

# Analyzing and Automating Manual Activities within Oracle Retail Environment Security

Submitted By

**Akshar Sondagar**

**17MCEI15**



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
INSTITUTE OF TECHNOLOGY  
NIRMA UNIVERSITY

AHMEDABAD-382481

May 2019

---

# Analyzing and Automating Manual Activities within Oracle Retail Environment Security

---

## Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering  
(Information & Network Security)

Submitted By

**Akshar Sondagar**

(17MCEI15)

Guided By

**Asst. Prof. Rupal Kapdi**



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2019

# Certificate

This is to certify that the major project entitled ”**Analyzing and Automating Manual Activities within Oracle Retail Environment Security**” submitted by **Akshar Sondagar (17MCEI15)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering (Information And Network Security) 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 major 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. Rupal Kapdi  
Guide & Assistant Professor,  
CSE Department,  
Institute of Technology,  
Nirma University, Ahmedabad.

Dr. Sharada Valiveti  
Associate Professor,  
Coordinator M.Tech - CSE (INS)  
Institute of Technology,  
Nirma University, Ahmedabad

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

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

## Statement of Originality

---

I, **Akshar Sondagar, 17MCEI15**, give undertaking that the Major Project entitled **”Analyzing and Automating Manual Activities within Oracle Retail Environment Security”** submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering (Information And Network Security)** 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: Ahmedabad

Endorsed by  
Asst. Prof. Rupal Kapdi  
(Signature of Guide)

## Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Prof. Rupal Kapdi**, 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 she has imparted has been a great motivation to me in reaching a higher goal. Her 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 D Bhavsar**, Hon'ble Head of Computer Engineering 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 Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- **Akshar Sondagar**  
**17MCEI15**

# Abstract

The project titled Analyzing and Automating Manual Activities within Oracle Retail Environment Security is associated with the Oracle Retail products applications and the server technologies used by the Development Support Services team of the industry. The aim of this project was to reduce the tremendous amount of workload of the DSS, by contributing to some aspects of the responsibilities of the team. Consequently, one part of this project aims to conduct product deployments and implementations on specific environment and architectural platforms in a phased manner, covering each possible detail at every step, while strictly following the appropriate installation guide and provide feedback to make the entire process simple and hassle-free, for customers and clients who are not very familiar with database and server technologies. So company requires to deploy all the products over all the servers. All the products uses various components like database, dashboards, etc. Because of these reasons it is very difficult for a human to sit and install each product on each machine as per the customer requirements. Then Automation comes into picture. Automation process helps in all these process to successfully deploy the product and run it successfully. It saves a lot of time, money and efforts of human. The second part of the project involves the development of automation tools for the DSS team, for various operations. The increasing complexity and time consuming process of software development phases demands the use of automated software tools.

# Abbreviations

<b>SVN</b>	Subversion
<b>JSch</b>	Java Secure channel
<b>SSH</b>	Secure Shell
<b>SSE</b>	Server Sent Events
<b>RSB</b>	Retail Service Backbone
<b>RIB</b>	Retail Integration Bus
<b>OID</b>	Oracle Internet Directory
<b>RCU</b>	Repository creation utility
<b>LDAP</b>	Lightweight Directory Access Protocol.
<b>TR</b>	Technical Request.
<b>SR</b>	Service Request.
<b>FMW</b>	Fusion Middleware.
<b>RMS</b>	Retail Merchandising System.
<b>AIP</b>	Advanced Inventory Planning.
<b>Alloc</b>	Allocation.
<b>RI</b>	Retail Insights.
<b>ReIM</b>	Retail Invoice Matching.
<b>SIM</b>	Store Inventory Management.
<b>ReSA</b>	Retail Sales Audit.
<b>RWMs</b>	Retail Warehouse Management System.
<b>FSCA</b>	Fortify Static Code Analyzer.

---

# Contents

Certificate	iii
Statement of Originality	iv
Acknowledgements	v
Abstract	vi
Abbreviations	vii
List of Figures	xi
<b>1 Introduction</b>	<b>1</b>
1.1 General Introduction to the topic . . . . .	1
1.2 Area of Computer Science . . . . .	2
1.3 Motivation . . . . .	2
1.4 Objective . . . . .	2
1.5 Hardware and Software Requirements . . . . .	3
1.6 Scope of the Project . . . . .	3
<b>2 Literature Survey</b>	<b>5</b>
2.1 Background Technology . . . . .	5
2.1.1 WebLogic Server . . . . .	5
2.2 Continuous Integration . . . . .	8
2.3 Server Sent Events . . . . .	9
2.4 SSH Authentication . . . . .	9
2.5 Fortify Static Code Analysis . . . . .	11
2.5.1 Types of Analyzers inside SCA . . . . .	11
2.5.2 Steps of Fortify SCA . . . . .	12
<b>3 Methodology</b>	<b>14</b>
3.1 Introduction . . . . .	14
3.1.1 Issues With Manual Build Of Environment . . . . .	14
3.2 Automated Build . . . . .	15
3.2.1 Concept Behind Automating Environment . . . . .	15
3.3 The Methodologies for these tools can be outlined as follows: . . . . .	16
3.4 Design . . . . .	16
3.4.1 Internal Tools Development . . . . .	16
3.4.2 Use Case Diagram . . . . .	18



3.5	Easy Data Tool . . . . .	19
3.6	End-To-End Smoke Test Automation Tool . . . . .	21
3.7	Host Monitoring Tool . . . . .	22
<b>4</b>	<b>Implementation</b>	<b>24</b>
4.1	Qualitative Product Deployment . . . . .	24
4.1.1	Overview of the Shell Script: . . . . .	26
4.2	Easy Data Tool . . . . .	27
4.3	End-To-End Smoke Test Automation Tool . . . . .	28
4.4	Host Monitoring Tool features and Results: . . . . .	31
<b>5</b>	<b>Implementation and Results</b>	<b>32</b>
5.1	RSB (Retail Service Backbone) Deployment Automation Tool Results . .	32
5.2	Easy Data Tool . . . . .	37
5.3	End-To-End Smoke Test Automation Tool Results: . . . . .	42
5.4	Host Monitoring Tool Results: . . . . .	43
<b>6</b>	<b>Conclusion and Future Work</b>	<b>48</b>
	<b>Bibliography</b>	<b>49</b>

# List of Figures

2.1	WebLogic Architecture . . . . .	6
2.2	WebLogic Domain . . . . .	8
2.3	SSH Output . . . . .	10
2.4	Vulnerability Reports . . . . .	13
3.1	Flow of DSS work . . . . .	15
3.2	Basic architecture of the RSB Automation Tool . . . . .	17
3.3	Usecase Diagram showing the interaction between user, client and server	18
3.4	Web Application Architecture . . . . .	19
3.5	Database Server Overview . . . . .	20
3.6	Relationship of DB Entities . . . . .	21
3.7	Host Monitoring Module flow Diagram . . . . .	23
4.1	Implementation flow of RSB Automation Tool . . . . .	25
4.2	Relationship of DB Entities . . . . .	27
4.3	Easy Data Tool Operations . . . . .	27
4.4	Model View Flow of the End -To-End Smoke Test Automation Tool . . . .	29
4.5	Host Monitoring Tool Operations . . . . .	31
5.1	Login screen of the RSB Tool . . . . .	32
5.2	Add Host Page . . . . .	33
5.3	RSB Installation Details Page . . . . .	34
5.4	RSB Deployment completion status . . . . .	34
5.5	RSB Application Login screen . . . . .	35
5.6	Invalid Host Credentials Error . . . . .	35
5.7	Database Server/ Application Server Connectivity Error . . . . .	35
5.8	Invalid Installation Directory Error . . . . .	36
5.9	Missing Zip files Error . . . . .	36
5.10	WebLogic Domain Errors . . . . .	36
5.11	Easy Data Tool Main Frame . . . . .	37
5.12	Adding Single Host . . . . .	38
5.13	Adding Multiple Host though file . . . . .	38
5.14	Update Host Details Page . . . . .	39
5.15	Critical Reports Page . . . . .	39
5.16	Decommissioning Host Page . . . . .	40
5.17	Decommissioning the Database Instance On a particular Server . . . . .	40
5.18	Report Builder Page . . . . .	41
5.19	DB/APP Report Builder Options page . . . . .	41
5.20	User Requirement Report based on server type . . . . .	41
5.21	List of Tables used in the tool . . . . .	42

5.22	Execution result of the Tool . . . . .	42
5.23	Input Sheet Structure . . . . .	42
5.24	Output Sheet Structure . . . . .	43
5.25	Output sheet Path Details . . . . .	43
5.26	Host Monitoring Tool Login Page . . . . .	44
5.27	Login Page Operation . . . . .	44
5.28	List of Domains in a Host . . . . .	45
5.29	Status of the selected Domain . . . . .	45
5.30	Properties Page Contents . . . . .	46
5.31	Start/Stop the select managed server . . . . .	46
5.32	Managed Server Logs Page . . . . .	46
5.33	Server Log Contents . . . . .	47

# Chapter 1

## Introduction

### 1.1 General Introduction to the topic

Now a days, Retail product business is growing at a very fast pace. Many companies are involved in the race of developing of software products so that this business can be moved ahead with a very good pace with fewer troubles and Oracle is also a giant participant in this retail business product development. Oracle is involved with many retailer all over the world like “GAP”, “More mega Store”, “Shoppers Stop”, etc. Once these products are developed, it’s a very tedious job to deploy over servers for the high availability of these products. For easy access, high availability and less failure rate, these servers are kept world-wide at different locations.

