

Automation Testing of the Retail Merchandising System

Submitted By
Shailey Bhardwaj
15MCEN03



DEPARTMENT OF INFORMATION TECHNOLOGY
INSTITUTE OF TECHNOLOGY
NIRMA UNIVERSITY

AHMEDABAD-382481

May 2017

Automation Testing of the Retail Merchandising System

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in CSE (NT)

Submitted By

Shailey Bhardwaj

(15MCEN03)

Guided By

Prof. Ajaykumar Patel



DEPARTMENT OF INFORMATION TECHNOLOGY

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2017

Certificate

This is to certify that the major project entitled “**Automation Testing of the Retail Merchandising System**” submitted by **Shailey Bhardwaj (Roll No: 15MCEN03)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Information Technology of Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this project part-II, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Ajaykumar Patel
Guide & Assistant Professor,
CE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Gaurang Raval
Associate Professor,
Coordinator M.Tech CSE (NT)
Institute of Technology,
Nirma University, Ahmedabad

Dr. Madhuri Bhavsar
Professor and Head,
IT Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Alka Mahajan
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Shailey Bhardwaj**, Roll. No. **15MCEN03**, give undertaking that the Major Project entitled “**Automation of the Retail Merchandising System**” submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **CSE (NT)** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Signature of Student

Date:

Place:

Endorsed by
Prof. Ajaykumar Patel
(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Prof. Ajaykumar Patel**, Assistant Professor, Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Madhuri Bhavsar**, Hon'ble Head of Information Technology Department, Institute of Technology, Nirma University, Ahmedabad for her kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr. Alka Mahajan**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation she has extended throughout course of this work.

I would also thank the Institution, all faculty members of Information Technology Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- **Shailey Bhardwaj**
15MCEN03

Abstract

One of the verticals of Oracle is in the retail industry that helps users to perform tasks in all manners from receiving merchandise from Suppliers or Partners to delivering those merchandise or services to end users of the industries, customers. We work on one of the products used in Oracle Retail called Retail Merchandising System (RMS). RMS is a large product and needs a lot of man power to test the application. Also, it becomes difficult for the customers to test the application manually. Automation testing is widely used in the industries nowadays because of its countable advantages over manual testing. So Oracle decided to provide automation testing of RMS to the customers for their ease. Also, with the emerging popularity of cloud, Oracle has also decided to provide RMS Automation Testing on cloud as Software as a Service (SaaS).

Abbreviations

| | |
|-------------|-------------------------------------|
| RMS | Retail Merchandising System. |
| OFB | Oracle Flow Builder. |
| ReIM | Retail Invoice Management. |
| RPM | Retail Price Management. |
| SIM | Store Inventory Management. |
| EBS | E-Business Suite. |
| OATS | Oracle Application Testing Suite. |
| ADF | Application Development Framework. |
| ROQ | Replenishment Order Quantity. |
| API | Application Programming Interface. |
| IDE | Integrated Development Environment. |
| OTM | Oracle Test Manager. |

Contents

| | |
|---|------------|
| Certificate | iii |
| Statement of Originality | iv |
| Acknowledgements | v |
| Abstract | vi |
| Abbreviations | vii |
| List of Figures | xi |
| 1 Introduction | 1 |
| 1.1 Introduction to RMS | 1 |
| 1.2 Automation Testing | 2 |
| 1.2.1 Automation Objectives | 2 |
| 1.3 Problem Definition | 3 |
| 2 Literature Survey | 4 |
| 2.1 Testing Automation | 4 |
| 2.2 Automation Advantages | 4 |
| 2.3 Techniques | 5 |
| 2.3.1 OpenScript :- | 5 |
| 2.3.2 Oracle Flow builder :- | 6 |
| 2.3.3 Selenium :- | 6 |
| 3 Introduction to RMS - Business Process | 7 |
| 3.1 Foundation Data - Organizational Hierarchy Overview | 7 |
| 3.2 Foundation Data - Merchandise Hierarchy Overview | 8 |
| 3.3 Item Maintenance | 9 |
| 3.4 Purchase Orders | 10 |
| 3.5 Deals | 10 |
| 3.5.1 Fixed Deal | 11 |
| 3.5.2 Complex Deal | 11 |
| 3.6 Replenishment | 13 |
| 3.6.1 Replenishment Methods | 14 |
| 3.7 Contracts | 16 |
| 3.7.1 Contract Types | 16 |
| 3.7.2 Ordering using Contracts | 17 |

| | | |
|----------|---|-----------|
| 4 | OpenScript | 18 |
| 4.1 | Introduction to OpenScript | 18 |
| 4.2 | Installation | 18 |
| 4.2.1 | System Requirements | 18 |
| 4.2.2 | Certified System Configuration | 19 |
| 4.2.3 | Environment Requirement | 20 |
| 4.2.4 | Software and Versions | 20 |
| 4.2.5 | Installing the Accelerator Kit | 21 |
| 4.2.6 | System Configuration | 23 |
| 4.3 | Features of OpenScript | 24 |
| 4.4 | Automation using OpenScript | 24 |
| 4.4.1 | Running the tests | 25 |
| 4.5 | Result Validation and Dashboards | 26 |
| 4.5.1 | Validating the results | 26 |
| 4.5.2 | Dashboard | 28 |
| 4.5.3 | End Point and Purge | 29 |
| 4.6 | Analysis | 29 |
| 4.6.1 | Advantages | 30 |
| 4.6.2 | Disadvantages | 30 |
| 4.6.3 | Conclusion | 30 |
| 5 | Oracle Flow Builder | 31 |
| 5.1 | Introduction to OFB | 31 |
| 5.2 | Generating Executable Scripts through OFB | 32 |
| 5.3 | Analysis | 34 |
| 5.3.1 | Advantages | 34 |
| 5.3.2 | Disadvantages | 35 |
| 5.3.3 | Conclusion | 35 |
| 6 | Selenium | 36 |
| 6.1 | Introduction to Selenium | 36 |
| 6.2 | Test Automation With Selenium | 36 |
| 6.3 | Analysis | 37 |
| 6.3.1 | Advantages | 37 |
| 6.3.2 | Disadvantages | 38 |
| 6.3.3 | Conclusion | 38 |
| 7 | Customized Datasheet Creation | 39 |
| 7.1 | Introduction | 39 |
| 7.2 | Current vs. New Automation Framework | 40 |
| 7.3 | Datasheet Format | 41 |
| 7.4 | Download Data | 43 |
| 7.4.1 | Batch File Creation | 44 |
| 7.4.2 | Reverse Engineering Approach | 45 |
| 7.5 | Outcome | 46 |
| 7.6 | Advantages | 47 |
| 8 | Future Work | 48 |

List of Figures

| | | |
|-----|--------------------------------------|----|
| 3.1 | Replenishment Summary | 14 |
| 4.1 | The OpenScript Interface | 19 |
| 4.2 | Repository Paths | 22 |
| 4.3 | EBS/Forms Enablement | 23 |
| 4.4 | Automation Process Flowchart | 25 |
| 4.5 | RMS Launcher | 26 |
| 4.6 | Result Report | 28 |
| 4.7 | Result Dashboard | 28 |
| 5.1 | OFB Overview | 32 |
| 5.2 | OFB Component Code | 33 |
| 5.3 | OFB Flow Tree | 33 |
| 5.4 | Generate OFT Script in OFB | 34 |
| 6.1 | Selenium Suite Overview | 37 |
| 7.1 | Datasheet Process Flow | 40 |
| 7.2 | Current vs. New Automation Framework | 41 |
| 7.3 | Sample Datasheet Format | 42 |
| 7.4 | Datasheet For Items | 44 |
| 7.5 | Batch File | 45 |

List of Tables

| | | |
|-----|--|----|
| 4.1 | Certified System Configuration | 20 |
| 4.2 | Software Required for OpenScript | 21 |
| 4.3 | Validation Area and Description | 27 |
| 4.4 | Description of Dashboard | 28 |

Chapter 1

Introduction

Retail industry is all about buy in bulk, break the bulk and then sell. Retail activities include

1. Planning : Planning of suppliers, products like how many products will be needed, which suppliers will be available to supply the products, how the products will be supplied from warehouse to store etc.
2. Buying : This activity includes buying of the products from supplier after planning.
3. Merchandising and Inventory Management : This activity includes procurement and inventory management at store level.
4. Selling and Customer Service : This includes selling of the products at store level and provide services to customers.
5. Support : Includes support for the customers.

1.1 Introduction to RMS

Retail Merchandising System is a completely integrated retailing solution that covers the core processes of a retail company. The system provides retailers with a complete end-to-end solution and is the most comprehensive and integrated solution for global retailing. The system endeavors to model the full “Value Chain”, all the links in the logistics pipeline from consumer to vendor.

RMS is the backbone of the MOM products. It is the central repository of all the MOM applications. It tracks inventory, items, sales, cost information and locations of the

items. In RMS, all the data about items and purchase order is stored. When the purchase order is created the invoice will be generated in RMS. This invoice will be matched with invoice generated in ReIM. If both are matching properly then allocation process will be started. In allocation process, item and PO data will be passed to SIM. From that the needs of store will be identified and right quantity for the items will be decided. The right price for the item will be decided with the help of RPM.

1.2 Automation Testing

In automated testing, automated tools run the test cases that repeat the predefined actions, comparing program's expected outcomes with the actual outcomes. If actual results do not match with the expected outcomes then there is some issue that needs to be addressed. Automated testing is better when the project is large and there are many user forms that need to be filled up. Also it requires less men power and resources. Possibility of the uncovered bug will be very less. Also we can reuse the test cases for the similar type of test data.

1.2.1 Automation Objectives

The various benefits and objectives of automation have been listed below.

1. Environment Validation The automation scripts validate the environment for stability so that all activities can commence. This can be a new migration or implementation or a patch. Once the scripts run successfully with the desired results across the features the system can be deemed stable and ready to be used and loaded with production data.
2. Regression Validation Automated Regression Testing is the testing area where automation can be leveraged to the maximum extent. All the previously executed test cases can be executed on the new build via automation. This helps in test accuracy along with confidence given that the product is working as expected. Automation also helps us in time saving for the regression cycle.
3. Patch Validation Patch validation is often time consuming as previously working features could be broken due to the new fix delivered. Automation here delivers huge savings in terms of time and testing. Retailers can either run the pre defined

accelerator kit or run their customized datasheets which encompass all their critical business work flows. Here the expectation is for all the regression scenarios to work even after the new patch is taken without breaking previously working features.

4. Customizations The sample pre defined datasheets have a US Hierarchy and fashion items which are taken through various transaction flows. These can be customized to suit the retailer's needs.
5. Time Savings The scripts offer significant time savings along with offering executable repeatable tests.

1.3 Problem Definition

In manual testing, testers and developers generates test cases, runs the system manually, try to find out defects in the system by comparing expected outcomes with the actual outcomes. Manual testing is better for the smaller size systems. Also cost is lower and it is more likely to find real user issues. But at the same time manual testing is repetitive and time consuming. Also there are some tasks which are difficult to test in manual testing. For example low level interface regression testing. This kind of testing is extremely difficult and is prone to mistakes and oversight when done manually. Also manual testing requires skilled and experienced staff to cover all type of errors. Oracle RMS is a large application and needs a lot of man power to test the application. So to deal with all these kind of problems, automated testing is required.

