for different products from the wiki page. It will fetch all the required information from wiki and it will create respective properties files for all the products like SIM, RMS, RPM, web services, RWMS etc., It will also create 3 commander files, so that our commander will pick these files, then check for all the product are working or not.

5.2 Main Files:

Environment Health Check Toolkit, Files used in this projects :-

1. Web_Content/**home.xhtml**- This is the main Page,
2. Web_Content/**app.properties**– It will Contains all properties file.
3. WebContent/tabView/**launchpadTab.xhtml** – Launchpad tab UI And provides fields to get some data from user.
4. WebContent/tabView/**liveExecutionTab.xhtml** –Live Execution UI. If some current run is in progress then it will populate.
5. Java Resources/view/**LaunchpadView.java** – This is called when we click submit button. After parsing wiki page, it will create the properties file and move to respective folder and after that it will create three commander files and move to the master folder.
6. JavaResources/view/**LiveExecutionController.java**–It will populate only if any runs are in progress and it will gets the data from below folder EIT_Automation_Workspace/report/ Exceution_Report .xlsx

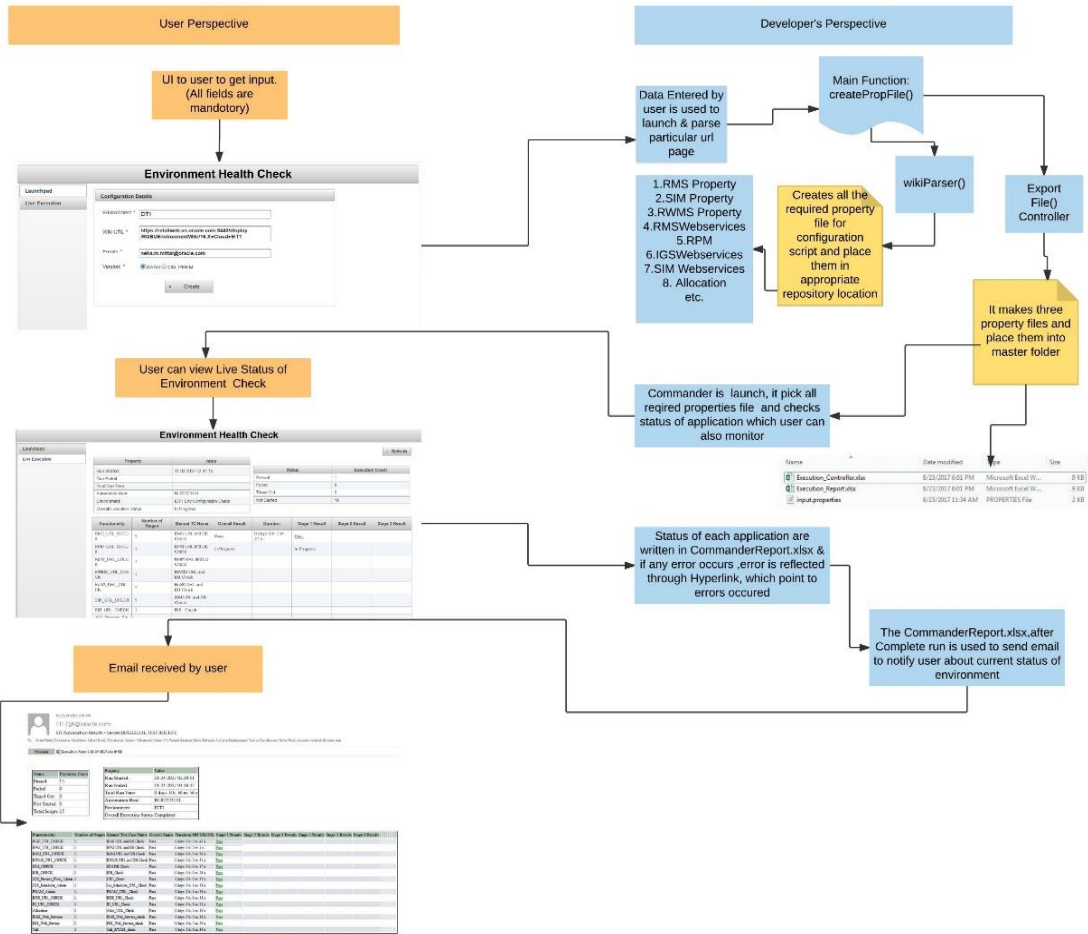


Figure 5.1 Working Model of Environment Health check

5.3 Work Flow of Environment Health Check:

User Perspectives: After launching this Environment Help check tool Portal that has 2 tabs In UI. 1st one is Launchpad, where we can provide wiki URL, Product of that wiki page and list of mail id of the users which is comma separated who should get the mail of the current status of different Product.