Product deployment and implementation is a very important part of any business activity which is performed in order to expand the business, increase profits, consolidate, manage and utilize all the existing resources in an efficient manner. When a software product purchased from a particular company has to be deployed, possibly on a large variety of hardware and OS platforms, great care needs to be taken at each configuration step so that the process can be completed both safely and successfully. Deployments should be treated as part of a development workflow, not as an afterthought. Details of the staging environment, the server IP address, the database version, the environment variables to be set, etc. must all be noted and verified accordingly.

“Manual Vs Automatic” has been a controversial topic for ages in Social Sciences and recently (a couple of decades) in Computer Science. While there is no doubt that human

intervention cannot be completely eliminated from computer-based systems, automation can help during various phases of system development in order to reduce problems in the finished product. Software automation refers to the activities and efforts that intend to automate engineering tasks and operations in a software process using well-defined strategies and systematic solution[1].

## **1.2 Area of Computer Science**

Development Support Services is an integral part of every software company, including Oracle Retail. The DSS team here has major responsibilities of preparing the Oracle products for production. This includes taking care of Version Control, Setting up Application Environments, Configuring Application and Database Servers, Mock Installations and Deployments, and Building and Packaging the products for the final release. This project deals with the area of Deployments and Software Development, giving an insight to all the phases of the software development lifecycle.

## **1.3 Motivation**

Product deployment is a task which is being used many times in IT industries. So this process should be written in such a way that can improve reusability of code and improve the security of the files being transferred during deployment process from development environment to the real production environment. Files should be transferred in a secure way that no one can trace this transformation and make a change in the file developed by individuals or companies. This automation process reduces work load of a human being, error rate, etc.

## **1.4 Objective**

The main objective of this project is to reduce the work load of human being and the error rate in deployment process of products. The project work is basically carried out in two parts where the first part deals with Applications Product Deployment of the various Oracle Retail products for the verification and validation of the installers. the second part deals with the Automated Server Tools Development which is helpful in the server configuration and monitoring for the Development Support Services team.

## 1.5 Hardware and Software Requirements

### Hardware Requirements:

<b>Operating system</b>	: Windows 7 or higher and Red Hat Linux 6
<b>Display resolution</b>	:1024x768 or higher
<b>Processor</b>	:2.6GHz or higher
<b>Memory</b>	:1GByte or higher
<b>Networking</b>	:Intranet with at least 10Mbps data rate

### Software Requirements:

<b>Oracle (Sun) Java Runtime Environment</b>	: JDK 1.7.0+ 64 bit
<b>Browser</b>	:Microsoft Internet Explorer 11 or Firefox
<b>Database Server OS</b>	:Red Hat Enterprise Linux 6 for x86-64r
<b>Database</b>	:Oracle Database EE 11g and 12c
<b>Application Server OS</b>	:Red Hat Enterprise Linux 6 for x86-64r
<b>Application Server</b>	:Oracle Web Logic Server 11g and 12c

## 1.6 Scope of the Project

As the market is going in a high raid way the products should be certified before it released to the customer. To do this the same product need to be deployed in various environment like QA,EIT, SOLARIES etc. for testing purpose.[2] By doing the Environment automation the creation time to build an environment will be get reduced, hence resource utilization will be done in a good way.

Successful verification and pre-mock deployments of different Oracle Retail products on specific environment and architectural platforms helps the team in easing up the detection of possible issues that can arise during deployments. The errors obtained during process is recorded in the correction sheets in detail, which eases the debugging process and fixing them at the earlier stage before the customer faces it after the product is released into the market.

- The RSB Deployment Automation Tool minimizes the time required for the RSB installation and deployment process.
- Easy Data Tool is a database tool for maintaining and monitoring backups of all

database servers on a day to day basis.

- End-To-End Smoke Test Automation Tool is mainly used for monitoring the status of the different Oracle Retail Application at the server downtime period.
- Host Monitoring Tool provides a secured java secure channel to enable application teams to Restart Servers and manage outage in case of upgrade/fix deployments at individual components.

# Chapter 2

## Literature Survey

### 2.1 Background Technology

This chapter gives us the background view to the project topic. The first part of this project aims to conduct a qualitative product deployments and executing them on specific hardware and architectural platforms, by following the install guide and provide feedback to make the entire procedure easy and error-free, even for those who are not familiar with the technologies. It involved the use of various software tools including Oracle Databases and WebLogic Servers of different versions that is 11g and 12c and Linux OS Environment.

Oracle Fusion Middleware is a comprehensive family, which is a collection of standards-based software products that spans a range of tools and services from Java platforms and developer tools, to integration services, identity management, business intelligence, and collaboration. Oracle Fusion Middleware offers complete support for development, deployment, and management.

#### 2.1.1 WebLogic Server

The WebLogic application server is an integral part of both the divisions of the project – deployment and development. For the mock deployments, it is an important part of the pre-requisites to create new servers inside a particular WebLogic domain to host the retail applications and then configure them accordingly. For the tools developments, once the tools are completed, it is required to be hosted on a WebLogic server to make it accessible across the entire team and any other users.



### Various types of servers:

- **Database Server:** Allows user to send request for a data in a particular way called SQL statements and returns record of data as per the requirement of the user.
- **Mail Server:** Transfers and stores mails over the network using LAN, WAN and across the internet.
- **FTP Server:** Allows secure file transferring between computers by ensuring security and protocol.
- **Web Server:** It gives static content to a web browser by loading a file from a disk and transferring it across the network to the user's web browser. This exchange is intermediated by the browser and the server, communicating using HTTP.
- **Proxy Server:** Act as a mediator between client and external server and improves performance and filters requests.
- **Application Server:** It acts as a middleware component for client and database servers and also applies some business logic to improve the performance of the application.

### Architecture of WebLogic

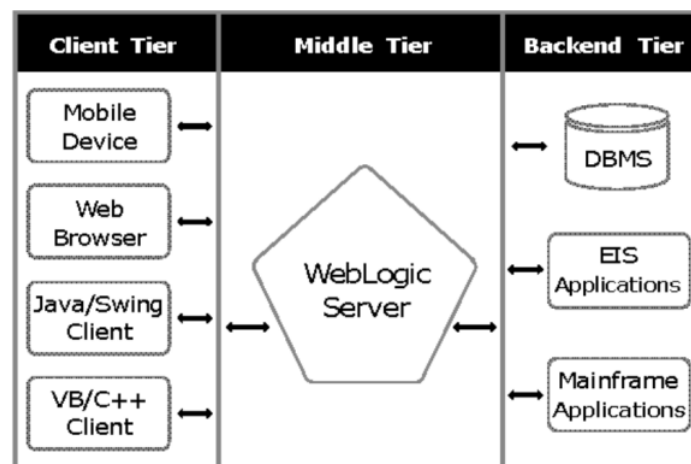


Figure 2.1: WebLogic Architecture

- **Client Tier:**

- Includes Mobile device, Web browser and other client programs.
- WebLogic provides standard interface to access its functionalities.
- WebLogic allows HTTP or HTTPS protocol to communicate with it.

- **MiddleWare Tier:**

- Web Logic Server itself act as a middleware between client and back end services like Database, Java Message Services, etc.
- Restrict clients to directly access backend services.
- Improves scalability, availability and reduces failure rate.

- **Back-end Tier:**

- Includes Database server, JMS server, JAVA Transaction API, etc.
- Provides data required by the users from database.
- Database server opens JDBC connections pool to be used by all application deployed over that server, which improves the application response.

## **WebLogic Domain**

- WebLogic server comes with several software pieces called as Admin Server, Managed Server, and Node manager.
- All these softwares all together is called as WebLogic domain.
- These components can be managed by GUI as well as command line console.

- **Admin Server:**

- It's a central point through which user can configure and manage all resources inside domain.
- Each domain consists of exactly one instance of WebLogic server acting as Admin Server.

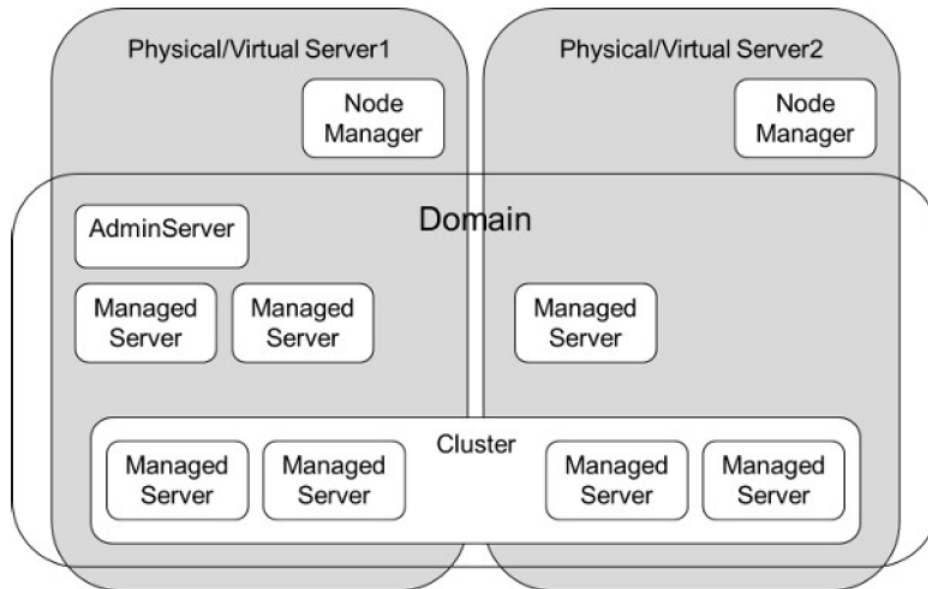


Figure 2.2: WebLogic Domain

- Application can be deployed and run over this admin server.
- **Managed Server:**
  - Domain can have zero or more domain server.
  - It's an instance of WebLogic server and is managed by Admin Server.
  - It can be clustered with other managed servers for availability, scalability and automation.
  - Applications are deployed over managed server for higher scalability.
- **Node Manager:**
  - It is a separate process to start, stop or suspend servers on its machine.
  - It is available one per machine.
  - it can be used to migrate servers from failed machine to another machine.
  - It is optional but recommended for application's high availability.