As cloud is emerging with its plenty of advantages, clients are also moving towards the cloud solution. Oracle has its own cloud and many products of the Oracle are available on cloud. Oracle has also decided to move automation of RMS on the cloud. The on prem solution for the automation needs to query the database for retrieving the values. But on cloud, the database access will not be available. So the problem is to find out the solution through which we can ensure that our test script has successfully passed without querying the database.

Chapter 2

Literature Survey

2.1 Testing Automation

Running your Oracle Retail Merchandising System (RMS) at its optimum efficiency means having a consistent patching strategy and migrating data quickly to those patches while still maintaining stable environments and minimizing downtime. In the past, testing new patches, environments and migrated data has been very time consuming, leading to long downtimes or skipped patches. Retailers need a tool that is highly scalable, flexible, modular, and efficient to quickly test their RMS application and environment for errors prior to deploying to their production environment.

2.2 Automation Advantages

- Scalable and Modular : The RMS Automated Testing tool allows you to add as many records as you need with which to test across any number of scenarios. Your testing can span various technology platforms (including forms and ADF) and can encompass your end-to-end merchandising solution or just a single functional area
- Flexible : Automation scripts validate the environment for stability, ensuring that all activities, such as a new migration, implementation, or a patch can be executed. Once the scripts run successfully with the desired results across all features, the system can be deemed stable and ready to be used and loaded with production data.

Automated Regression Testing is the testing area in which you can leverage automation to the fullest extent. You can execute all previously executed test cases

on the new build through automation. This ensures that testing is accurate and provides confidence that the product is working as expected. Automation also yields significant time savings for the regression cycle.

Patch validation is often time consuming, since previously working features can be broken because of a new fix. Automation delivers huge savings in terms of time and testing. Retailers can either run the pre-defined accelerator kit or run their customized datasheets that encompass all their critical business work flows. With automation, you can expect that all the regression scenarios will work even after the new patch is applied, without breaking previously working features.

- **Efficient** : Quality assurance (QA) professionals have been required to spend numerous hours or days testing a new patch depending on its size and scope. Automated testing provides the QA team with a time-saving tool that allows them to focus on other core business activities. The time saved with automation can free up resources and take the squeeze out of deadlines. The scripts and customizable datasheets offer significant time savings along with executable repeatable tests.

2.3 Techniques

2.3.1 OpenScript :-

OpenScript is a great tool developed by Oracle to support automation of Oracle Application Testing Suite. It supports Oracle EBS/Forms, ADF, JD Edwards, Siebel and also Web applications. It has great in-built report generation functionality which can give step by step execution details. It is based on record and playback functionality. Developers can record actions and then can transform those actions into executable test scripts which can be reusable.

- **User Interface** : Oracle ADF
- **Business Logic** : PL/SQL
- **Database** : Oracle DB

2.3.2 Oracle Flow builder :-

OFB is a keyword driven application tool developed by Oracle in which we have many in-built components for automation of the application. We can generate flows by arranging the components in sequence and then flows can be transformed into executable test scripts and can be executed with OpenScript.

- User Interface : Oracle ADF
- Business Logic : PL/SQL
- Database : Oracle DB

2.3.3 Selenium :-

Selenium is an open source tool which is widely used for automation of web applications. It is faster than OpenScript and OFB. It has support for many browsers and Operating Systems. Also test scripts can be written in many programming languages i.e. java, csharp, python, ruby, php etc. It is open source so we can modify the source code as per the requirement. It is very flexible and easy to use.

- User Interface : Oracle ADF
- Business Logic : PL/SQL
- Database : Oracle DB

Chapter 3

Introduction to RMS - Business

Process

The following functional areas exist in the Oracle Retail Merchandising System (RMS). Each functional area has business processes designed to help you complete a task.

- Cost Management
- Financial Management
- Foundation Data
- Inventory Control
- Item Maintenance
- Purchasing
- Replenishment
- System Administration
- User Tools

3.1 Foundation Data - Organizational Hierarchy Overview

The organizational hierarchy shows the structural hierarchy of business perspective. There are six levels of organizational hierarchy. Flexible Hierarchy structure and setup

options to accommodate multiple businesses and companies. It is the structure that enhances retailers ability to target selling outlets. Location grouping tools are used to accommodate cross hierarchy similarities.

The following are the six levels of Organizational Hierarchy in RMS:

1. Company : The highest merchandise and organizational entity defined in RMS. Only one company can be defined.
2. Chain : The next level after the company level in the organizational hierarchy. The definition of a chain is based on the needs of the company, but a chain can be used to group various store formats, concepts, and geographical locations within the organization.
3. Area : The next level after the company level in the organizational hierarchy. The definition of an area is based on the needs of the company, but an area is used typically to define a geographical group within the organization. An area can belong to only one chain.
4. Region : The next level after the company level in the organizational hierarchy. The definition of a region is based on the needs of the company, but a region can be used to group geographical locations within the organization. A region can belong to only one area.
5. District : The next level after the company level in the organizational hierarchy. The definition of a district is based on the needs of the company, but a district is used typically to group geographical locations within the organization. A district can belong to only one region.
6. Channel : A channel is different modes to sell the items i.e. Internet, Bricks and Mortar, Telephone, Catalogues etc. We can assign channel to locations while creating virtual warehouse or store.

3.2 Foundation Data - Merchandise Hierarchy Overview

The merchandise hierarchy is the classification of products at different level. It is used for easy management and tracking purpose. There are six levels of merchandise hierarchy.

We can create preferred structure as per the project requirement so customizations are supported.

Item Attributing Management at merchandise hierarchy levels is done to improve data consistency thus improving analysis and operations.

The following are the six levels of Merchandise Hierarchy in RMS:

1. Company : The highest merchandise and organizational entity defined in RMS. Only one company can be defined.
2. Division : The first level of merchandise hierarchy within an organization. The division is used to signify the overall type of merchandise that a retailer offers, such as hardlines or apparel.
3. Group : The second level in the merchandise hierarchy is Group. A division can have multiple groups.
4. Department : The third level after Group is Department. A group can have multiple departments. Key information about how inventory is tracked and reported is stored at the department level.
5. Class : The fourth level below department in the merchandise hierarchy is Class. A department can have multiple classes. A class provides the means to group products within a department.
6. Subclass : The fifth level after class in the merchandise hierarchy is Subclass. A class can have multiple subclasses. A subclass provides the means to classify products within a department/class combination.

3.3 Item Maintenance

The Item Maintenance information can be accessed through these procedures:

1. Set Up Diffs : RMS allows you to create different types of items. Differentiators are set up while creating the items. Differentiators are used to differentiate the items based on different attributes like size, colour etc. Diffs are used to distinguish items by their characteristics.

2. Create Items : There are six types of items in RMS. Regular, Pack Items, Deposit Items, Consignment/Concession, Catch weight and Transformable. Child items can be created for parent items. Supplier, location details are mandatory while creating the items.
3. Submit an Item for Approval
4. Reclassify Items : This is the process of moving an item or item list from one department or class or subclass to the another.

3.4 Purchase Orders

RMS is the central system for maintaining POs, this ensures visibility throughout the retail organization. Key Purchase Order Elements are Supplier, Delivery Date, Items, Costs, Quantities and Locations. All items in one PO must be from the same supplier. Single Item or Pack Item can be entered. No limit on the number of items or delivery locations on a purchase order. POs can be created multiple ways:

- Manual Creation : New Order can be generated or Order from an Existing PO can be created.
- Automated : Automated orders can be created through Store Orders, Replenishment or Contracts.
- Vendor Managed Inventory (VMI) : Order is created via supplier.

3.5 Deals

Deals are a set of discounts and/or rebates negotiated with a supplier that share a common start and end date. A deal provides the means by which retailers can manage reductions in the supplier's purchase order cost of an item. Following type of deals are available in RMS. Deal contains Supplier or Deal Partner, Status, Active and Close Date, Billing Type, Recalculation Indicator and Security Indicator. A deal partner, such as a manufacturer, distributor or wholesaler, that gives rebates to the retailer. Deal partners can be identified for each item/supplier/origin country combination. These deal partners can be named dynamically by each client using the dynamic hierarchy dialog. Deals can be Annual,

Promotional or PO-specific. In Annual deal no end date is required. It can be closed manually by user or automatically upon creation of a new annual deal for the supplier. Promotional deal is automatically closed on specific end date. In PO-specific deal start date is set to the order's not-before-date. No end date is required.

Deals can be created and edited in worksheet status. Once entered a deal must be submitted and approved via batch before it becomes active. Deals must be approved prior to start date or it becomes inactive. Deals can be manually closed or deactivated (system will set start date to current date). Deals can be purged after specified number of days (system option). The Recalculate orders indicator, if checked, will recalculate any approved, non-received orders that fall within the deal dates according to the new deal discounts. Deals are applied online by clicking the Apply Process button in the Purchase Order dialog. Then check the Apply Deals box in the Recalculation Options form. Deals will be applied in the nightly batch regardless if the user has hit the Apply button in the Ordering dialog. The discount can be applied when the PO is approved or when the PO is received.

3.5.1 Fixed Deal

In some cases the vendor will give a fixed amount of money to the buyer for promotional reasons, gondola placing or shelf space. Sometimes it is possible to tie these fixed values to a specific item category (merchandise hierarchy), but very often it is not possible.

Fixed deals will allow you to specify if the received money can be tied back to a specific category and as such be included in margin calculations (outside of RMS) or not.

3.5.2 Complex Deal

1. Off invoice : In some cases the vendor will give a fixed amount of money to the buyer for promotional reasons, gondola placing or shelf space. Sometimes it is possible to tie these fixed values to a specific item category (merchandise hierarchy), but very often it is not possible. Fixed deals will allow you to specify if the received money can be tied back to a specific category and as such be included in margin calculations (outside of RMS) or not.

Non Merchandise : If the fixed deal can not be traced back to a specific group of merchandise, the user will not check the merchandise indicator. This will require the user to enter the non merchandise code. These codes are held in RMS and

interfaced to Invoice Matching. They allow invoice matching to map the income to the correct accounts in the general ledger. Such a code could be Store Opening Offer or Supplier pick up.

A store opening offer is a one time offer that some vendors pay the retailer to stock the new store. In such a case it is difficult to assign the money specifically to a merchandise group. Supplier Pick up may be a special case where your delivery truck for fast delivery picked up merchandise from the supplier's warehouse.

Merchandise : In case the deal income can be related to a category, the user can check the merchandise checkbox and add merchandise and location hierarchy information by using the merchandise button.

Merchandise and Organizational information needs to be added so that RMS can post the income to the correct General Ledger accounts.

2. Bill Back : This is a deal set up by the buyer to receive money back from the supplier calculated on top of the off invoice price. This is often done because the supplier has a fixed sales price and does not want to lower the off invoice price for compliance reasons or price protection reasons. Very often this is a straight percentage or amount off per unit (only one threshold). To a large extent a Bill Back deal could be compared to an off invoice deal calculated and charged at a later time. For this reason also, the calculation type is only linear. It often improves the operational margin of the buyer by reducing the cost of the item. It is also called as Retro deal. Types of special bill backs:

- (a) Rebates : A rebate deal (also know as a performance deal) requires the buyer to reach certain goals before the deal kicks in. The supplier tries to improve turnover of its products by giving incentives to the buyer to buy more of its products. The income of these deals does not affect the margin of an item directly and are often setup on a category level.