2nd one is Live Execution, after providing all details and click submit button, it will go to this tab. Where we can see the live execution report of the different product that are present in the wiki URL link.

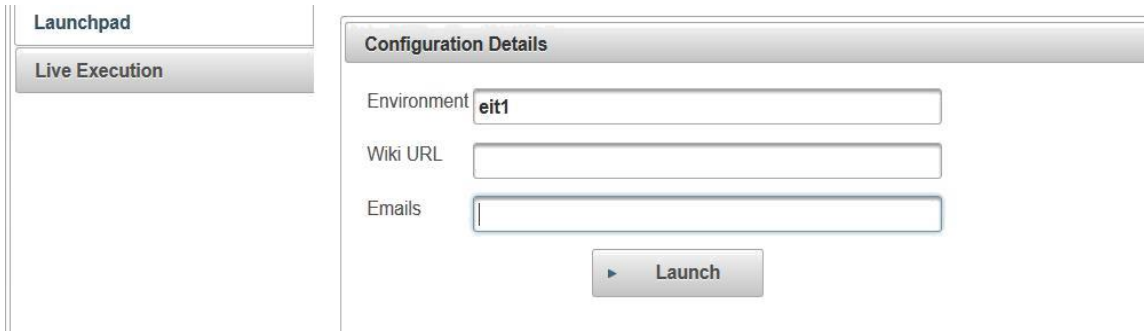


Figure 5.2 Launch pad view to user

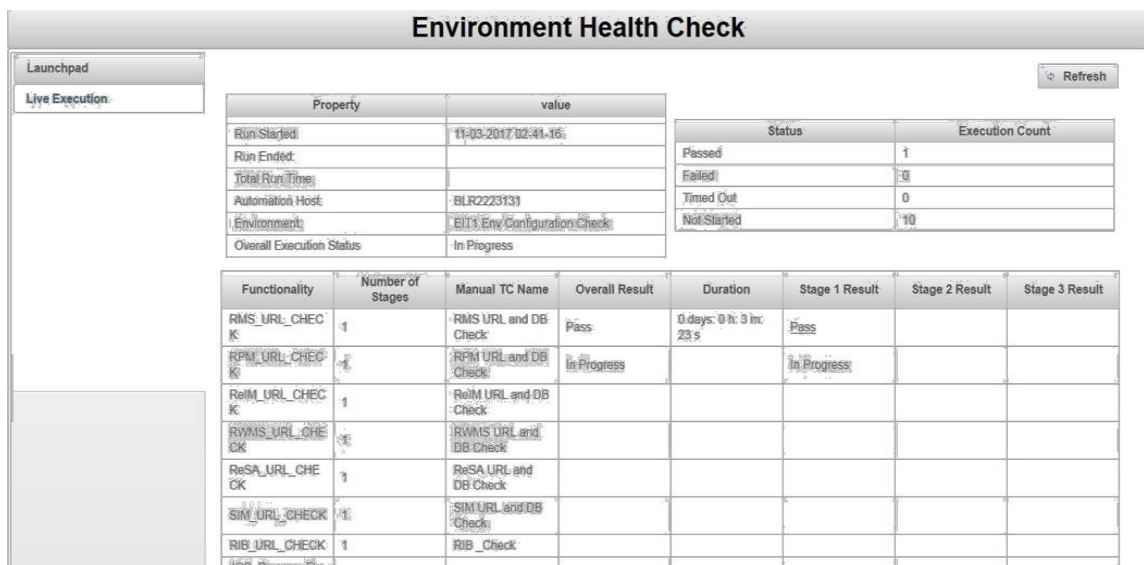


Figure 5.3 Live execution of status tab

Developer Perspectives:

After launching the portal, user have to give wiki link URL, environment and list of mail id separated by comma, after that click on submit button. The submit button called the `LaunchpadView.java/create_PropFile()` function. This will call the `wiki_parse ()` function to parse the information from wiki link page and then it will create/move product properties file to respective folders. This will also call `export_ControllerFiles()` function this will create 3 files and then it will move them to the master (Main) folder.