## 2.2 Continuous Integration

Continuous integration is a DevOps software advancement practice where developers frequently combine their code, roll out improvements into a central repository, after that

builds are automatically generated and test cases are run. The key objectives of continuous integration are to discover and address bugs faster, enhance quality of softwares, and reduce the time it takes to validate and release new software updates.

There are many tools available for this process Ansible, Git, etc. But all these tools have drawbacks some or the other way.

- A lot of problems in the tracking changes made by the various members of the development team.
- Securing access to the machine and regularly updating to patch the latest OpenSSL vulnerabilities.

## 2.3 Server Sent Events

One of the automation tools – the RSB re-deployment tool – uses a HTTP technology known as Server Sent Events. Server-Sent Events are real-time events emitted by the server and received by the browser.

The paper “Server-Sent Events with Yaws” provides great insight to the importance of HTTP polling at the client’s side and the web applications’ adaptation to Server Sent Events. It also points out the advantages of SSE over other HTTP polling methods such as HTTP streaming, long polling and WebSockets.[3]

## 2.4 SSH Authentication

SSH - Secure Shell is a protocol that allows user to control their remote servers over the internet. It is created as a secure replacement of telnet protocol for the communication to and from the remote server. All the application servers are installed on remote virtual machines. So it is a secure way to transfer product related files over the remote virtual machines. For this process user needs to be authenticated using their username and password on respective virtual machines. SSH provides a secure way to connect to virtual machine by either using a symmetric encryption method or using asymmetric encryption method. Symmetric encryption method is a bit insecure because it uses only one secret key to communicate between two parties which can be easily decrypted by brute force methods. Whereas asymmetric method uses two keys, private key and public

key and which introduces difficulty in cracking the message. So we are using key based authentication method means asymmetric authentication method.[4]

There are few steps to generate SSH keys and transfer between two remote machine:

- Generate SSH keys using RSA algorithm.

```
ssh-keygen -t rsa
```

- Give file name to store key otherwise it will store in home directory under `~/.ssh/id_rsa`.
- It will give output as shown in figure 2.4 after generating keys.

```
Your identification has been saved in /home/demo/.ssh/id_rsa.
Your public key has been saved in /home/demo/.ssh/id_rsa.pub.
The key fingerprint is:
4a:dd:0a:c6:35:4e:3f:ed:27:38:8c:74:44:4d:93:67 demo@a
The key's randomart image is:
+--[ RSA 2048]-----+
|           .oo.    |
|           . o.E   |
|          + . o    |
|         . = = .   |
|          = S = .  |
|         o + = +   |
|          . o + o . |
|           . o     |
|                   |
+-----+

```

Figure 2.3: SSH Output

- Public key is store in `~/.ssh/id_rsa.pub` file. Copy the content of this file.
- Create new file in another remote machine. `~/.ssh/authorized_keys` and paste the content of `~/.ssh/id_rsa.pub` over here.
- Then disable password login for the server then disable it via setting "PermitRoot-Login" to "without-password" in `etc/ssh/sshd.config`.

## 2.5 Fortify Static Code Analysis

- Reduce business risk by identifying vulnerabilities which cause the biggest threat.
- Identifies exploitable vulnerabilities quickly.
- Reduce development cost by identifying vulnerabilities early in the SDLC.
- Educates secure coding practices to developer while they work.
- Bring development and Security team together to identify and fix the bug.
- Dashboard shows key result of the application testing project[5].

Fortify consists of different types of analyzers, which will allow these different types of analyzers to install as per our convenience.

- **Fortify Static Code Analyzer:** It will analyze the whole code which is being taken inside the build process with specifically designed set of rules to provide the security inside the developed code base.
- **Fortify Runtime Application Protection:** It will monitor and protect the application which is already deployed over the application server against common attacks, unintended use of application, targeted hacking attempts, etc. In addition to this, it will provide protection against input verification, exception handling, etc.
- **Web Inspect Agent:** It identifies vulnerabilities before the application is being deployed over the actual production server during the QA activities and prevents security flaws before they exploited.

### 2.5.1 Types of Analyzers inside SCA

#### Buffer

It detects buffer overflow vulnerabilities that involve writing or reading more data than a buffer can handle. It will check whether or not there is a condition that causes buffer to overflow. If any path to buffer exists that can cause buffer overflow condition, it will point out the variable that cause the overflow. If that variable can be user controlled then SCA reports it as well and displays data flow trace to show how it can be tainted.

## **Configuration**

It will search for weakness, violation of policies in application deployment configuration. For example, it will check reasonable timeouts in user sessions in a web application.

## **Content**

It will search for the security violation of the HTML content. It will also search for dynamic HTML content such as PHP, JSP, etc.

## **Control Flow**

It will find out potentially dangerous sequences of operations. It will detect whether a set of operations are performed in a certain order or not.

## **Data Flow**

It identifies potential vulnerabilities in user control inputs that may cause vulnerabilities. e.g. It will check whether user controlled input string of unbounded length is copied inside statically bounded buffer or not, whether this string is used to construct SQL queries.

## **Semantic**

It detects potentially dangerous use of functions inside code base. E.g. It detects deprecated functions in java, unsafe functions in C or C++ such as `getc()`.

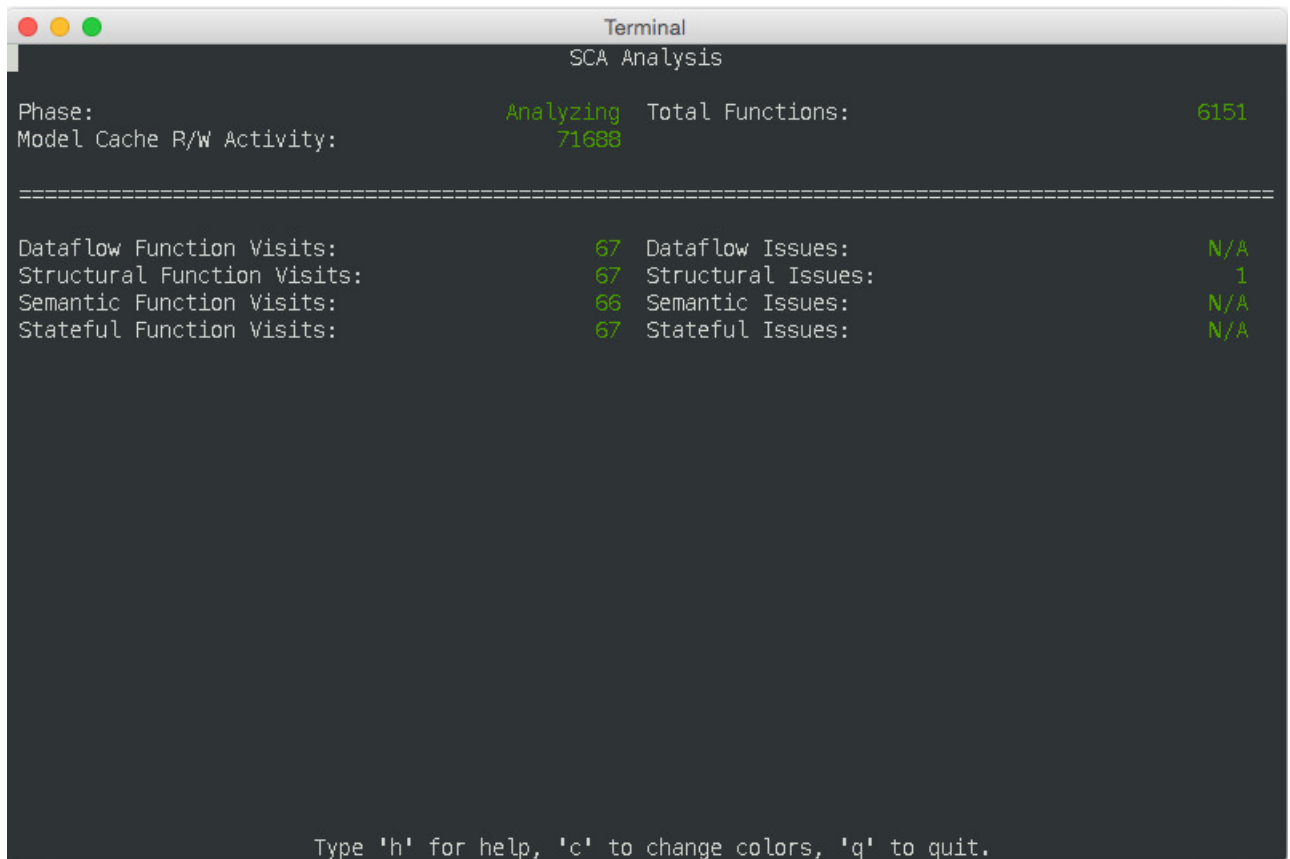
## **Structural**

It detects potentially dangerous flow in the structure of the application. It identifies violations of secure programming practices, by understanding the flow of the application which may not be detected during inspections of code base. E.g. loggers are not defined static or final, flags instances of dead code which are never executed because of certain conditions which is always false, etc.