- (b) Vendor Funded Promotions : During certain periods of the year a supplier or retailer will want to sell more product to reduce inventory or to compete against a competitor. To do this a temporary price reduction could be a good strategy since the consumer will want to buy more products when they are

cheaper. To keep profit margins intact, the retailer will often negotiate a deal with the supplier where the supplier funds part or the entire promotion.

Sometimes an entire category of items will be put on promotion (beer for example) and multiple suppliers will be asked to contribute to the promotion. In such a case it is not required that the item sold is actually supplied by the supplier funding the item.

This also allows for flexibility where multiple suppliers may deliver the same item. Since RMS can not distinguish whose item got sold, both suppliers will contribute based on their contribution percentage.

- (c) Vendor Funded Markdowns : Sometimes goods have depreciated too much or the end of a season is reached so the retailer has trouble turning these units over. Instead of having them all returned to the vendor, the vendor often will support the retailer in discounting the merchandise.

This practice is done for high end fashion items or fast depreciating items like technology based merchandise (computers for example).

The vendor can either support the retailer by contributing a percentage of the retail price discount for each item on hand, or by giving him a fixed amount for all the items he still has in stock.

3.6 Replenishment

Replenishment is the process of automated ordering of items. It monitors retailer's inventory conditions at item/location level. As reorder points are met, RMS replenishment generates orders or recommended order quantities (ROQ) to meet need.

The RMS Replenishment module helps reduce manual processes and increase efficiency by providing:

- Automation of recommended order quantity calculations
- Automation of ordering process from supplier to stores and warehouses
- Automation of transfer process from warehouses to stores

Purchase order generated through replenishment which can be to warehouse, direct to store, or cross-docked to store through a warehouse. Store ROQ is generated through

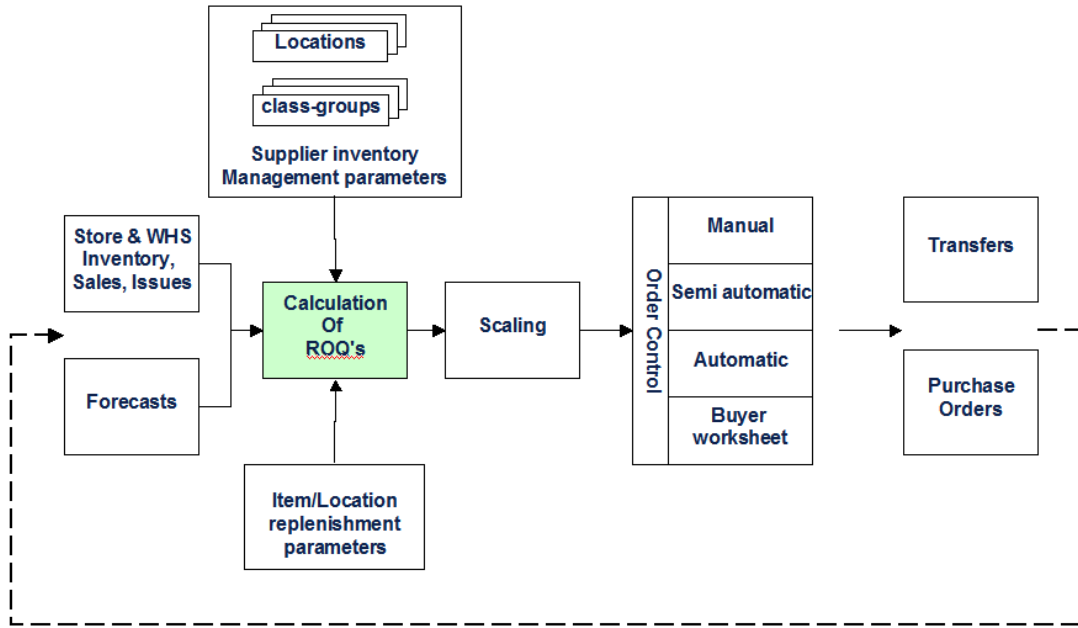


Figure 3.1: Replenishment Summary

replenishment, where stock is obtained by a transfer from the warehouse. Multiple parameters must be set in order for an item to be placed on replenishment. These include:

- Schedule and frequency of replenishment
- Source of merchandise (Supplier or warehouse)
- Replenishment output (Status of replenishment generated orders)
- Algorithm to use when determining need

Only transaction level and simple pack items can be replenished. Replenishment parameters can be set at 3 different levels; supplier, item/location, and item. Some of these parameters can be set up and maintained at higher levels (supplier) and then defaulted to lower levels (item/location, item). Summary of Replenishment process has been shown in Figure 3.1.

3.6.1 Replenishment Methods

There are five total algorithms used to calculate the ROQ. Three of these methods don't use forecasts as part of their algorithm. They are:

- Constant: It is the simplest method. Inventory is ordered when stock falls below a user defined "maximum". Order is required when available inventory is less than

maximum. It does not consider demand trends, future forecasts, or seasonality. It is effective for very slow to slow selling items and items ordered in multiples.

- **Min/Max:** In this method, user defines maximum and minimum stock. Order is triggered when available inventory is less than minimum and fills to the maximum. It does not consider demand trends, future forecasts, or seasonality. It is effective for very slow to slow selling items and items purchased in large multiples
- **Floating Point:** It has same principle as Min/Max method. But in Min/Max method, user defines the maximum stock and in Floating Point method, maximum stock is calculated from sales history(store) and issues history(warehouse). Minimum stock is defined by user (often used if not enough sales history is available to calculate a need). It needs 15 months of sales history for proper need calculations. Calculation is based upon sales rate.

Two of these methods leverage forecasts. They are:

- **Time Supply:** It is the preferred method to prevent stock-outs but not minimize inventory. User specifies min and max days of supply. Actual min and max calculated using forecast data. A time supply horizon is used to calculate the average rate of sales. This can be specified in days:
 - Increase time supply horizon: to smooth out spikes
 - Decrease time supply horizon: where available forecast days is less than max days of supply

Time supply (supplier to warehouse) replenishment calculation methods:

- Time supply replenishment method: bases warehouse ROQ on sum of stores' ROQ it transfers stock to
- Time supply issues replenishment method: bases warehouse ROQ on issues forecast
- **Dynamic:** This is the most complex replenishment method. It minimizes stock on hand, whilst preventing stock outs. It replenishes only quantity required between each replenishment cycle. It considers forecasted demand, review and lead times,

and available inventory in calculation. Dynamic replenishment also considers the following attributes when calculating ROQ:

- Service level: Determines the level of safety stock that is desired.
- Inventory selling days: Desired inventory turn objectives.
- Lost sales factor: Allows any lost sales in the last replenishment period to be factored into the next replenishment period

3.7 Contracts

In Oracle Retail, a contract is a binding agreement with a supplier to purchase a volume of items at a particular cost price over time. It improves supplier relationships by helping suppliers to plan their production in advance. It provides retailers with early margin figures and eliminates risk. The balance of the contract is decreased when an order is written against it, enabling commitment tracking. Stock is sourced from the best contract by replenishment according to predefined logic. Contract Orders are visible from PO and/or Contracts dialog. Item cost is defined on the contract at item/supplier level. Bracket costing, Deals, Scaling and Truck Splitting do not apply on contracts.

3.7.1 Contract Types

RMS has four different types of contracts with varying characteristics and different levels of flexibility.

- Plan / Availability: The plan of manufacturing quantity is decided by ready date. Supplier availability is matched against plan as declared. Item cost is defined. Quantities are specified by availability date at item level. Replenishment orders raised based on need and availability. Retailer will specify the items to be ordered, total quantity of each item, dates on which orders are delivered and negotiated cost of each item. Retailer create orders on the dates specified for available amounts only.
- Plan / No Availability: The plan of manufacturing quantity is decided by delivery due date. Items will be shipped to locations required. Supplier availability is not declared. Item cost is defined. Orders are automatically generated without need taken into account based on contract and system parameters. Manual orders

cannot be raised from Type B contracts. Retailer will specify the items to be ordered, total quantity of each item, dates on which orders are delivered, locations which will receive the items and negotiated cost of each item. Orders are generated automatically based on the plan dates, regardless of whether or not the locations need the shirts.

- **No Plan / No Availability:** There is no plan of manufacturing. No supplier availability declared. Cost is specified at item level. Orders are created manually or through replenishment. Retailer will specify the items to be ordered, negotiated cost of each item and total cost of the item. Retailer creates orders at any point during the contract either by replenishment need or manually.
- **No Plan / Availability:** There is no plan of manufacturing. supplier availability is declared. Cost is specified at item level. Orders are created manually or through replenishment. Retailer will specify the items to be ordered, negotiated cost of each item and total cost of the item. Supplier communicates availability to Retailer as production progresses. Retailer generates orders at any point during the contract only if the items are available.

3.7.2 Ordering using Contracts

During batch replenishment, recommended order quantities are generated and sourced from contracts. For these, RMS will choose best contract (A, C, or D types only) based upon Earliest availability of the item, Lowest Cost of the item, Closed Contracts (A) before Open Contracts (C,D) and C over D. Manual orders can also be raised from a contract. Type A ordering like standard PO dialog and Item List manual ordering for A, C, and/or D type of contract. Item List must have only items that are on the contract. Manually selection is done for contract, locations, and quantities for each item on list.

Chapter 4

OpenScript

4.1 Introduction to OpenScript

OpenScript is a scripting platform developed by Oracle for mainly purpose of creating the automated test scripts that can be used with the OATS. It provides an Eclipse-based environment for generating automated test scripts. The test scripts are built using the Java language. It is easy to use for both developers and non-developers. The OpenScript Interface has been shown in Figure 4.1.

OpenScript has wide range of features that can help users to create automation scripts with ease. Like other testing tools OpenScript also provides proxy settings, correlation libraries, data banking, a standard set of capability such as automatic/manual screen shots and so forth.

4.2 Installation

The following things need to be installed to get automation started. The details of this are explained in the below sections. Oracle Functional Testing's OpenScript scripting platform has the following system requirements:

4.2.1 System Requirements

The following table provides the browser, operating system, and database combination details for which this RMS Automation release has been certified.

- Operating System (32-bit and 64-bit versions): Windows 2003, Windows 7, Windows 8, Windows 2008, Windows 2008 R2, Windows 2012 Server.