If the scripts run started so live Execution tab fetch information from report file i.e. Execution Reprot.xlsx. It will show the current run data to dashboard so that user can monitor live test scripts Run status.

5.4 Main Modules and their working:

User has a portal where they can provide product wiki page, environment wiki page url and mailing list with comma separated of people that wants the status of different product whether the Products are UP and running or not .In Parallel, Commander will place required configuration file in master file folder. At last It will trigger the mail with list of product with the health status with the particular environment.

The Main logic for fetching WIKI page of the environment and creating the properties file and then moving these required files are codes in **Java Resources/view /Launchpad View.java file.**

Library required –

JSOUP: Launch and fetch wiki link URL.

Important Functions in LaunchpadView.java file–

- a. Create_PropFile() – In this method which is called when we press the submit button on Launchpad from the portal. It will call function to generate all files. Such as wiki_Parser() and export_ControllerFiles().
- b. Wiki_Parser() – It uses JSOUP library to launch and fetch the details from Wiki URL. It fetches all information from wiki page and create the properties properties. It creates multiple properties files for multiple products.
 1. Input arguments: Wiki URL. Returns Boolean value True/false.
 2. Main variables used :
 - a. Inputval:– Product are present in app.properties file so that it will parse.
 - b. inputparam – of Type HashMap<String,LinkedHashMap<String,String>> , All properties for multiple product. Inner hash map store all different key-value pair for 1 product and are map to its product in outer hash map. We can store all product required fields are to be stored in this variable inputparam.

After number of Iteration of variable we can get all the required properties files.

- c. testScriptList – This is used to store all the product with comma separated are actually present on that wiki page. It is used to find commander test scripts from the master script i.e. ExecutionMasterScript.xlsx.

- d. export_ControllerFiles() – It will create three files for commander, first one is input.properties, Second one Execution_report.xlsx and the third one is Execution_Controller.xlsx. In Input properties file are created by copying EIT_Automation_Workspace \environments \System.properties file and the adding the mailing list to it and then it will move to master(main) folder. Rest 2 excel sheet is made to find the all test scripts name from this file i.e. ExecutionMasterScript.xlsx file.

4.5 Result and discussion:

It will create multiple properties file for all the different product automatically from wiki Link URL. Previously we have to create manually for all the different product for each of the environment. We need to create each environment file for different product because for running the test script it is required. After All this manual work for creating properties file for specific environment and moving to particular directory and after that we are creating commander files for each of the environment so that we can archive the task called “One Click Automation”.

6. EIT Tools and Utility Portal

Administration: Administration block contains various utilities like Password reset for Admin, Add User, Adding of New Connection and Admin selection for environment/products.

EIT Dashboard: EIT Dashboard used to analysis of results of various batches, analysis of the foundation data.

EIT Reports: It contains the generating reports for Product Traceability Matrix, Message Coverage, foundation data available and system options with comparison report.

EIT Tools: In This functionality which contains tools for Transaction validation, Data verification, POS Upload Generation and Configuration management.



Figure 6.1 EIT Tools and Utilities Portal

6.1 Product Traceability Matrix

This Utility is used to know how many products are covered in particular test scenario for execution purpose.

Functional Requirements:

User should able to select a level1 folder for specific project.

User should able to select a specific option from the following available options for sub-folders:

All Folders

Multiple Folders

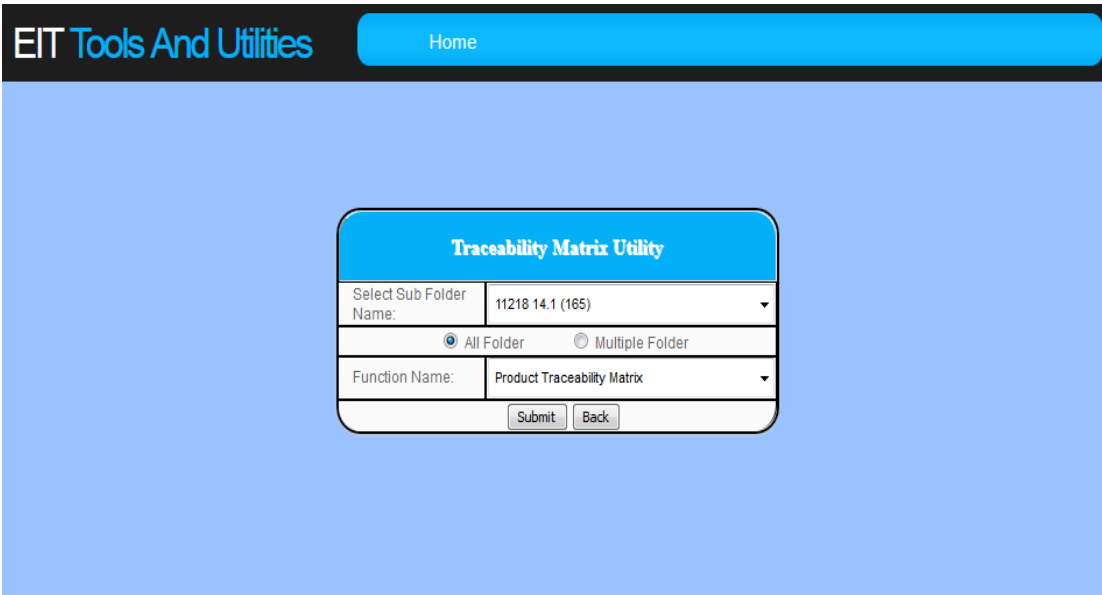
User should able to generate charts (Bar chart and Pie chart) for the generated result.

GUI Design



The screenshot shows the home page of the 'Traceability Matrix Utility'. At the top left, the text 'EIT Tools And Utilities' is displayed. To its right is a blue button labeled 'Home'. The main area of the page is light blue and contains a white rounded rectangle with a blue header that says 'Traceability Matrix Utility'. Below this header is a form with a label 'Select Folder Name:' followed by a dropdown menu showing '5802 New Release Planning'. A 'Submit' button is located at the bottom of the form.