### **2.5.2 Steps of Fortify SCA**

- It can be run as standalone process or it can be integrated as a part of build integration process.

- Translate the source code into intermediate code format.
- Scan this above translated code and produce security vulnerability report.
- Audit this above vulnerability report by transforming resulting FPR file to Fortify Software Security Center or directly with the result displayed on screen.



```
Terminal
SCA Analysis
Phase: Analyzing Total Functions: 6151
Model Cache R/W Activity: 71688
=====
Dataflow Function Visits: 67 Dataflow Issues: N/A
Structural Function Visits: 67 Structural Issues: 1
Semantic Function Visits: 66 Semantic Issues: N/A
Stateful Function Visits: 67 Stateful Issues: N/A

Type 'h' for help, 'c' to change colors, 'q' to quit.
```

Figure 2.4: Vulnerability Reports



# Chapter 3

## Methodology

### 3.1 Introduction

Agile software Development is a group of software development methods in which requirements and solutions evolve through collaboration between self-organizing, cross-functional testing. It promotes adaptive planning, evolutionary development, early delivery, continuous improvement and encourages rapid and flexible response to change. Based on the project the developer has to collect the necessary requirements from the customer and then start the development process. Here the project specifications are specified as per the team requirements and are explained further.

#### 3.1.1 Issues With Manual Build Of Environment

- **Time Consuming:** It takes approximately 2 to 3 weeks to completely build an environment.
- **More Resources Required:** A number of resource personnel are required to build an environment.
- Since it is done manually there are chances of more manual errors.
- **Fixing errors is comparatively time taking:** This is because building an environment itself takes around 2 weeks and if there is an error at some stage which was missed out then it again takes double amount of time to go back and fix that error and again continue with further steps.

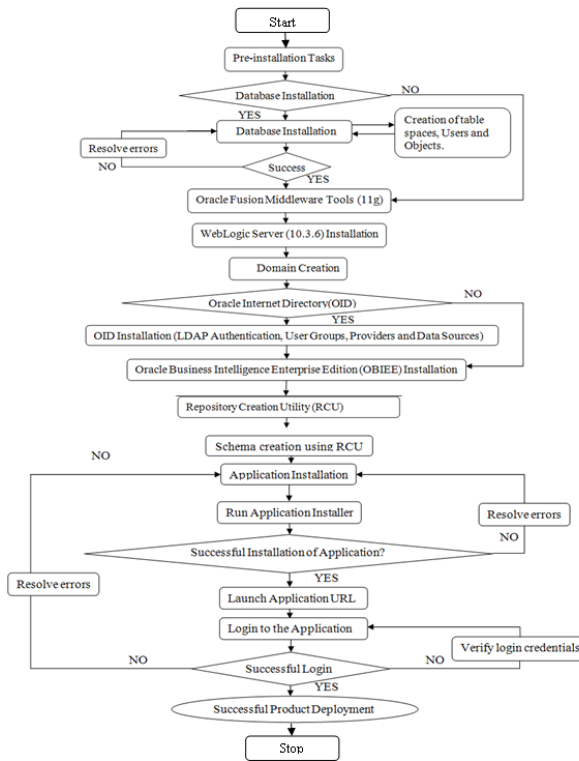


Figure 3.1: Flow of DSS work

## 3.2 Automated Build

### 3.2.1 Concept Behind Automating Environment

The main concepts which help in thinking about automating the environment build procedure are the following:

- **Data Available:** All the required data is available with different divisions of DSS team. This is because the manual build is completely done by different divisions of DSS team. So the team knows what all inputs are to be given at different stages.[6]
- **Standardized Process:** There can be a number of ways to install a product but to automate it we choose one standardized process. So out of many different ways that the DSS team uses one can choose one particular procedure and try automating it.
- **Complexity:** Complexity comes into picture in terms of following:
  - Data is not available at one single place. So it needs to be consolidated.
  - One standard procedure needs to be finalized and accepted by all.
  - The processes and concepts are not exactly written in any kind of document.

### 3.3 The Methodologies for these tools can be outlined as follows:

The oracle retail product RSB consists of four distinct phases, which will be automated in this tool. The four phases are as follows[7]:

1. **Download phase:** This phase involves the download of the required essential installation archive files of the latest RSB build from the SVN repository[8]
2. **Configuration phase** The tool is designed for the re-deployment of RSB in a particular environment. This means that the tool is developed with the assumption that the current environment already has RSB deployed on it previously. In such cases, the domain and server details remain the same. In the configuration phase, the file containing these details is copied into a folder to be accessed by the application later.
3. **Compilation phase** This involves setting up security credentials which creates/updates the wallet file. The credential and wallet details are also taken from the previous deployment.
4. **Deployment phase** This phase involves the deployment configuration tasks. It deploys all the RSB decorators, the rib4oms injector service and the RSB Admin App.[9]

Finally, all the admin, proxy and managed servers are re-started to ensure smooth transition.

## 3.4 Design

### 3.4.1 Internal Tools Development

**RSB Automation Tool** This tool is developed as a web application front using Java, JSP, Servlet and JavaScript. Thus, it uses MVC architecture where the Model is the backend shell script, the database while logging into the host and the Java class, the

Control is the servlet and the View is represented by the JSP pages and CSS.

From the Java class, shell scripts are invoked through JSch (Java Secure Channel) which uses the credentials provided by the database to login to the host. The shell scripts contain the operations required for the re-deployment of RSB, namely download, configuration, compilation, deployment phases and restarting of the servers. There is also an option where the user can resume deployment from a specific phase if an error occurs in any of the phases.

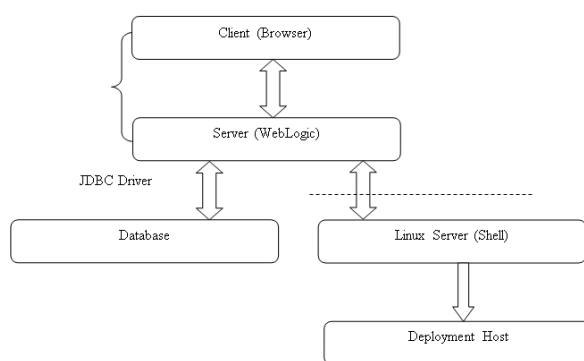


Figure 3.2: Basic architecture of the RSB Automation Tool

The block diagram above represents the basic architecture of the RSB Automation Tool. It is client-server architecture, with an interaction between the web browser and the backend servers. There are three servers involved here.

The first server is the WebLogic server on which the tool itself is deployed. The second server is the database server. This server contains a database wherein there is a list of all the RSB hosts and its details (credentials). Once the user enters the name of the RSB deployment host in the Start page, it is sent to the database to match to the corresponding host and then JSch is used to login to the server using credentials retrieved from the database.

The third server involved here is the deployment server itself into which the user logs in. This is the server on which the RSB re-deployment takes place.

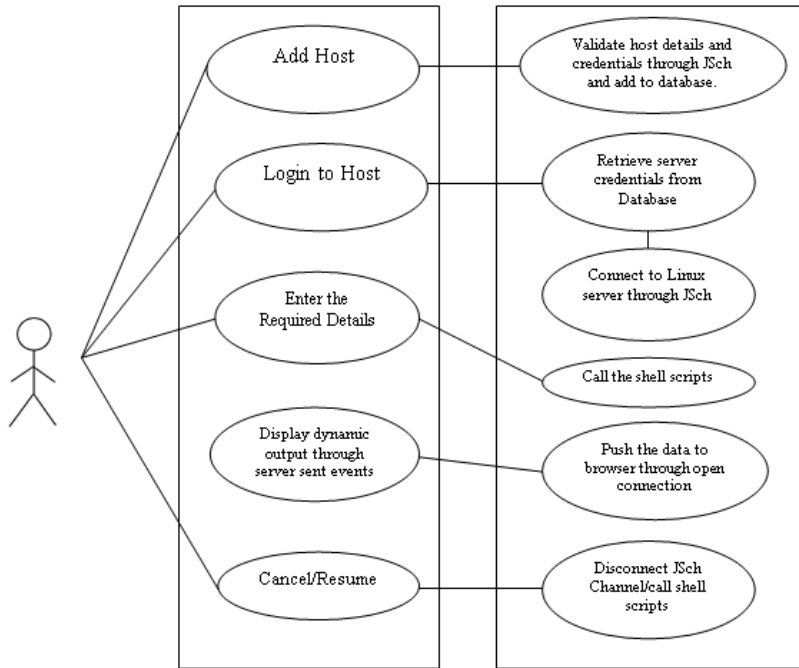


Figure 3.3: Usecase Diagram showing the interaction between user, client and server

### 3.4.2 Use Case Diagram

The use case diagram above describes the various actions associated with the tool and the interactions between the user, client and the servers. The user has the option to add the RSB host to the database. Here, the host details – host name, username and password – are entered and validated through the JSch channel. If the connection to the channel is successful, then the host details are valid and added to the database.