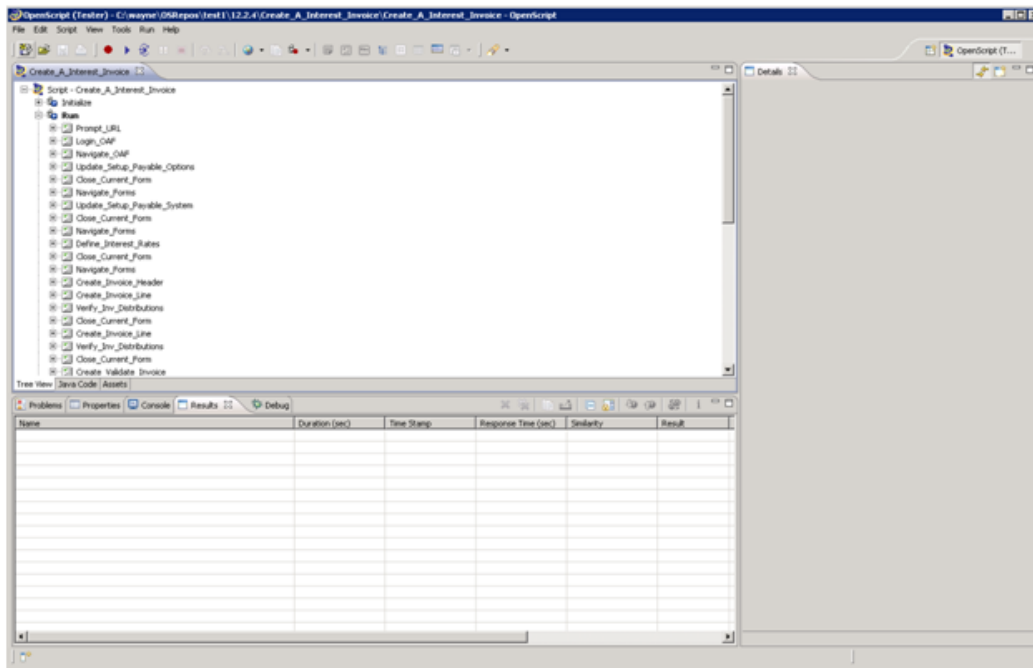


Figure 4.1: The OpenScript Interface

- System: x86, 32-bit or 64-bit processor, 2.6 GHz or faster
- Disk Space: 4 GB minimum
- Browser: Internet Explorer 8.x., 9.x, 10.x, 11.x; Firefox (ESR) 10.x, 17.x, 24.x, 31.x; Chrome 27 to 32 (playback only).
- Oracle Forms: Oracle E-Business Suite Release 12 (Forms 10g) running on Sun JRE and E-Business Suite Release 11i (Forms 6i). Note that E-Business Suite versions running on Jinitiator are not supported with the Oracle EBS/Forms Functional Test module. If your E-Business Suite version runs using Jinitiator, you should use Oracle Application Testing Suite version 12.3.0.x (or previous versions).
- Java Runtime Environment: JRE 1.6 minimum, JRE 1.7, JRE 1.8.
- Java Development Kit: JDK 1.7.0_71.
- Microsoft .NET Framework 3.5 installed prior to installing Oracle Application Testing Suite for Windows 8 and Windows 2012 Server.

4.2.2 Certified System Configuration

The following table shows the software and its requirements.

| System/Software | Requirement |
|------------------------|---|
| Browsers | Internet Explorer 11, Firefox 45.3.0 ESR |
| Java | 1.8.0_101 x86 and x64 |
| OpenScript | OATS 12.5.0.3 Build 1012 |
| Operating System | Windows 7 (64-bit) |
| Database | Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 |
| WLS | 12.2.1 |
| RIB | 16 |
| RMS | 16 |

Table 4.1: Certified System Configuration

4.2.3 Environment Requirement

For an ADF Environment, the following settings need to be configured in the Web.xml file on the ADF Server for OpenScript to work. The server must be redeployed to have the settings applied.

```

<context-param>
  <param-name>oracle.adf.view.rich.automation.ENABLED</param-name>
  <param-value>>true</param-value>
</context-param>
<context-param>
  <param-name>javax.faces.PROJECT\_STAGE</param-name>
  <param-value>Development</param-value>
</context-param>

```

4.2.4 Software and Versions

The below tools against the version if applicable need to be installed for script execution and development.

| Software | Requirement |
|--------------------|---|
| Java JRE | 1.8.0_101 x64 and x86 versions installed only. |
| OpenScript | Version 12.5.0.3 Build 1012 |
| Microsoft Excel | Editing datasheet files. |
| Apache Open Office | Version 4.1.2 |
| JODConverter | Version 2.2.2 |
| Notepad++ | Editing batch and other binary files. |
| WinSCP | SFTP client to upload and download files to remote servers. |
| PuTTY | SSH client to login to remote servers. |
| Java JDK | 1.7.0_71 required only if developing external Java libraries. |
| Eclipse | Developing and modifying the Eclipse project libraries. |
| NetBeans | Developing and modifying the NetBeans project libraries. |

Table 4.2: Software Required for OpenScript

4.2.5 Installing the Accelerator Kit

The Accelerator zip file can be extracted to any location on your local machine. If the machine will be configured to run against multiple users in parallel, it is recommended to extract the zip file to predefined path for each user who will be executing the scripts. This path will be referred to as <SCRIPT_HOME> for future reference in this document. Once the files have been extracted, some configuration will be necessary to setup the scripts for your local machine.

1. Pre-Requisite Library Installation : To execute the Admin API and Induction test scripts, the JODConverter library files need to be downloaded and moved into the appropriate folder. Perform the following steps to set this up:
 - Download the 2.2.2 library zip from <https://sourceforge.net/projects/jodconverter/files/>.
 - Extract the files to a new directory. Navigate to the extracted folder and go to jodconverter-2.2.2/lib.
 - Copy all jar files into <SCRIPT_HOME>/Oracle Retail Automation/OpenScript/OATS_Common_Library/Libraries.
 - This completes the configuration of the JODConverter library.

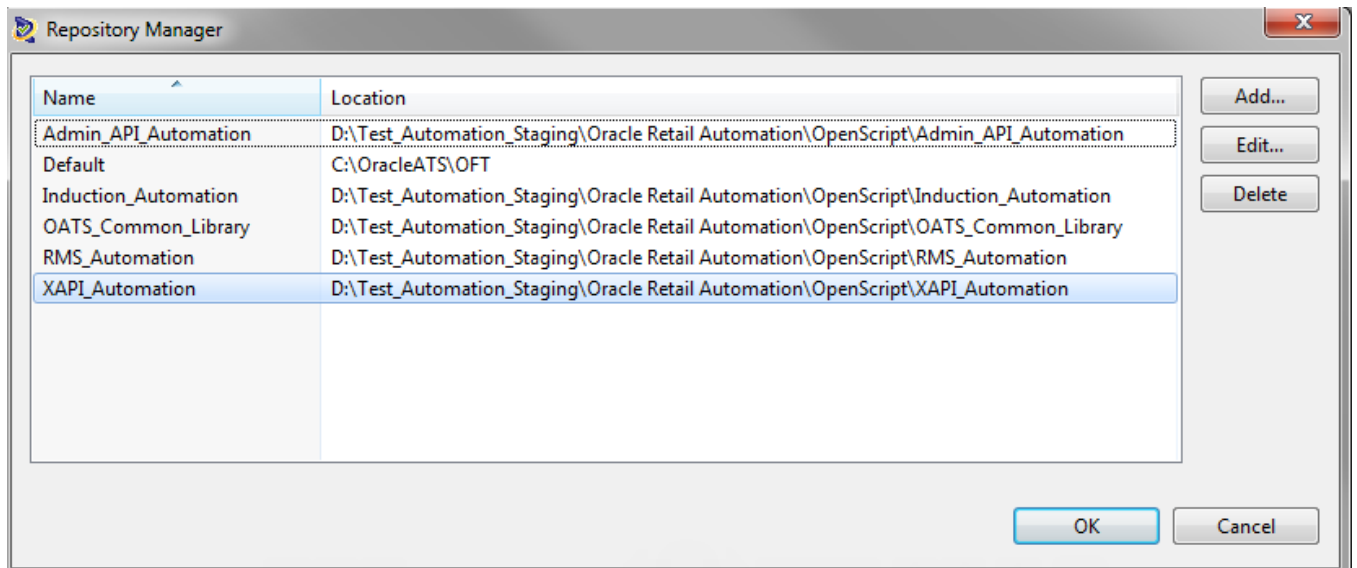


Figure 4.2: Repository Paths

2. Configure Repositories : OpenScript needs a few modifications to locate and run the test scripts. Repositories will need to be added manually for the successful run. These names must be exactly given as:

- Admin_API_Automation
- Induction_Automation
- RMS_Automation
- XAPI_Automation
- OATS_Common_Library

The following steps are to be done to manage the repositories:

- (a) Open OpenScript by navigating to Start Menu > All Programs > Oracle Application Testing Suite > OpenScript. Right click on the icon and select Run as Administrator.
- (b) Click Tools > Manage Repositories. Add and edit the repositories to look like the following below. The locations can be updated to match your local user's configuration. Click OK and close the window.

Repository path has been shown in Figure 4.2.

3. Enable EBS/Forms Automation : With OpenScript still open as Administrator, click Tools > Manage EBS/Forms automation. Select your Java version and click

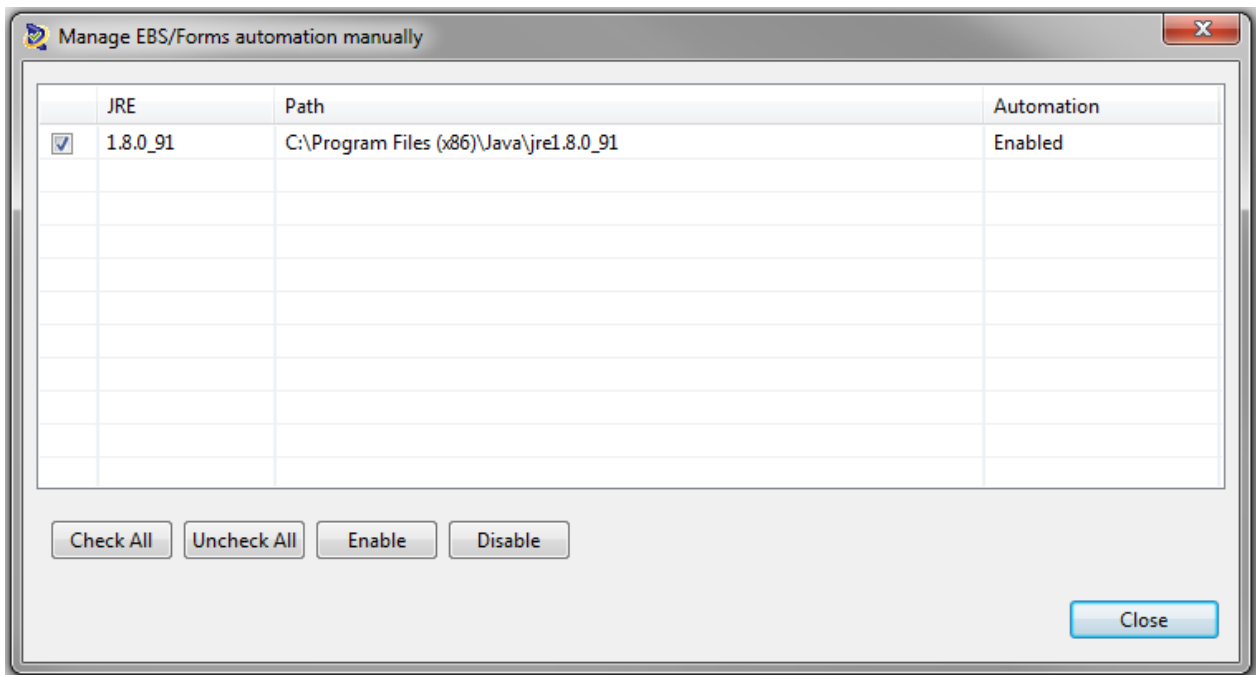


Figure 4.3: EBS/Forms Enablement

Enable. This will allow the Forms scripts to execute properly. Your window should look like the shown in Figure 4.3.

4. Verify Accelerator Kit Installation : To verify OpenScript is configured successfully, click File > Open Script. You should see the 6 repositories that were configured in the Repository Manager. Expand RMS_Automation and click the Test Scripts folder. Open the RMS Automation test script and click on the Java Code tab. If errors are present in the editor, click File > Restart. If errors are still present check the Assets tab and verify all repositories have been configured correctly. This is the main test script that is executed by the RMS_Automation.bat file which can execute test scripts across all projects. You will need to configure the system settings for your local installation and testing environments before running any tests.