Figure 6.2 Product traceability matrix home page



Fig

Figure 6.3. Product traceability matrix sub folder selection page

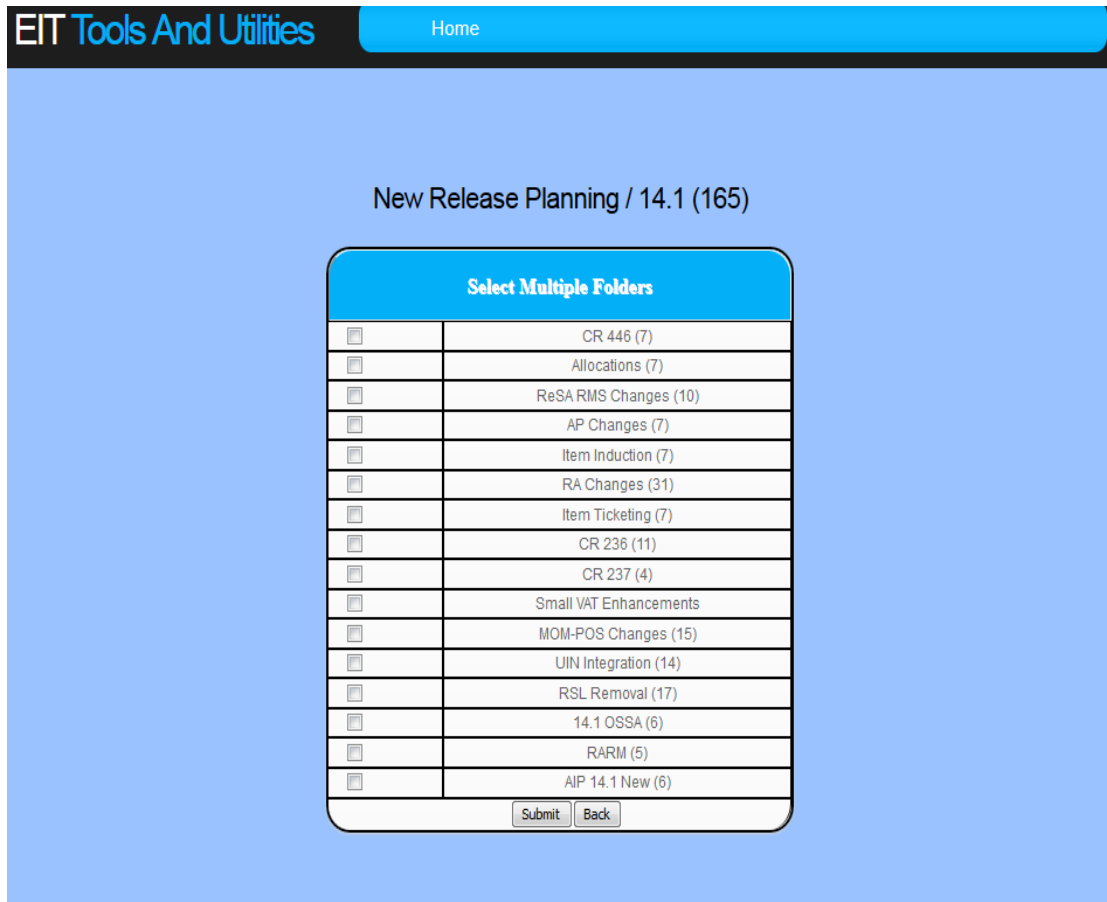


Figure 6.4 Product traceability matrix multiple sub folder selection page

7. Result Analysis

7.1 Automate Nightly Run:

- This application status check code provide an efficient way to increase the pass results for daily nightly run of test scripts that occurs from different automation server. This application runs the scheduled test scripts on particular automation box only when all the desired application modules and their DB are UP and working. This also help team to know current status of different applications. This helps team to know that if some scheduled scripts fail then there would error in test scripts not from application general failure. This functionality is added in various automation box, so that pass results of test scripts increases

7.2 Environment Health Check

1. This toolkit provides one click automation for environment health check, provides status for each applications.
2. It provides live execution for current status of application.
3. It send mail to user at end of run to provide complete status of each application.
4. This toolkit is designed to be used by our team for any new environment.
5. This toolkit will also be provided to other team.



Status	Execution Count
Passed	11
Failed	3
Timed Out	1
Not Started	0
Total Scripts	15

Property	Value
Run Started:	11-10-2017 06:30:08
Run Ended:	11-10-2017 08:46:39
Total Run Time:	0 days: 2 h: 16 m: 31 s
Automation Host:	EITAuto_Prod_1
Environment:	EIT1
Overall Execution Status	Completed

Figure 7.1 Mail Generate I

Functionality	Number of Stages	Manual Test Case Name	Overall Status	Duration(HH:MM:SS)	Stage 1 Results
RMS	1	RMS URL and DB Check	Timed Out	0 days: 1 h: 14 m: 53 s	Timed Out
RPM	1	RPM URL and DB Check	Pass	0 days: 0 h: 14 m: 46 s	Pass
ReIM - Invoice Match	1	ReIM URL and DB Check	Pass	0 days: 0 h: 2 m: 55 s	Pass
RWMS	1	RWMS URL and DB Check	Fail	0 days: 0 h: 14 m: 29 s	Fail
ReSA	1	ReSA URL and DB Check	Pass	0 days: 0 h: 1 m: 36 s	Pass
SIM	1	DB Check	Pass	0 days: 0 h: 0 m: 12 s	Pass
RIB	1	RIB_Check	Fail	0 days: 0 h: 5 m: 2 s	Fail
JOS_Process_Flow	1	URL_Check	Pass	0 days: 0 h: 0 m: 57 s	Pass
JOS_Scheduler	1	Jos_Scheduler_URL_Check	Pass	0 days: 0 h: 1 m: 20 s	Pass
POAM_Admin	1	POAM_URL_Check	Pass	0 days: 0 h: 1 m: 19 s	Pass
RDE	1	RDE_URL_Check	Pass	0 days: 0 h: 0 m: 16 s	Pass
RI	1	RI_URL_Check	Pass	0 days: 0 h: 0 m: 23 s	Pass
Allocation	1	Alloc_URL_Check	Fail	0 days: 0 h: 4 m: 34 s	Fail
RMS_Web_Services	1	RMS_Web_Service_check	Pass	0 days: 0 h: 8 m: 20 s	Pass
IGS_Web_Services	1	IGS_Web_Service_check	Pass	0 days: 0 h: 5 m: 28 s	Pass
List of Down RIBs					
rib-rms.Vendor_pub_1					
rib-rms.Season_pub_1					
rib-rms.Order_pub_1					

Figure 7.2 Mail Generate II

Test suite initiation failed due to following apps are down, RIBS, please raise SR and resolve it.