The required details to be entered by the user include the version of the RSB to be installed and deployed, the environment name, the URL of the SVN repository where the RSB archive files are stored, the SVN credentials and the WebLogic server credentials.

The user has the option to cancel execution in between, which will result in disconnecting the channel and thus effectively terminating any of the scripts executing in that particular channel. There are several validation checks – checks for the version, environment name, URL and the credentials.

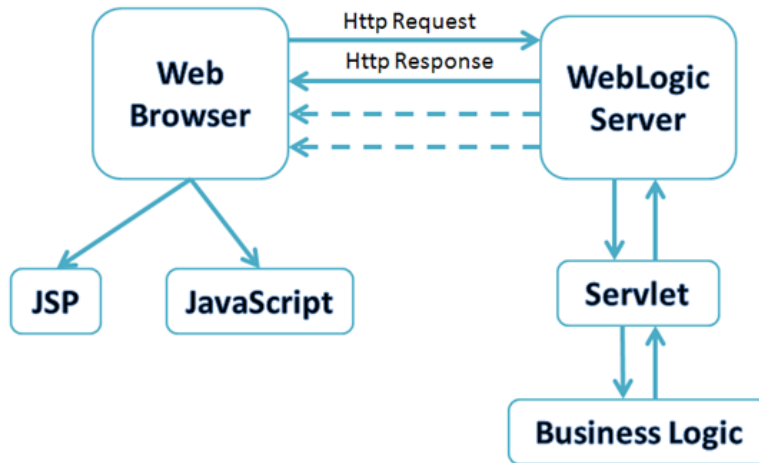


Figure 3.4: Web Application Architecture

### 3.5 Easy Data Tool

As prior to proceed further our team consists of three sections they are Build, Application and Database so as to help the database team to make their life ease. Developed a database tool. Let's discuss in brief of the methodology applied in this tool.

Easy Data Tool is a database tool useful for maintaining and monitoring backups of all database servers on a day to day basis. The main features of the tool are as follows:

- Generate daily backup reports based on the type of backup script available for the all the specified servers.
- Maintain and modify the backup report generated.
- Obtain required backup reports generated at a specific time.
- Check whether any backup report is missing due to some database issue.
- Add/Update new server details through user interface or by uploading the file.
- Deletion of specific servers.
- Decommissioning of the servers based on the instances.

While maintaining the backup of servers there are many factors which needs to be taken care based on the server category, attributes and type of backup. Database Server

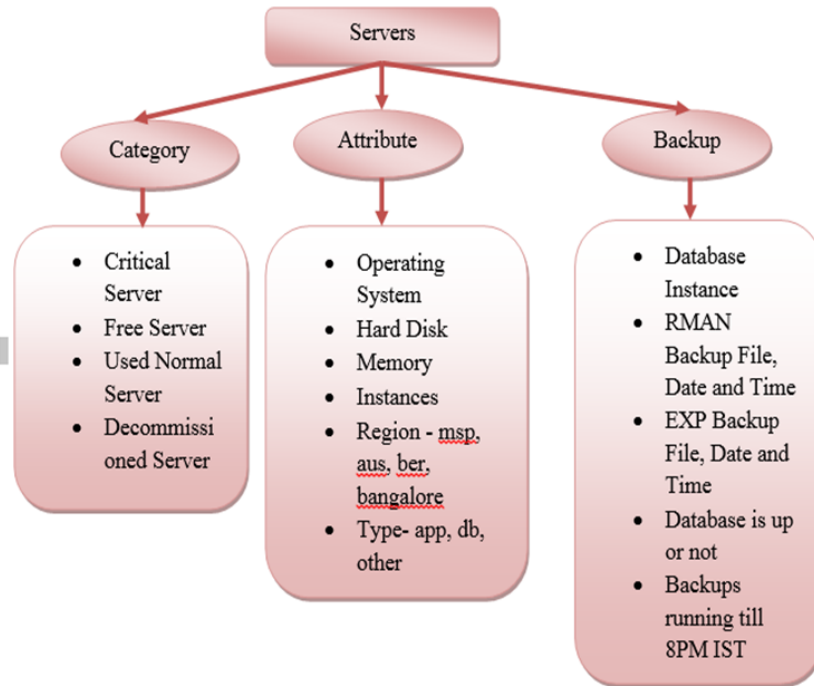


Figure 3.5: Database Server Overview

Overview diagram depicts the various factors of the Database server where critical servers are the servers which are in use and has to be maintained by the team, while free servers are the ones which are not in use currently, but can be used in future.

The server details are stored in the form of tables in the database used by the tool. As a result SQL Developer tool is used for the creation of the tables and connection to the respective servers in a specific database.

The tool is also involved in the process of decommissioning the respective servers /databases based on the user requirements. While decommissioning any of the server/database we have to update the main database table for its entries in it. For this a shell script is written at the backend known as BACKUP SHELL SCRIPT which will be executed when the decommissioning is done and the main table is updated.

Similarly we have NEW HOST SHELL SCRIPT for adding a new host to the main table. Here the user can also update the environment and wiki details of any particular server to which it belongs. The entire flow of the process is depicted in the Relationship of DB Entities diagram.

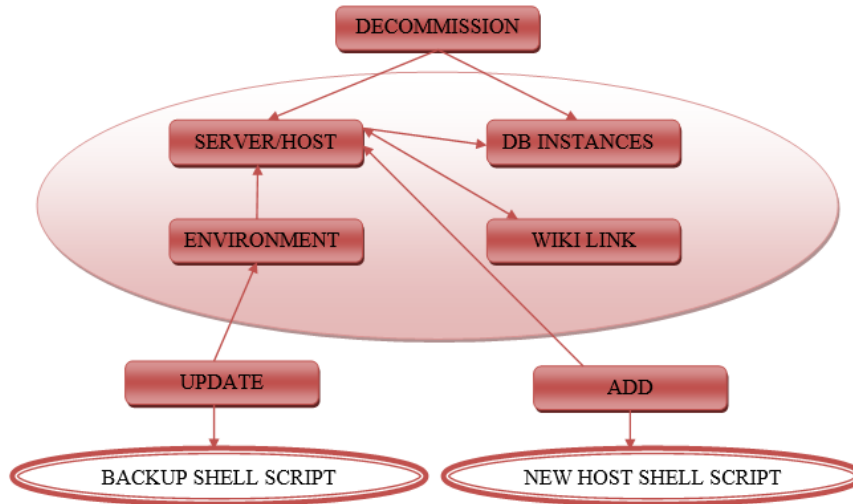


Figure 3.6: Relationship of DB Entities

Since the tool is a database tool, only database servers are involved in it. As we know the maintenance of database servers is a very important task in the industry because of its severity in proper functioning of any of the applications. In order to maintain all these database servers periodical backup of the servers has to be taken in case of any accidental failure of the database objects due to some issues which is taken care by the tool.

Now we can proceed further for the implementation of the Easy Data Tool in the chapter 4 and chapter 5

### 3.6 End-To-End Smoke Test Automation Tool

Oracle Application Testing Suite (OATS) is an integrated, comprehensive Web application testing solution that provides all the tools you need to ensure the scalability and reliability of your business-critical applications.

The End-To-End Smoke Test Automation Tool is developed using OATS tool. The tool implemented is mainly used for monitoring the status of the different Oracle Retail Applications at the server downtime period. During the server downtime all the application servers dedicated for each of the Oracle Retail Application is shutdown for either upgrade/maintenance activity and rebooted to apply the changes to it. At this time in order to know whether all the application servers have been rebooted properly or not, is a very tedious task. But with the help of the tool it is very easier. Since the tool outputs the status of the servers in an excel sheet, where the user can make out which



applications are running fine and which of them are facing issue.

The tool uses 2 different hosts which are used for running the 2 different versions of the Oracle Retail Applications i.e. 13 and 14 versions. Following functionalities are involved in the implementation of the tool:

- Creation of test scripts.
- Creation of Object Libraries for different applications.
- Input and Output excel sheets.
- Mailing the generated results output sheet,

The complete functioning of the tool at the server end and client end is displayed in the model view diagram in the implementation section.

### **3.7 Host Monitoring Tool**

The Host Monitoring Implementation tool developed is based on monitoring different domains created within the WebLogic server by the various applications deployed. This application gives an easy GUI for:

- Listing all the Domains in the specified application server host after successful login.
- Listing all the Managed Servers of any Domain.
- Starting and Stopping all/some/one Managed Servers in any Domain.
- Checking the Log of Managed Servers and Admin Servers.
- Displaying the status of the selected domain along with the port number on which it is running.
- Modifying the properties file of the WebLogic server.

In the Host monitoring application as mentioned before we perform various operations with respect to the domains. The working of the application goes as shown in the module flow diagram.