4.2.6 System Configuration

The folder <SCRIPT_HOME> -> Oracle Retail Automation -> OpenScript -> RMS_Automation -> Data -> Properties contains the files that hold the system settings and property files used by the automation scripts. These will need to be configured for your installation to run properly.

4.3 Features of OpenScript

Some of the main features of OpenScript are covered below.

1. Using Record and Playback

User can record the browser activities in Firefox or IE. User's actions are recorded and transformed into executable scripts. After recording, user has option to playback. The same thing can be played again with playback button.

2. Integrated tools

Oracle Test Manager (OTM), Oracle Load Test tool (OTL) can be used to manage the test scripts after generation of automation scripts based on script type like load testing, functional testing etc.

3. Flexible Execution

It shows particular execution result to the user. It shows error messages, warning messages, success messages etc.

4. Modular

OpenScript supports Oracle ADF applications, Web, Oracle Forms Applications, Siebel Applications and Web Services. It has support for Java APIs so it can support anything in which Java API is associated.

4.4 Automation using OpenScript

The automation process has become very easy with OpenScript for Oracle EBS and ADF applications. The automation flow chart has been shown in Figure 4.2.

From the Figure 4.2, the automation process is as follows:

A – Picks the Datasheet key prefix to run.

B – Based on the Datasheet key prefix records with status 'Always' record is picked for execution.

C – Based on Action the respective function is called from Test Engine Library.

1 – If excepted or actual Databank from the batch file is set to Forms environment, respective functions are called form Forms library.

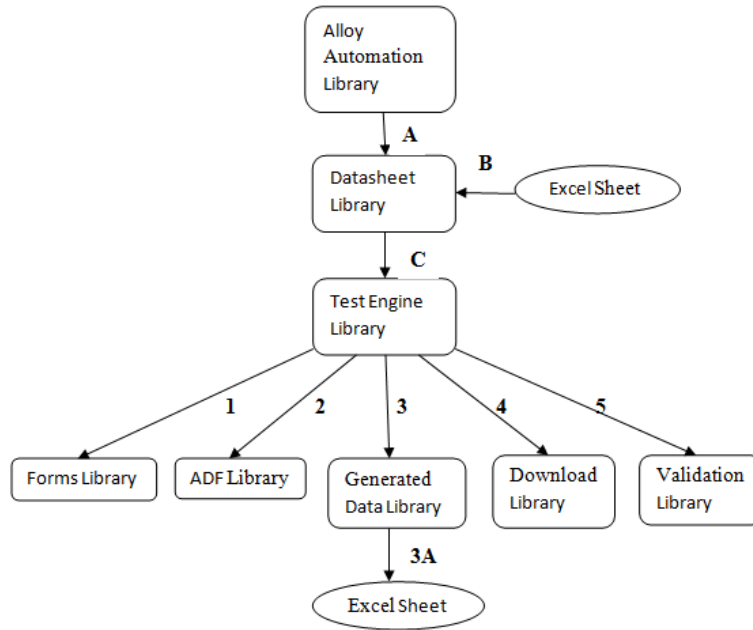


Figure 4.4: Automation Process Flowchart

2 – If expected or actual Databank from the batch file is set to ADF environment, respective functions are called from ADF library.

3 – If the Action is 'CREATE' and scenario is executed successfully in Forms and/or ADF value is stored.

3A – Created value will be updated in Datasheet file, under Generated Data Worksheet.

4 – After successful execution data will be downloaded from Database.

5 – If execution is successful in Forms and ADF then validation is carried on.

4.4.1 Running the tests

To run and test the scripts locally, run the OpenScript -> RMS_Automation -> Batch -> RMS_Automation.bat file as Administrator. The below popup will appear as shown in Figure 4.5.

Automation Project: Different Projects that can be run through automation.

Script Action: Datasheet to run.

Datasheet Folder: Folder from which the data should be fetched.

Dataset ID: Specifies the records that are to be created from the given datasheet.

Expected or Actual Environment Databank: The environment in which the records should be created (value will be the Environment file name that is created in

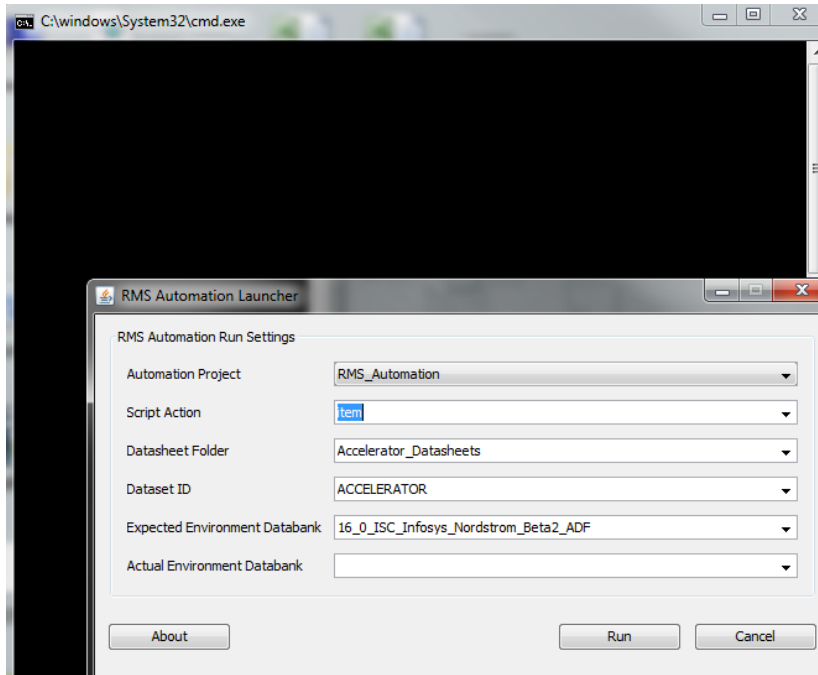


Figure 4.5: RMS Launcher

RMS_Automation/ Data/Environments folder).

After entering all the required data, click “Run” to start the execution. Now the script will fetch the item datasheet from the following path.

RMS_Automation/Data/Datasheets/Accelerator_Datasheets/Foundation Item/RMS Automation Item Datasheet.xls

In the parent sheet all the records with the Dataset as “ACCELERATOR” and Status as “Always” will be created.

4.5 Result Validation and Dashboards

OpenScript gives users the privilege to generate the reports for whatever they are developing or testing. It gives the pass and failure count for the tests with detailed description.

4.5.1 Validating the results

The dashboard utilizes Oracle JET framework. Once the scripts are configured along with the web server, mail and dashboard the execution begins. Results are mailed out to all the people whose ID has been configured. The mail will have the results consolidated by functionality along with the count. Percentages across all buckets and run times are also provided. Result Report is shown in Figure 4.6.

| Area | Description |
|---|--|
| Truncate Results and Check Web Server Hard Disk Usage | Web server use percentage should be configured. The default value is 7. Older results are truncated. |
| Perform Environment Requirement Checks | Major system options that automation scripts have not been run against are listed out and warnings printed. This is done based on a system option compare against the scripts and the environment that the scripts are being run in. |
| Display Datasheet Run Configuration | The input datasheet file and the environment details are listed out. |
| Scenarios | The parent Record ID spliced with the feature Record ID is displayed. This is based off the parent datasheet tab. It represents the test which has been executed along with the result. Passed results and failed results can be drilled down to get the datasheet to view the DB validations and values saved. Failed tests include screen shots which give additional pointers for analysis. |
| Validation Failure | This indicates a failure during validation. Optimal use for this column is when a compare feature is being deployed. Some compare options could be : ADF > ADF with patch Datasheet > ADF (where the datasheet has all the previous data saved and is being used as a reference for the current run) |
| Execution Failure | This represents a failure either with the automation code or Alloy code or environment. |
| Data Failure | This indicated a data issue in the input datasheet and involves the values in the datasheet to be sorted. |

Table 4.3: Validation Area and Description

Foundation Hierarchy

| Datasheet | Passes | Validation Failures | Execution Failures | Data Failures | Total | Run Time | Results Link |
|------------------------|--------|---------------------|--------------------|---------------|-------|----------|---|
| Organization Hierarchy | 7 | 0 | 0 | 0 | 7 | 00:15:17 | Organization Hierarchy Test Results |
| Store | 3 | 0 | 0 | 0 | 3 | 00:18:16 | Store Test Results |
| Warehouse | 2 | 0 | 0 | 0 | 2 | 00:24:20 | Warehouse Test Results |

Figure 4.6: Result Report

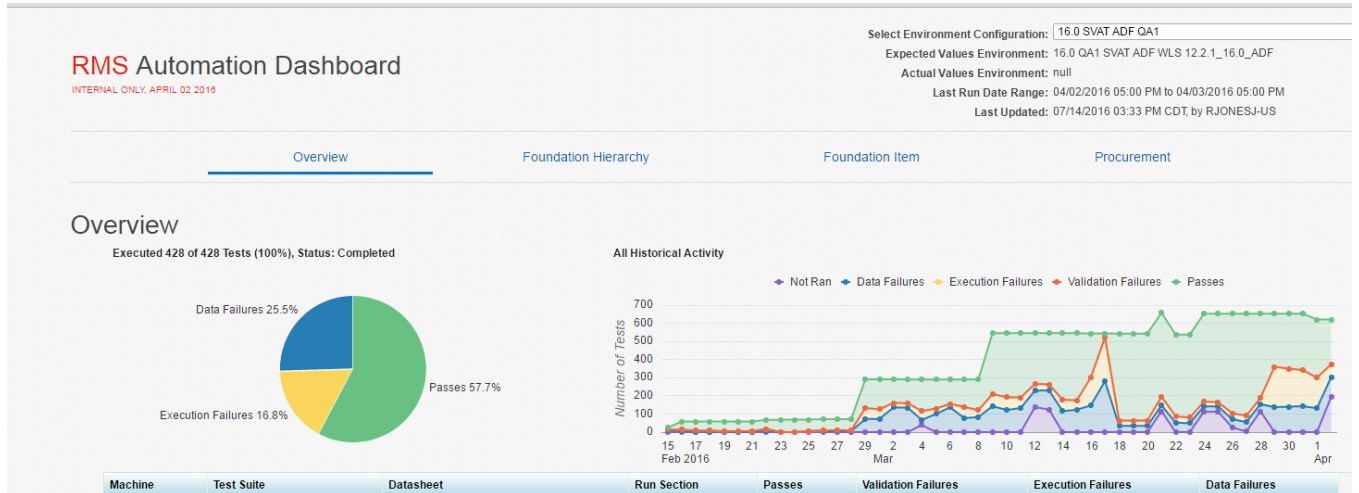


Figure 4.7: Result Dashboard

4.5.2 Dashboard

The dashboard visually represents the automation execution status with various actionable items. Dashboard is shown in Figure 4.7.

| Dashboard Block | Description |
|-----------------|---|
| Environment | The block displayed the environment configurations against which the dashboard is being displayed. |
| Overview | This represents the overall status along with the various percentages. These include Passes, Validation failures, Execution Failures and Data Failures. |
| Functionality | Feature functionality covered by automation scripts are listed here. These are Admin API, Foundation Hierarchy, Foundation Item, Procurement, Inventory, End to End, XAPI and Induction. The dashboard user can get the detailed status by functionality for additional analysis. |
| Run Summary | This detail the run summary against the different dataset ID's along with the totals. The dataset ID's are defined in the datasheets. |

Table 4.4: Description of Dashboard

4.5.3 End Point and Purge

DB tables validation termination points

The scripts have trace enabled hence all the tables irrespective of data being present or not will be validated against if there are triggers on it driven by the code. Validation ends in the RIB tables of RMS. If the RIB is turned on and automation is running then the scripts will validate all the RMS RIB tables to ensure there are no errors and the impacted MFQUEUE tables are truncated. The assumption here is if there are no errors on the RMS RIB tables the messages should flow to the integrated applications if the environment supports it. If the environment is standalone then the validations will still include the MFQUEUE tables and the RMS RIB tables. There will be no errors here as these tables are impacted only in an integrated environment.