Property	Value
Server	msp52458
Environment	EIT1
Down Application	RIBS,
RMS	UP
RWMS	UP
SIM	UP
RPM	UP
ALLOCATION	DOWN
RI	UP
RDE	UP
RESA	UP
RMS_WEBSERVICE	UP
IGS_WEBSERVICE	UP
SIM_WEBSERVICE	null
RIB	DOWN
REIM	UP
RMS SCHEDULER(VDATE)	UP

Figure 7.3 Mail Generate Part III

7.3 Product Traceability Matrix

While selecting “All Folders” from the dropdown menu, It will directly show a test case count related to respective products as shown in the Fig. 4.6. If you will select multiple folders, it will show results for respective folders those are selected. While clicking on “Generate Chart” Button it will create bar and pie chart for generated result as shown in the Fig.7.2.



Figure 7.4 Test cases count for each product

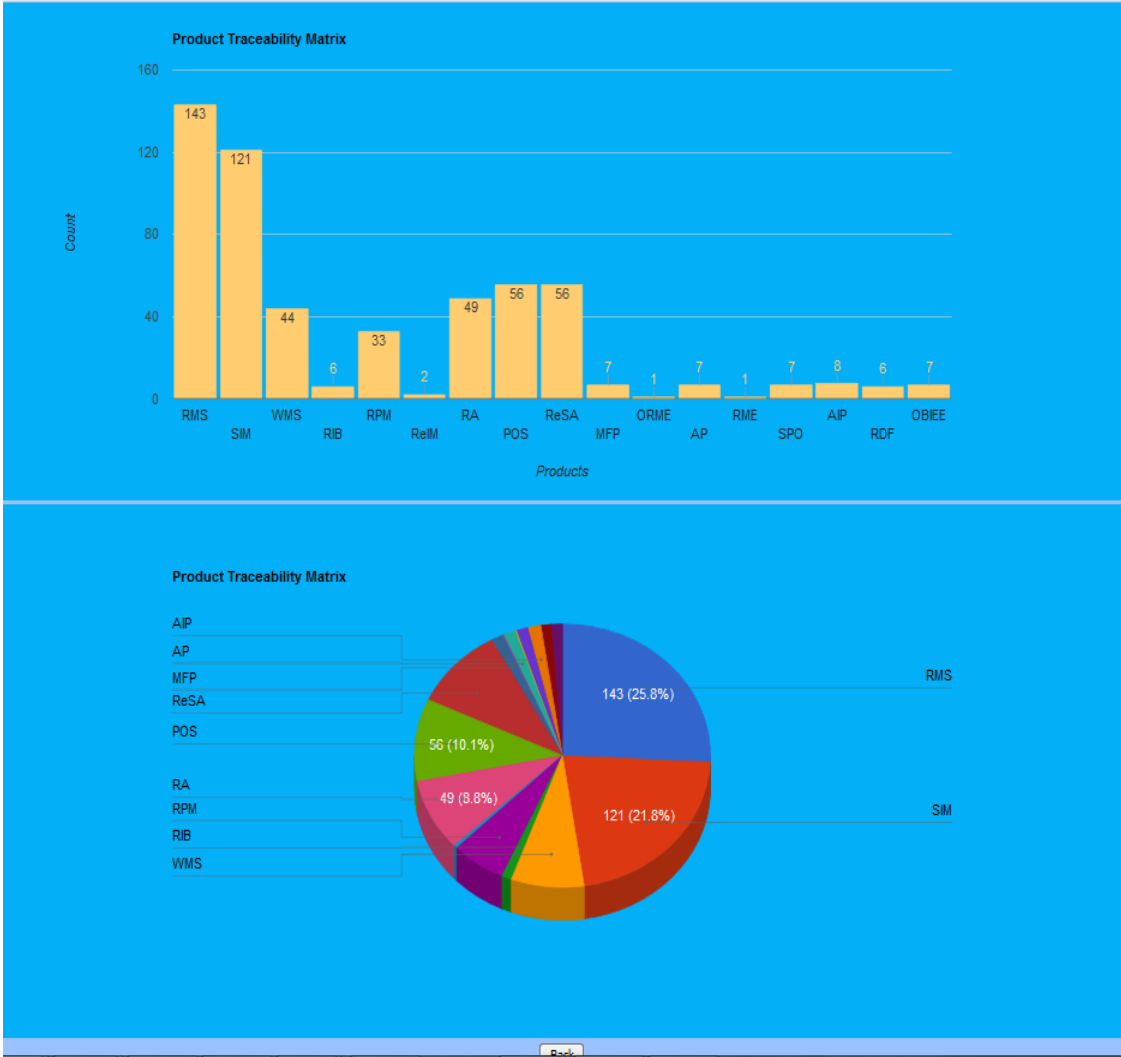


Figure 7.5 Product Matrix graphs

8. Conclusion

8.1 Conclusion

The utility toolkit, libraries and Test scripts provides the great help in EIT team for automation work. If we writing the scripts all are depends upon the toolkit mentions above.

Automation is not a one-time step it is ongoing commitment we have to write as much as script so that manual effort will reduce. And also it reduces the human error or we can say that it will make testing reliable so that we can give the quality of products to the customers.

My work is to support the automation work in EIT team so that they have to reduce the task by doing manually. And to provide the **One Click Automation** to my team.

I have also written the test case for our team to reduce the manual efforts to test the scenario.

8.2 Future Scope of work

To make the robust and dynamic of Environment health check Tool so the different team can use this. Already one of the team is using this.

Also automate the new set of test cases which is still executing by manually using synergy, open script and any other tool.

9. References

[1] **Elfriede Dustin, Thom Garrett, Bernie Gauf** *Implementing Automated Software Testing*. Addison-Wesley, USA, 2009.

[2] **Oracle Retail**. *Oracle functional testing*, 2010.

[3] **Oracle Retail**. *Oracle retail merchandising system*, 2012.

[4] <http://www.oracle.com/us/products/applications/retail/merchandise-management/merchandising-system/overview/merchandising-system-overview-1567847.html>