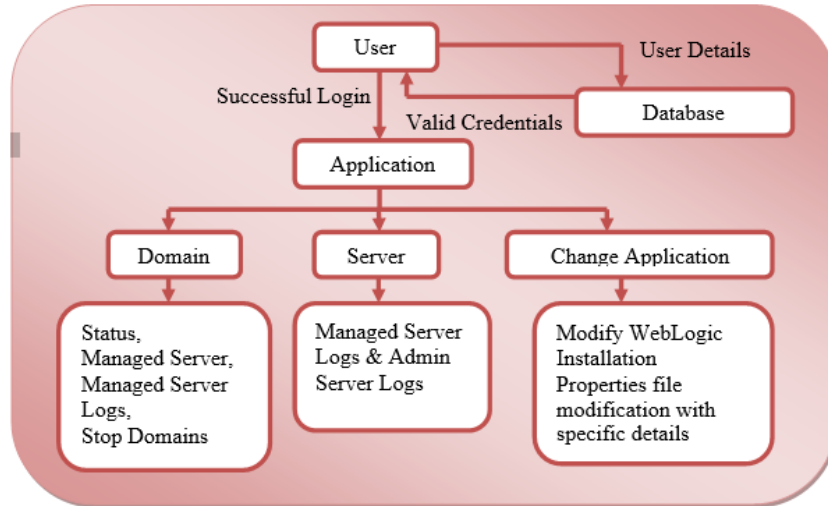


Figure 3.7: Host Monitoring Module flow Diagram

The tool uses multiple Java Secure Pages (jsp) pages to display the different functioning of the domains in the host. Since jsp uses Http session to transfer data between the different pages, Java Secure channel is used for secured session for executing the scripts in the UNIX server and for secured transfer of data from jsp page to java functions and transmit back the processed data to the jsp function call.

Using the tool the user can view the list of the domains in a particular server. The tool gives a provision for viewing the status of each specific domain once it is selected. A single domain will consists of an admin server and 1 or more managed servers which can be started or stopped based on the functionalities provided. The user can also view the logs of the managed servers and the domains by selecting the managed server logs options. The properties file of the weblogic installation in the host consist details of all the domains created and the environment details, which can be modified by the user by opting for CHANGE APPLICATION PROPERTIES options. Here the user can modify the properties file details like environment information, JDK home, Mailing List, Database details, Application user details, WIKI link details etc.

# Chapter 4

## Implementation

### 4.1 Qualitative Product Deployment

Product Deployment is a methodical procedure of introducing the process of deploying the different oracle retail products in various application and database servers by the customers for their business process management

Multiple steps are involved in the product deployment process. Before deploying any product, firstly we have to review the entire product for its configuration requirements in areas like network topology, file systems, software's, database resources and so on. Once all the requirements are met, we can proceed with the deployment process.

The design, steps and procedure involved while deploying the different Oracle Retail products on specific environment and platforms is shown with the help of the following flowchart.

- **Download:** In order to carry out the RSB installation successfully we are supposed to download few of the archive files and extract them into specified path in the installation directory.
- **Configuration:** Here the RSB Database objects are to be created by running few database scripts and updating the environment specific properties file.
- **Compilation:** Here the security credentials are set by running the RSB compiler

scripts.

- **Deployment:** Here the WebLogic Server has to be configured for the redeployment of the RSB application in the existing RSB domain.

## RSB Automation Tool

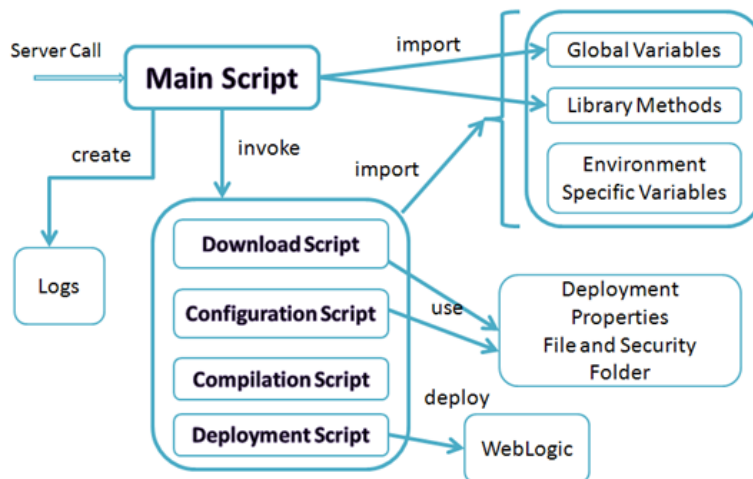


Figure 4.1: Implementation flow of RSB Automation Tool

The implementation flow of RSB Automation Tool is represented in the above diagram. The front end is a JSP page where the user can log into the host. The host will be matched with the database, credentials will be retrieved and the JSch channel will be established to the Linux machine. Once the user enters the required details, the Main Script will be called which in turn calls the four scripts – Download, Configuration, Compilation and Deployment. An additional script to restart the server is also called. While downloading, the files which are not present in the SVN repository are notified to the user via a browser alert.

There are few pre-requisites that are required to be fulfilled by the user before running this tool. In the mount path where the scripts exist, there is a folder named ‘ENVIRONMENT’. This folder consists of environment homes specific for specific environments. Environment is the environment/host where the redeployment is to be taken place. The input that the user enters in the ‘Installation Environment’ field must match this environment. From the existing deployment, a copy of the rsb-deployment-env-info.properties file and security folder must be taken and pasted in this folder. There is a configura-

tion file named 'configVar env environment.cfg' (Environment-specific variables), which is specific for a particular environment. Values in this file must be entered by the user.

GlobalApp.cfg (Global Variables) is also a configuration file which contains the definitions of a list of variables used in the redeployment. This need not be modified in any way. scriptFunctions.lib (Library Methods) consists of functions for file checks and displaying the missing files. There are two types of logs – RSB.log which records the script logs and rsb-deployment.log which records the logs of the redeployment.

Once the user click on 'Deploy' option, the JavaScript Event-JS.js executes which reads the form data, processes it and calls the Servlet which in turn will call the scripts through JSch. The HTTP connection is 'kept alive' during the script execution. As a result, the execution flow is displayed on the browser as well. This is achieved through Server-Sent Events. The user also has the option to Cancel/Resume the deployment.

#### **4.1.1 Overview of the Shell Script:**

- The RSB Deployment Automation tool scripts are located in a mount path from where it is executed.
- It consists of three primary scripts for the four phases of the RSB deployment, which are in turn called by a main script.
- There exists an environment specific folder which consists of environment homes specific for specific environments.
- There is a configuration file known global variables file which is specific for a particular environment. Values in this file must be manually set by the user.
- Library file consists of a function which checks for availability of files necessary for the RSB redeployment.
- A deployment information folder is imported from the previous successful RSB installation which consists of the deployment property details and the security credentials that is very important for the redeployment process.
- The Library file, Global Variable file, Environment Specific Variable file and the deployment information folder file are imported by the main script and the sub

scripts during the RSB installation process.

- Log file in the installation directory logs any errors that are captured during the redeployment process.

## 4.2 Easy Data Tool

Easy Data Tool is a tool for maintaining and monitoring backups of all critical database servers and the critical environment on a day to day basis by generating various reports at scheduled time. The backups generated are very helpful during the server failure/recovery time. Few of the features developed by the tool have been explained in the table.

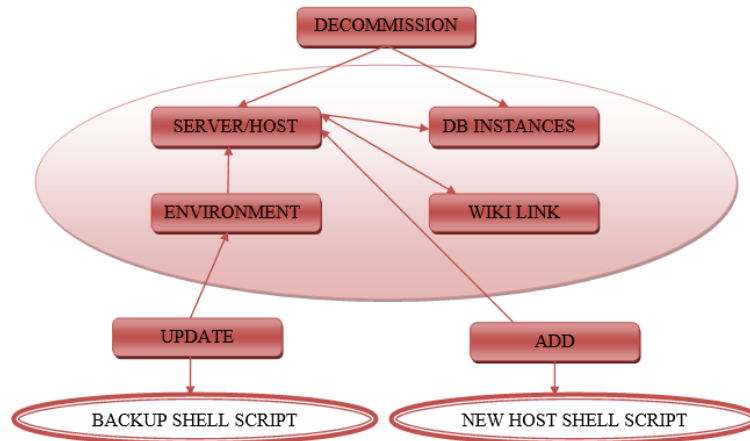


Figure 4.2: Relationship of DB Entities

Operations	Description
Add Host	Adding new host details to the database.
Add Host Through File	Adding multiple new host details to the database table through an input sheet in specified format.
Update Host	Updating the host details in the database.
Decommission Host	Deleting the server/ database instance from the database tables' entry.
Report Builder	Generate user requirement reports based on the server type, zone and OS (Operating System).
Critical Server Report	Generates Critical Environment Report and Backup report for the critical servers as per the scheduled time.
Mail Notification	Mails the generated critical and backup reports to the specified mail ids.

Figure 4.3: Easy Data Tool Operations

## 4.3 End-To-End Smoke Test Automation Tool

Following functionalities are involved in the implementation of the tool:

- Creation of test scripts.
- Creation of Object Libraries for different applications.
- Input and Output excel sheets.
- Mailing the generated results output sheet.

The complete functioning of the tool at the server end and client end is displayed in the model view diagram.

### **Creation of test scripts:**

Based on the type of applications separate test scripts are created with the help of the object libraries. The test scripts are invoked from the main script which will call each of the individual functions within it based on the inputs received from the input sheet. The inputs to the main test script functions are read from the input excel sheet maintained at the repository directory row by row which carries all the login credentials necessary to test the application status. While inputting the credentials multiple check constraints are present at various levels for validating the application objects displayed in the browser and finally the results of these check constraints are outputted as results in the console window. The scripts written for the various applications are as follows:

### **Web and Forms Application Test Script Functions:**

The script has the function for all the Oracle Retail web and forms applications. Ex: Retail Integration Bus (RIB), Allocation, Retail Warehouse Management System, Retail Merchandising System etc.