Recommended purge strategy

Automation generates data based off the number of records present in the datasheet. There is no cleanup of automation data after the scripts run. All data remains in the environment.

Some suggestions for the purge strategy include:

1. Database Image : Before the automation scripts kick off a database backup can be done. Once the scripts execute and the results are satisfactory, the image can be restored to the saved backup.
2. User ID Based Automation : A predefined user ID can be utilized for the automation scripts. This can be used as the basis for data identification for a purge strategy where all the automation data is consolidated and the purge batches are run sequentially. Care should be taken to ensure that the data passes the history/days the system options maintain for the feature so that the purge is successful.

4.6 Analysis

OpenScript is very useful platform for testing the applications. The user will have to run the batch file to trigger the automation. Input data can be inserted through the excel sheets.

4.6.1 Advantages

1. Robust architecture.
2. Supports simultaneous execution.
3. Supports all browsers.
4. Better Security.
5. Detailed test execution report. Shows particular failure point so resolving the error becomes easy.

4.6.2 Disadvantages

1. Slower with ADF application.

4.6.3 Conclusion

OpenScript is a tremendous tool for automation testing and it will work better with on prem solution because of its security, robustness and failure recovery mechanism.

Chapter 5

Oracle Flow Builder

5.1 Introduction to OFB

OFB is a keyword driven testing application used to test the softwares and build the test automation flow. These test flows can be translated into the OpenScript scripts and after that it can be executed. It is easy to use and doesn't need any trained software testers to use the application. There are inbuilt keywords given in the OFB to create component. Components can then be connected together to build the component sets and flow and flow can be converted into executable scripts.

OFB is mainly designed for testing Oracle E-Business Suite applications. It has starter kit with 2100+ components 200 flows for testing Oracle EBS. It has following features :

- Home: Shows Access Righths, new Notifications, newly created Components. Access Rights can be requested and changed through Home screen.
- Components: Shows available components from component tree. Add and update components options are available on right click of component.
- Component Sets: The components which are used together frequently can be linked together to generate component set. Directly component set can be used to created flows.
- Flows: Flows can be created by components or component sets. Flows then can be translated into executable OpenScript scripts.
- Notifications: Shows the informational messages generated in the application.

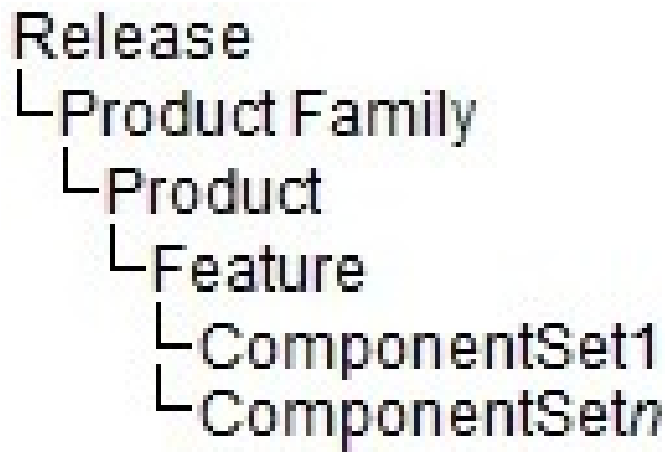


Figure 5.1: OFB Overview

- History: Shows history of components, component sets and flows.
- Settings: Used for basic settings of the application like email notification references.
- Reports: Reports can be generated for flows, components and component sets.
- Administration: Provides options to create release, product family, feature. Also library can be added and email server can be set up.

5.2 Generating Executable Scripts through OFB

The basic procedure of creating executable script in OFB is as follow :

1. In administrator, we can define release, product family, product and feature. The hierarchy is shown in Figure 5.1.
2. Component can be added through Component screen. Developers can define component code that is inbuilt keywords, objects and parameters. Component Code has been shown in Figure 5.2.
3. Flows can be created from Flow screen. Components are added in sequence to create flow. Flow Tree has been shown in Figure 5.3.
4. After creating the flow, test script can be downloaded as shown in Figure 5.4. The script then can be executed via the OpenScript.

| S.No | Keyword | Object | Display Name | Attribute Values | Output Parameter | Function Name | Mandatory | Rerunnable | Tooltip | Default Data |
|------|---------------|---------|-------------------|------------------|------------------|---------------|-----------|------------|---------|--------------|
| 1 | SETAPPTYPE | WEB | | | | | No | No | | |
| 2 | SETWINDOW | | *Add Attachment | *Add Attachment | | | No | No | | |
| 3 | SELECT | LISTBOX | Add Attachment Ty | @name='AddAttad | | | No | No | | |
| 4 | WAIT | WINDOW | *Add Attachment | *Add Attachment | | | No | No | | |
| 5 | STARTOPTIONAL | | | | | | No | No | | |
| 6 | STARTKEY | | Show More Options | | | | No | No | | |
| 7 | CLICK | LINK | Show More Options | @text='Show More | | | No | No | | |
| 8 | ENDKEY | | | | | | No | No | | |
| 9 | ENDOPTIONAL | | | | | | No | No | | |
| 10 | WAIT | WINDOW | *Add Attachment | *Add Attachment | | | No | No | | |
| 11 | SETTEXT | EDIT | Description | @name='Attach_0. | | | No | No | | |
| 12 | STARTOPTIONAL | | | | | | No | No | | |
| 13 | STARTKEY | | Go | | | | No | No | | |
| 14 | CLICK | BUTTON | Go | @value='Go' | | | No | No | | |
| 15 | ENDKEY | | | | | | No | No | | |

Figure 5.2: OFB Component Code

Flow Creation

Unlock
Add Attachments
Assemble
Make Component Set

sample_purchasing_flow

- sample_purchasing_flow
- sample_purchasing_scrn

Figure 5.3: OFB Flow Tree

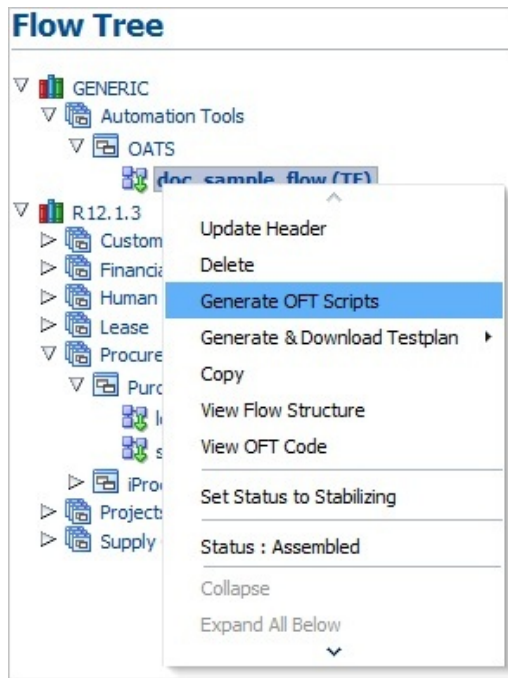


Figure 5.4: Generate OAT Script in OFB

5. The result file of the testing will be generated in OATS folder. That report will have detailed execution steps. We can know success or failure of the test from the report. The test plan is also automatically generated. We can download test plan from Flow.

5.3 Analysis

OFB is a very good application for testing. But with advantages it also has some disadvantages.

5.3.1 Advantages

1. It is a keyword-driven testing application so we can use keywords and generate the code.
2. We can use OATS generated report to check whether entity has been created or not. No need to query database.
3. Test plan is automatically generated through OFB.
4. The input data can be inserted through excel sheets or via OFB application.

5.3.2 Disadvantages

1. Once the function library is added, we cannot add/modify functions to the function library. Also we cannot delete the function library from OFB once it is added.
2. ADF objects are not working with OFB.
3. We have many less options for ADF in component code.

5.3.3 Conclusion

OFB provides solution to the cloud problem. We can easily detect entity creation without querying database through OATS report. OATS gives very detailed execution report so that if there is any failure in the process, we can easily track that. But its major drawback is that it doesn't work well with RMS ADF application. So for Oracle EBS application this is very good but for RMS ADF application this cannot be used.

Chapter 6

Selenium

6.1 Introduction to Selenium

Selenium is an open source solution for testing the web applications. It is widely adopted and supported by a large community. It is highly flexible and reliable. We can add functionality to both test scripts and framework for customization of test scripts. This is Selenium's biggest advantage over other open source automation tool. Source code can be downloaded and updated as per requirement of the project. It has many different options available for supporting different different kind of test automation problems. It can also support executing tests on multiple browsers. It can support multiple programming languages like java, csharp, php, ruby, python, javascript and perl. Selenium test scripts will not work without Javascript. So Javascript should be enabled in browser to run the test scripts. It supports all the Operating Systems.

6.2 Test Automation With Selenium

Selenium is not a single tool but it is made of different different softwares. Selenium Suite Overview is shown in Figure 6.1.

1. Selenium 2 Selenium RC and WebDriver are merged into one and it is now called as Selenium 2. It has object oriented APIs for better support of test automation. It supports WevDriver APIs and Selenium 1 technology. It provides maximum flexibility to port test automation.
2. Selenium IDE Selenium IDE is kind of prototyping tool for generating the test scripts. It also uses record and playback method for building test scripts. It is

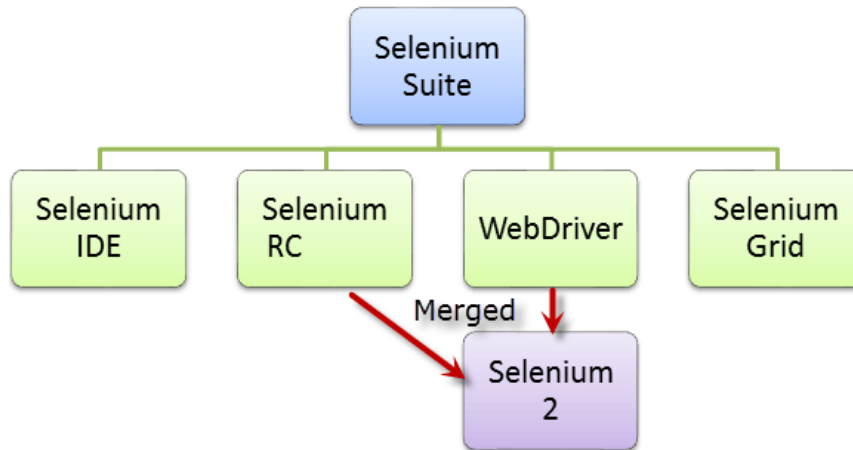


Figure 6.1: Selenium Suite Overview

available only in Mozilla Firefox as add-on. It records the user action performed on browser and transform them into reusable scripts.