[5] https://docs.oracle.com/cd/E25294_01/doc.920/e15487/chap1_fm.htm#autoId2.

[6] <https://www.oracle.com/industries/retail/products/supply-chain-planning/warehouse-management/index.html>

10. Annexure

10.1 Algorithms/pseudo code

10.1.1 Product Traceability Matrix

- 1) while database connection do
- 2) if select the parent folder_id from OTM Database
- 3) while Store all the folder_id from parent_folder
- 4) end while
- 5) while select folder_id from folder_id[i]
- 6) Store the testcase_id and testcase_name from this folder_id
- 7) while select the testcase_id from the testcase_id[i]
- 8) while select the product_name
- 9) if testcase_name.contains(product_name)
- 10) product_count_name[++]
- 11) end if
- 12) end while
- 13) end while
- 14) end if
- 15) else if select multiple parent folder_id
- 16) if select the parent folder_id
- 17) while Store all the folder_id from this parent folder
- 18) end while
- 19) while select folder_id from folder_id[i]
- 20) Store the testcase_id and testcase_name from this folder_id
- 21) while select the testcase_id from the testcase_id[i]
- 22) while select the product_name
- 23) if testcase_name.contains(product_name)
- 24) product_count_name[++]
- 25) end if

```
26)             end while
27)             end while
28)         end if
29)     else if select all parent folder_id
30)         while select the parent folder_id from folder_id[]
31)             while Store all the folder_id from this parent folder
32)                 end while
33)                 while select folder_id from folder_id[i]
34)                     Store the testcase_id and testcase_name from this folder_id
35)                     while select the testcase_id from the testcase_id[i]
36)                         while select the product_name
37)                             if testcase_name.contains(product_name)
38)                                 product_count_name[]++
39)                             end if
40)                         end while
41)                     end while
42)                 end if
43)             end while
```

Neha Mittal

ORIGINALITY REPORT

1 %	1 %	0 %	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	scholar.lib.vt.edu Internet Source	1 %
2	ethesis.nitrkl.ac.in Internet Source	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		