### **Object Library:**

Is the file created to capture the specific object for validation of the applications when it is launched.

### **Main Test Script**

This script is used to call the forms and web application functions that are present in the function library and object library.

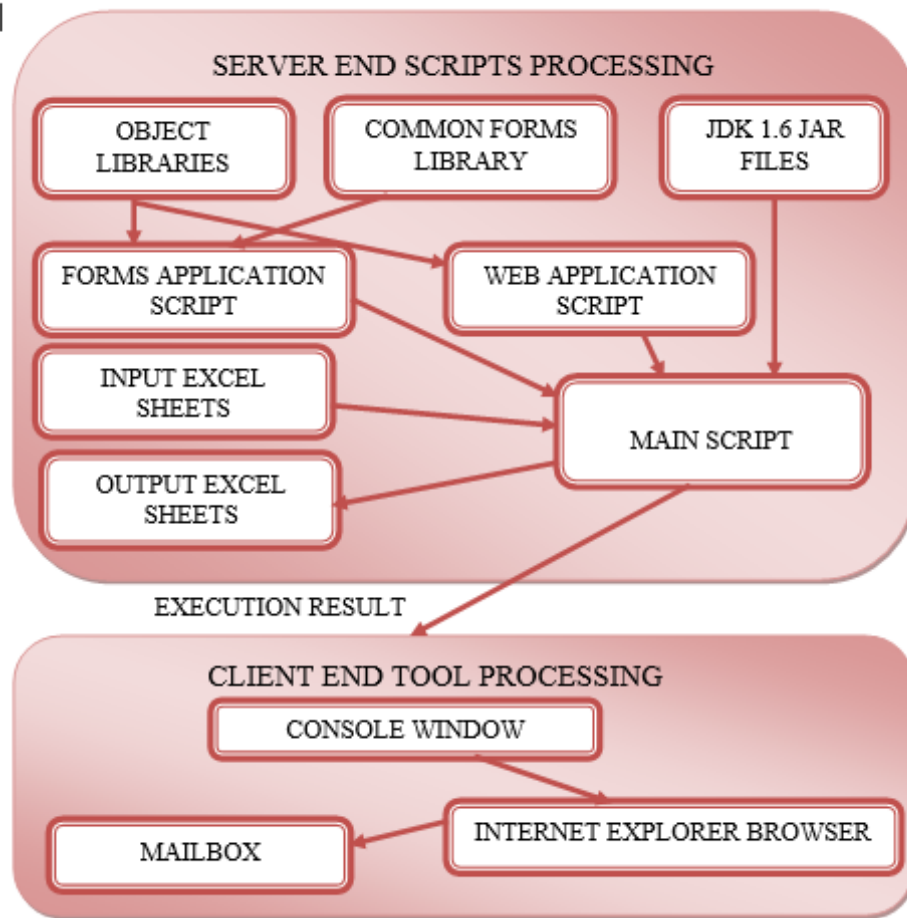


Figure 4.4: Model View Flow of the End -To-End Smoke Test Automation Tool

**Creation of Object Libraries for different applications:** The object libraries have to be created before generating the test scripts, because it is very important in order to validate the objects of various applications in the browser once the application is launched. After the creation of the object library all the application contents launched in the browser are recorded in the form of objects which are used in the validation process of the application status by the check constraints in the test scripts.

**Input and Output excel sheets:** The input sheet consists of list of all the applications along with their URL, Environment, Username, Password, Facility, Version and Result fields which is read row by row by the main test script and is passed to the respective functions based on the application name for validating its status. There are multiple versions in 13 and 14 versions like 13.1 PROD, 13.2 QA, 14.1 QA etc.

The results obtained after the execution of the scripts is saved in the Results column of the output excel sheet. The results can be either passed or Unknown Error/App is



down/Authentication Error/App is up but Unknown Error/Invalid Inputs/Java Version not up to Date/SSO Required.

**Results Description:**

- **Passed:** Here the URL is loaded and the user is able to login to the application successfully with the given credentials without any glitch. The user is also able to find the respective objects within the application login page.
- **Authentication Error:** Here the URL is launched successfully. But the credentials given are invalid as a result the tool is not able to login to the application.
- **App is Down:**This error occurs when the application URL is not launched because the application may not be up.
- **App is up but Unknown Error:**Here the URL is launched and the objects in the login page are also present. But when the credentials are inputted to the input objects and tried to login following errors might occur.  
The application takes very long to load the index page.  
Application gets hanged without displaying any error.
- **Invalid Inputs** Here the URL is launched successfully. But the credentials inputted by the input sheets are either invalid or null.
- **Java Version not up to Date:** Here the application is up but due to the incompatible java version the application URL is not launched.
- **SSO Required:**Here the application is up but the SSO credentials are not available in the input sheet provided.

**Mailing the generated results output sheet:** The results output sheet generated during the execution of the tool is mailed as an attachment to the respective mail ids of the specific user as specified.

## 4.4 Host Monitoring Tool features and Results:

The Host Monitoring Tool developed is basically for configuration and monitoring of multiple domains created within the application server by the various Oracle Retail applications deployed. Many operations are involved in the tool as described in the table.

<b>Operations</b>	<b>Description</b>
Login	Used to login to the specified host/server where the various domains have to be monitored.
Add Host	Adding new host details to the database if not available in prior.
Domains	Lists all the domains in the host.
Server Logs	Displays the Logs of the managed and admin server of a particular domain.
Domain Functionalities	Allows the user to start/stop the domains and its managed servers and also to view the domain status.
Change Application Properties	Allows modifying the properties file of the weblogic installation.

Figure 4.5: Host Monitoring Tool Operations

# Chapter 5

## Implementation and Results

### 5.1 RSB (Retail Service Backbone) Deployment Automation Tool Results

Steps involved in RSB Deployment Automation tool user interface is as mentioned below:

- 1.The developed application is launched using the URL present in the Application Server where the tool resides. The login page appears as shown in figure 5.1. Here the Host Name field is the name of the application server where you want to run the redeployment of RSB which must have an existing RSB deployment present.

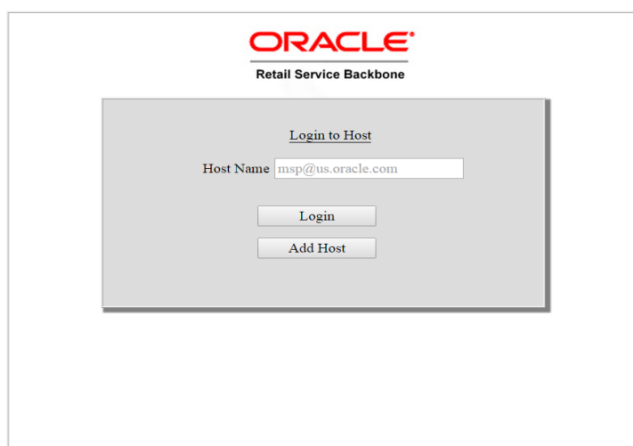
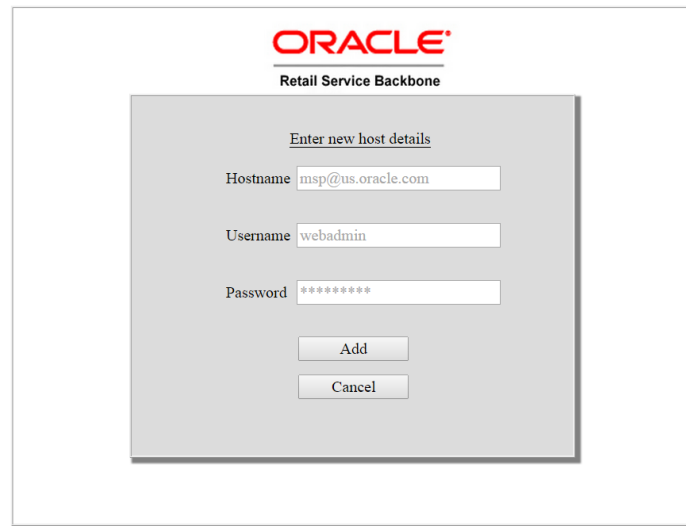


Figure 5.1: Login screen of the RSB Tool

- 2.If the server name the user wants to enter is already present in the database, then the user can directly login to the required server by just entering the server name in the text field and clicking the ‘Login’ button and go to Step 4.
- 3.The server name the user wants to enter is not present in the database, then the server has to be added by clicking on ‘Add Host’ button and after that the user can login into the newly added server through the login page.



ORACLE<sup>®</sup>  
Retail Service Backbone

Enter new host details

Hostname msp@us.oracle.com

Username webadmin

Password \*\*\*\*\*

Add

Cancel

Figure 5.2: Add Host Page

- 4.If the user has entered the server name and pressed ‘Login’, the user is redirected to the main inputs page which consists of various fields as shown in figure below. The Installation Environment field is the name of the environment where the RSB redeployment must take place, Zip Files Path refers to the path of the folder in which the RSB packaged files are present that includes all the zip files required for RSB installation, WebLogic Username and Password are the credentials of the WebLogic domain where the RSB domain exists, on which the redeployment is to take place. Once all these details are inputted the user has to click on ‘Deploy’ button to start the deployment process.