3. Selenium-Grid Selenium-Grid scales Selenium RC solution for large test suites that can be run in multiple environments. It allows parallel execution of the test scripts on different remote machines at the same time. It is comparatively faster and boost the performance by running tests parallel.

6.3 Analysis

6.3.1 Advantages

The open source tool Selenium has some specific advantages and also it is not without its drawbacks.

1. Very easy to install and use.
2. It is reliable and flexible.
3. Supports Google Chrome, Firefox, Internet Explorer 7, 8, 9, 10 and 11, Safari, Opera, HtmlUnit, Android and iOS.
4. Test report can be generated through JUnitReport so no need to query database.
5. Selenium can be used for UI, acceptance, regression and unit testing.
6. Faster with ADF application.

6.3.2 Disadvantages

1. Selenium is open source so we can have security related issues. Security will be more in OpenScript.
2. No inbuilt reporting capability so we need plugins like JUnit for test reports.
3. It has lot of challenges with IE browser.
4. Content Management like document upload/download is not supported by Selenium.
5. Not an Oracle ingrown tool

6.3.3 Conclusion

So Selenium provides the solution to our cloud problem. It has some disadvantages but those disadvantages can be overcome and with help of some extra plug-ins and libraries we can resolve the issues. So Selenium is the better solution for automation of RMS on cloud.

Chapter 7

Customized Datasheet Creation

The framework which is currently being used by Oracle for automation testing requires lots of manual efforts for creation of datasheets. The purpose of this task is to reduce the manual effort and make the automation testing completely automated. The datasheets creation will be done with single click and thus eliminating the human resources to create the datasheets.

7.1 Introduction

Software testing is the crucial part of Software Development Life Cycle. Different kind of approaches have been introduced for testing the software in a way that all the requirements are being covered. Manual testing is necessary part of Software Development Life Cycle, which requires lots of manpower and also it is time consuming. After that, automation testing was introduced. The purpose of automation testing is to test the software automatically to check whether all the requirements for the system are being met. The obvious advantages of the automation testing is that it is faster than manual testing, it requires less human resources and it is less prone to the human errors. The current framework which is being used by Oracle for the automation testing is taking input from excel sheets and test the system with that data. These input spreadsheets are the test cases defined for the software. These test cases are written in a particular format in which the scripts written for the automation will take them as an input data to the software and fill them in the system. After completion of testing, the automation report will be generated with number of passed and failed test cases. Based upon the generated report, one can figure out the bugs in the system which are the cause of failures. In

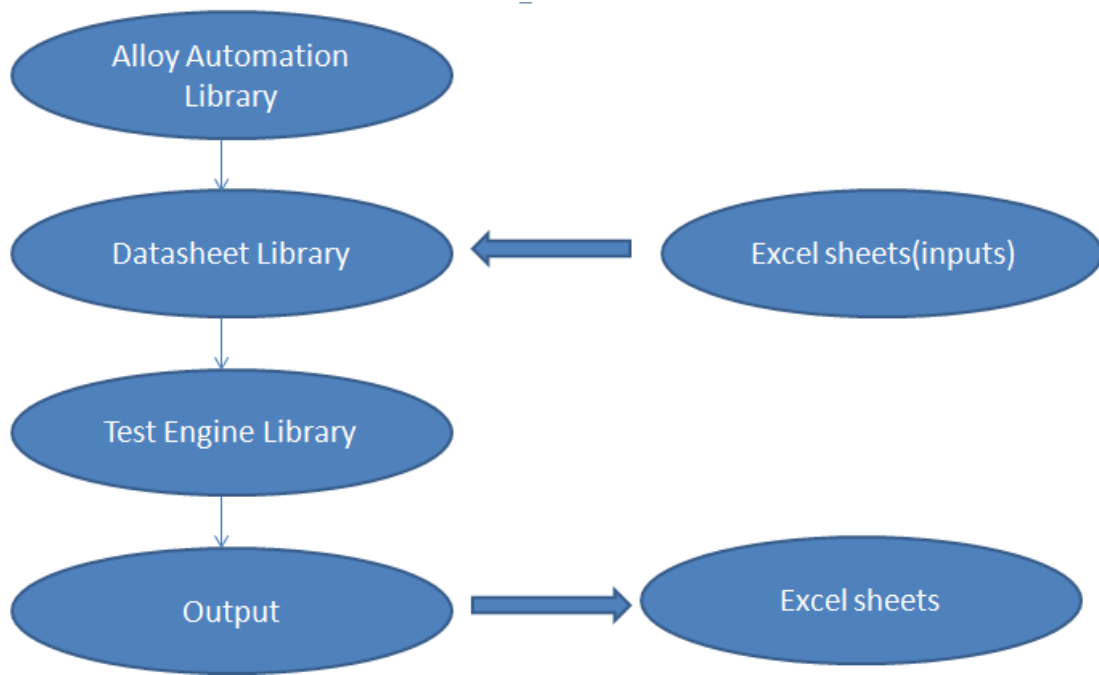


Figure 7.1: Datasheet Process Flow

Figure 7.1 the datasheets process flow has been shown.

The current framework is doing extremely fine with taking input data from excel datasheets, filling them in the system and testing the system. But here the major part is creating the datasheets with all the required test cases. Now the testers are generating the test cases and writing them into the datasheets. But as we know, the humans are more prone to the error and also there are chances of mistakes while creating datasheets which cause problem in the testing going forward and will lead to the non-existing bugs. Also, the customers are not aware of how to fill the datasheets. And filling large amount of data is tricky part. Therefore, to avoid these problems, “Customized Datasheet Creation” was introduced. As per this project, the datasheets will be created automatically without any human intervention. The idea is to fetch data from database which already exists and convert them into the existing required datasheet format. Thus, the datasheets will be automatically created and can be uploaded directly to the system. This saves time and also it is very user friendly.

7.2 Current vs. New Automation Framework

The current format for the automation testing is the customer has to create the datasheets covering all the scenarios. There is one batch file which is used to trigger the run. It will call all the scripts which are used to automate the system testing. The automation scripts

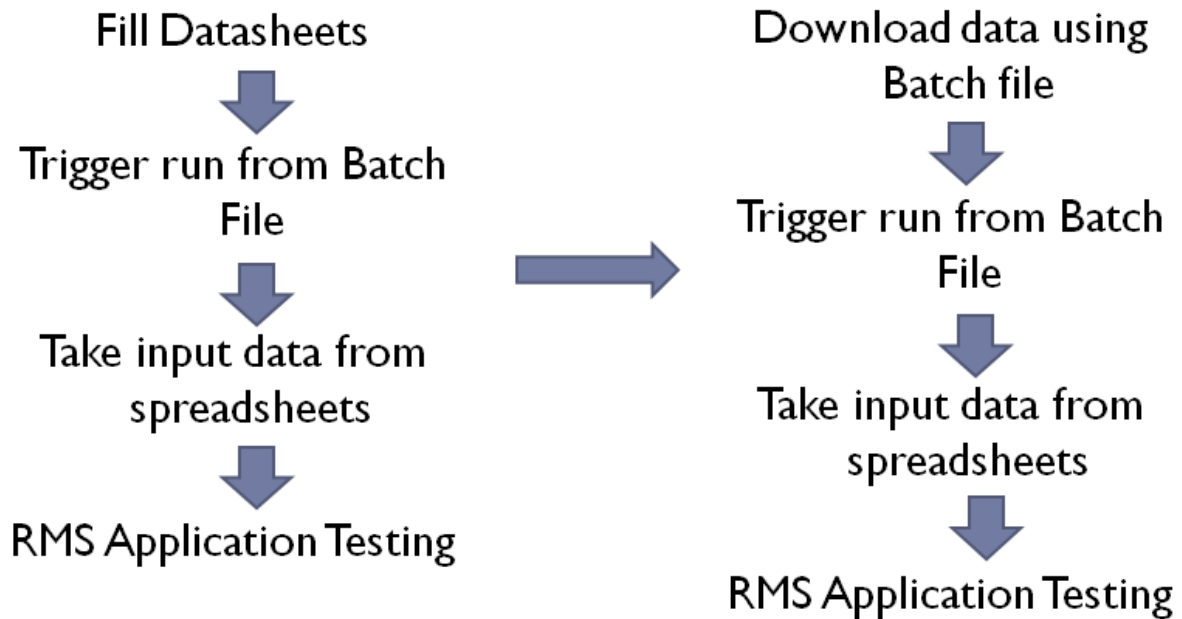


Figure 7.2: Current vs. New Automation Framework

will take the excel sheets as an input and test the system with that data. In the new framework, everything will be same except the first part. In the first part, for creating the datasheets, the customer or tester has to run the one more batch file which will trigger the download action. The data will be downloaded in the predefined datasheet format. The comparison is shown in the Figure 7.2.

7.3 Datasheet Format

The format in which we are inserting data into RMS through automation script is predefined. For every module, there can be one or more datasheets. For example, RMS consists of modules like Organizational Hierarchy, Suppliers, Stores, Warehouse, Partners, Items, Purchase Orders, Transfers etc. For each of these modules there exist datasheets. Each datasheet will have parent worksheet and other worksheets. Other worksheets refer to the screen in RMS or some functionality. Sample datasheet format has been shown in Figure 7.3.

Each parent sheet is having fixed format. It includes following fields :

- Record ID - It should be unique and it also map to the test scenario.
- Dataset ID - Based on the input given in the batch file. particular set for execution.
Example : ALLOY_SMOKE.

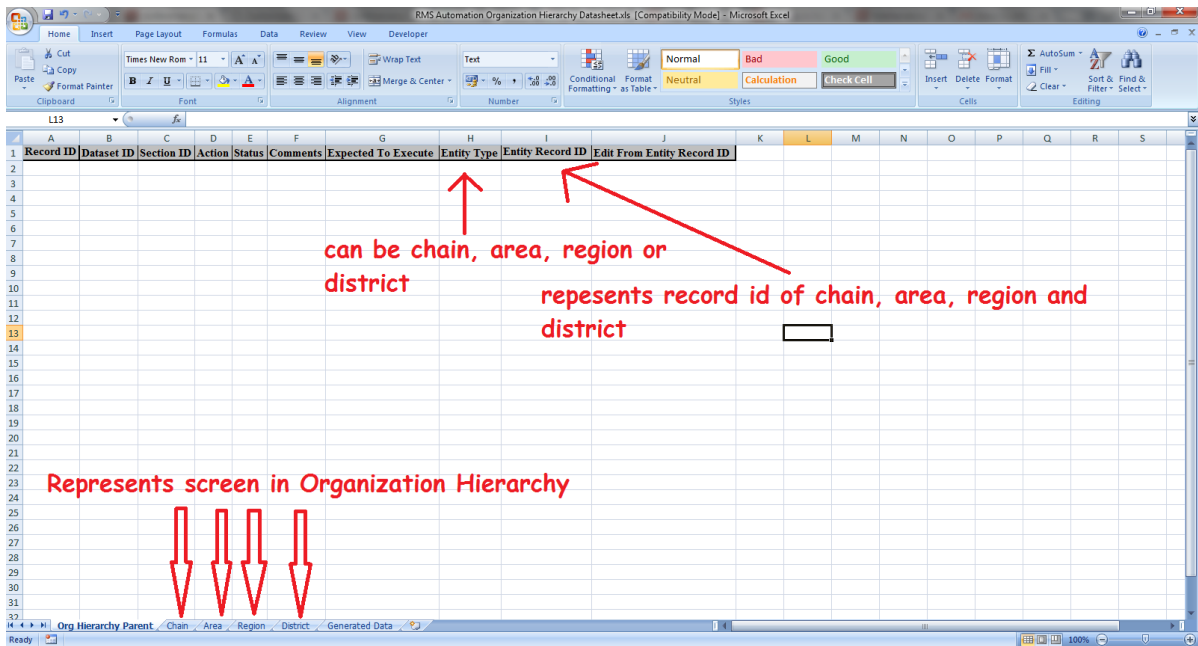


Figure 7.3: Sample Datasheet Format

- Action - Specify action like create, edit, ship, receive etc.
- Status - It indicates the run status of the record. Example : Always, Not Ready.
- Comments - Comment about test case.
- Entity Type - Specify entity of module.

Record IDs are unique identification of the test case across all the modules. When the test case will pass successfully then in its datasheet it will make an entry with Record ID in the “Generated Data” worksheet. Now if in some other module that entity is required then customer can directly refer to that Record ID and give that as an input and the scripts will automatically fetch its ID from database while entering into the RMS. This is how it is used to uniquely identify each record generated in the system through the automation. It maps Parent sheet with 1:1 relation. There is one more concept of Grouping Name. Grouping Name is used to enter multiple data into some module. It is grouping of data which we want to enter in some specific module. For example, items are delivered from multiple locations, so in the item locations screens, we can add multiple locations like stores and warehouses. Here we can use Grouping Name. All the stores and warehouses will be defined under one Grouping Name and that name will be given in the Item Location Grouping Name. This saves repeating of the test case for

multiple input combinations. It maps Parent sheet with M:1 relation. Grouping Names are reusable. Whenever test case demands the same grouping of data we can use the priory used Grouping Name.

The datasheets can be customized to include scenarios applicable to the retailer. The key columns are explained below. The user has to take care of the constraints while customizing the sheets.

1. Dataset ID : This field is used to group a set of scenarios to be executed together in a datasheet. These Ids can be customized as per user convenience. E.g.: All the smoke scenarios can have a single dataset id as “Smoke” and these records can be executed for Smoke testing.
2. Grouping names : This indicates a set of values grouped under a common name. E.g.: List of all the address required to create a supplier is grouped under a common name “Fashion Retailer Address”.
3. Generated data : This tab gives the consolidated view of the data combinations created. The data creation is based off the scenario combinations and the parent sheet. This maps the Record ID field and the value in the application. The user can then choose to reference the value as either the Record ID field or hard coded values based on the entity value as given in the generated data sheet.
4. Section ID : The section ID column can be utilized to run a specific set in the datasheet. It needs to be passed which running the scripts separated by “:”. With this only data matching the section ID will be run.
5. Batches : Batches with the pre, post conditions along with the correct sequence must be built into the code. The areas which have batches have the Run batch equals to Y in the datasheets. This in turn calls the code which the batch is executed and results validated.

Figure 7.4 shows the actual datasheet for items.

7.4 Download Data

For generating the datasheets, we need data. For RMS, we already have data in the database for its older versions. For new release of RMS, we can directly use data. So

| Record ID | Dataset ID | Section ID | Action | Status | Expected To Execute |
|-----------|-------------|------------|--------|--------|---------------------|
| 2 | ACCELERATOR | | Create | Always | Yes |
| 3 | ACCELERATOR | | Create | Always | Yes |
| 4 | ACCELERATOR | | Create | Always | Yes |
| 5 | ACCELERATOR | | Create | Always | Yes |
| 6 | ACCELERATOR | | Create | Always | Yes |
| 7 | SYSTEM | | Create | Always | Yes |
| 8 | SYSTEM | | Create | Always | Yes |
| 9 | SYSTEM | | Create | Always | Yes |
| 10 | SYSTEM | | Create | Always | Yes |
| 11 | SYSTEM | | Create | Always | Yes |
| 12 | SYSTEM | | Create | Always | Yes |
| 13 | REGRESSION | | Create | Always | Yes |
| 14 | REGRESSION | | Create | Always | Yes |

Figure 7.4: Datasheet For Items

I came up with the idea of reusing that data only. The data will be downloaded from database in the predefined format of our datasheets. For that, I have made batch file which will trigger download function and will call the other automation scripts which have functions for downloads. This is how the data will be downloaded and almost all the test cases will be covered and the system will be able to be tested covering all the scenarios.

7.4.1 Batch File Creation

The automation scripts are written in the OpenScript. OpenScript is a tool designed by Oracle to write the scripts. To call the scripts written in the OpenScript from the download batch file, we have to call OpenScript batch file. It will initialize the instances of the OpenScript. The batch file will take one text file as an input. In that text file following fields are specified :

- Machine Name: In which machine user wants to download data
- Entity: The entity for which he wants to download data i.e. Items, Purchase Orders, Transfers etc.
- Environment: There are many environments available. User here can specify the environment from which he wants to download data.

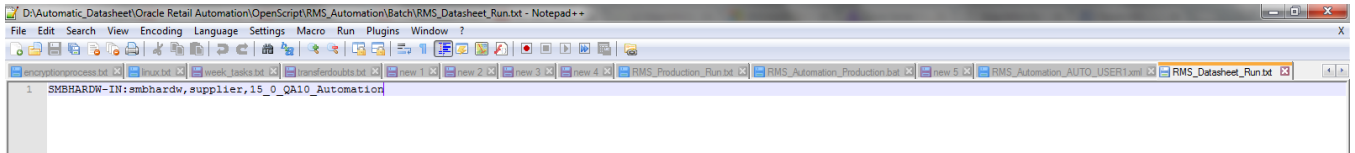


Figure 7.5: Batch File

The host machine is the machine name of the computer from which user is going to trigger the download action. RMS is a big application comprised of many modules like Stores, Warehouses, Inventory, Transfer, Purchase Orders, Items, Deals, Contracts etc. So in this text file, user can define the module for which he wants to create the datasheets. There are many environment available for RMS. Each environment is integrated with database. For each release there is at least one environment available. Till now, RMS had many versions so there are many environments available with different data. So this field is for specifying the environment from which user wants to download the data. In the Figure 3.1 batch file has been shown.

7.4.2 Reverse Engineering Approach

In RMS, everything is connected. The chain starts from Organizational Hierarchy and will end on either items, PO or transfers. There are three clusters defined in RMS.

1. Foundation Data
2. Foundation Items
3. Procurement

Foundation Data is the base cluster. Other two clusters are totally dependent on this cluster. Foundation Data includes entities like supplier, store, warehouse, area, region, partners. This all information is mandatory while creating the items. Items will require the supplier information from which the items will be delivered, also location information from where the items will be delivered. So to create the items, foundation data is required. The purchase orders will be generated for the items. So to create the PO, items are necessary. Transfers also require items. So this is how all the clusters are dependent on each other.

So when the user downloads the item, item datasheet will be generated. But while uploading it in the environment if the foundation datasheets are not present, it will

not create the item and will throw the missing data error. So I had to use reverse engineering approach for this. That means, if user is downloading PO, respective items and foundation data will be also downloaded. When user is downloading items, respective stores, warehouse, suppliers will also be downloaded. So when user will try to upload the same in the environment, it will be created successfully without any issues.

7.5 Outcome

The download batch file is creating the datasheets in seconds. All the datasheets will be generated and ready to upload in a very short period of time. This is really advantageous in terms of saving the time. When these datasheets are created manually it takes days and sometimes months to get completely created. It has been observed that the total time taken for testing the application is equally divided between the time taken to create the datasheets and time taken to test the application with that data.

Total Testing Time = Time Taken to create datasheets + Time Taken to test the application

The time taken to create the datasheets is 50% and time taken to test the application is 50%. But with automation of creating the datasheets the time to create the datasheets has been drastically reduced. It has been observed that the datasheets are getting created in seconds which leads to time reduction in almost 48%. So now,

Total Testing Time = Time Taken to create datasheets(2%) + Time Taken to test the application(50%)

Therefore,

$$TotalTestingTime = 52\% \tag{7.1}$$

So from this we can say,

$$TimeSaved = 100\% - 52\% = 48\% \tag{7.2}$$

Thus, it is major advantage in time reduction. Testing can be faster and easier with

this approach. Also it eliminates the human error possibilities, so failure because of missing data or improper data is rare to appear. Thus it is increasing the efficiency of testing.

7.6 Advantages

There are many advantages of automating the datasheets. It is not only useful to the testers but also to the customers who are not aware with creating the datasheets. There are also chances that testers may forget to include some test scenario or do some mistake while creating the test scenario but with this automation of datasheets, all this possibilities are eliminated. Following are the advantages of the automation of datasheets.

- Customer can create datasheets without having any prior knowledge about how to create datasheets.
- User friendly. Just have to run file to create datasheets.
- Faster. Create datasheets in seconds.
- Also can be used for regression testing of the system.
- By this way user can have backup of his important data.

Chapter 8

Future Work

The automation framework is doing great. It is useful in many ways. Oracle is using one more software that is Oracle Test Manager(OTM). In OTM, all the test cases have been reported. Those test cases will be executed and their status “pass” and “fail” will be remarked. The next step in automation is to pull all the test cases from our automation datasheets to OTM and after execution of the test cases their status should be automatically updated to the datasheets. This will increase the coverage of testing.

Chapter 9

Conclusion

OpenScript, OFB and Selenium all are great tools for testing the web application. They have support for all the browsers, operating systems and many programming languages. OpenScript and OFB are Oracle developed tools and Selenium is an open source tool. OFB works well with EBS/Forms. But OFB doesn't have support for ADF. Selenium has support for ADF elements. Security will be more in OpenScript and OFB than in Selenium. So for on prem solution we can use OpenScript and for Cloud solution we can use Selenium. The problem of not to query database in the Cloud will be resolved because we can use JUnit Report for tracking the status of test script. JUnit will give detail report of execution so that we will be able to know that our test script has passed successfully or it has failed. Also it shows particular failure point of the test so solving errors will be easy. And with "Customized Datasheet Creation", the datasheets will be created automatically without any human intervention which is great and also it increases the testing efficiency.

Bibliography

- [1] Functional Testing Flow Builder User's Guide, Available online at http://docs.oracle.com/cd/E59555_01/OFBUG/toc.htm
- [2] Fei. Wang, Wencai. Du, *A Test Automation Framework Based on WEB*, 2012 IEEE/ACIS 11th International Conference on Computer and Information Science
- [3] Oracle Application Testing Suite Documentation, Available online at https://docs.oracle.com/cd/E25294_01/index.htm
- [4] Cognizant, *Taking testing to the cloud*, in: Cognizant Whitepaper, September 2011, Available online at <http://www.cognizant.com/Taking-Testing-to-the-Cloud.pdf>, Last Accessed: September 2014.
- [5] YASH Technologies, *Oracle Forms to Oracle ADF, A migration tool*, White paper, 2014
- [6] Pratima Bhat, Bushra Alam, Dr Shankaraiah, Karamjit Kaur *Spreadsheet Editor - A Migration Solution from Oracle Forms to ADF*, International Journal of Computer Trends and Technology (IJCTT)
- [7] Ravi Prakash Verma, Bal Gopal, Md Rizwan Beg *Algorithm for Generating Test Case for Prerequisite of Software Requirement*, International Journal of Computer Applications (0975 - 8887)
- [8] Abhijeet Sartape, Prof. Vasgi B., *Data-Base Security Using Different Techniques: A Survey*, International Journal of Computer Trends and Technology (IJCTT)