Figure 5.3: RSB Installation Details Page

- After the deployment process is completed successfully following screen is displayed.

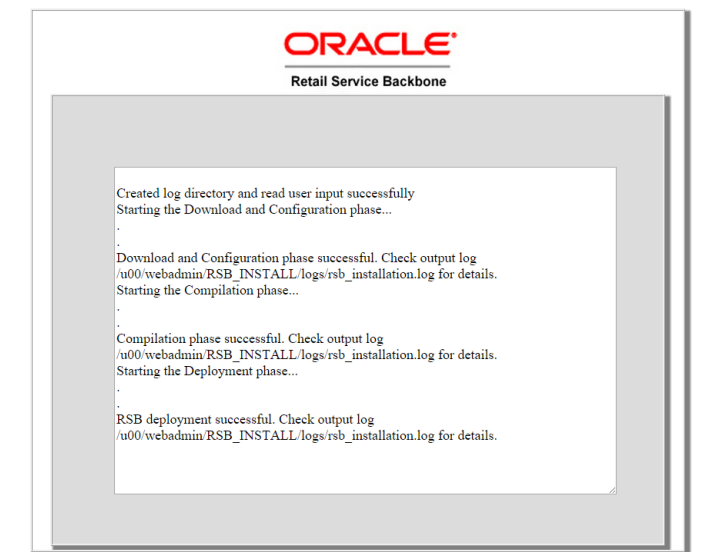


Figure 5.4: RSB Deployment completion status

- Once the deployment is successful the user can login to the RSB application by launching the redeployed RSB URL present in the RSB domain to get the screen as in figure 5.5

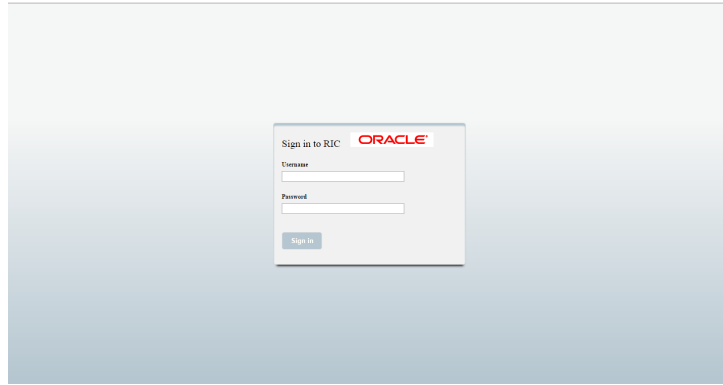


Figure 5.5: RSB Application Login screen

- While carrying out the RSB redeployment process the user can come across few expected errors, if any of the details inputted in the various fields are incorrect following error might occur as shown in the below figures.

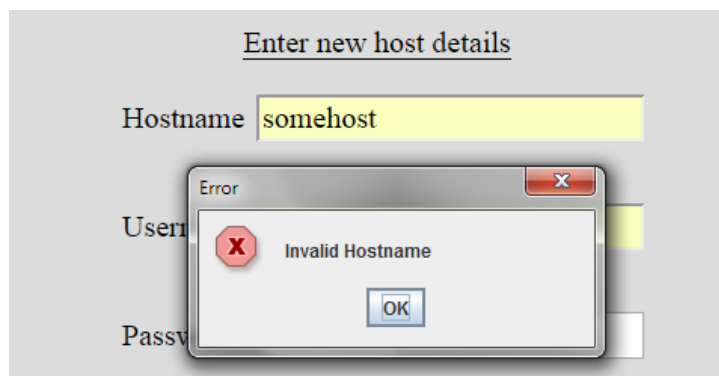


Figure 5.6: Invalid Host Credentials Error

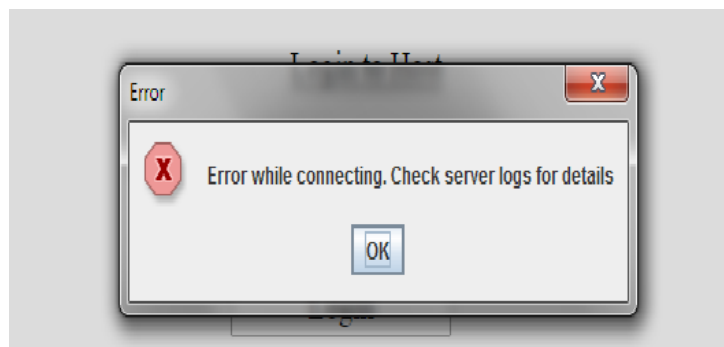


Figure 5.7: Database Server/ Application Server Connectivity Error



Figure 5.8: Invalid Installation Directory Error

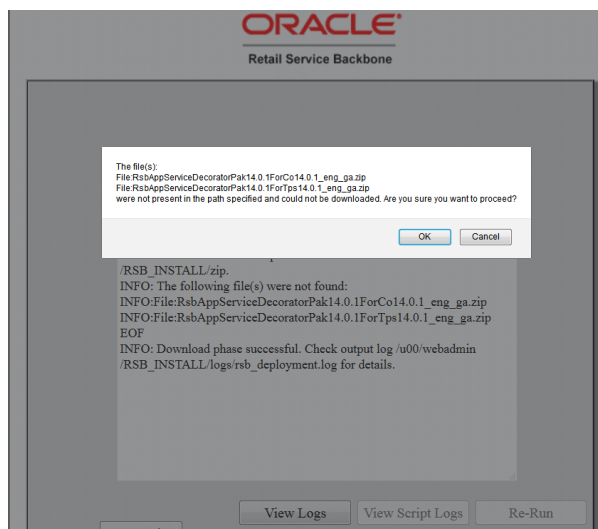


Figure 5.9: Missing Zip files Error

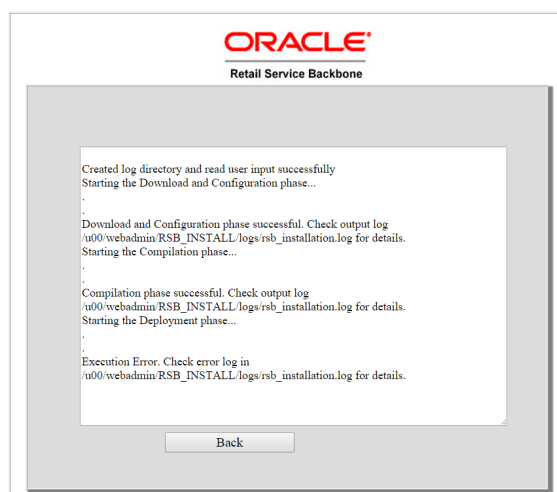


Figure 5.10: WebLogic Domain Errors

## 5.2 Easy Data Tool

Easy Data Tool is a database tool, as a result only database servers are involved in it. The tool mainly takes care of periodic critical server report generations, maintenance and monitoring of the servers by scheduling timely backups of the multiple databases based on the server's categories, attributes, type and priorities. Following operations are carried out by the tool:

- Adding new host to the database through file/User Interface.
- Updating the details of the existing host.
- Decommissioning servers.
- Generating Server Reports based on server type.

The above mentioned functionalities are shown in the below figure.

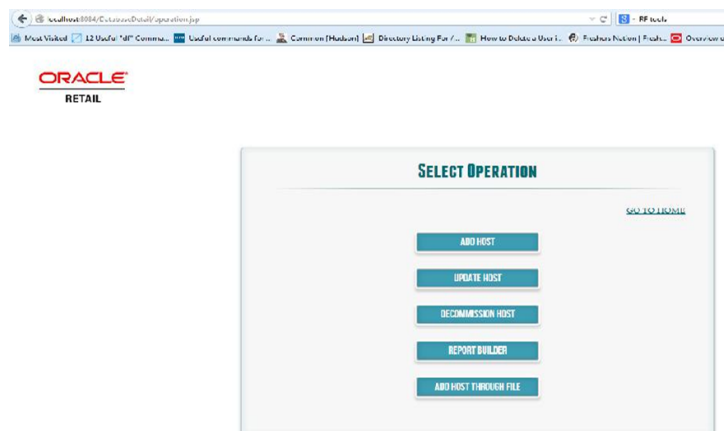


Figure 5.11: Easy Data Tool Main Frame

Adding new host to the database through User Interface: Here the user has to fill the details of the new host that is being added to the database for future use. As shown below once the user enters the necessary details of the new server being added and clicks on the save button for the operation to be completed.

Adding new host to the database through file:

As mentioned earlier if there are multiple hosts to be added to the database at a single shot then we can carry out the operation by uploading a file as shown in the below figure.



Figure 5.12: Adding Single Host

Only thing which has to be noted here is that the file should have all the columns in sequence.

Figure 5.13: Adding Multiple Host through file

While performing the add host operations if any of the host details are wrong or already present in the database then an pop up message is displayed on the screen to alert the user.

Updating existing host details:

For modifying any of the server details in the database we have to perform the update host operation. Here the user can update any server to critical, modify the environment details of the server and update the critical server details as depicted in the figure.

On selecting the UPDATE THE CRITICAL SERVERS BACKUP DB operation in update details page the user is redirected to critical server page where multiple options are displayed like.

- Generate Critical Environment Report.

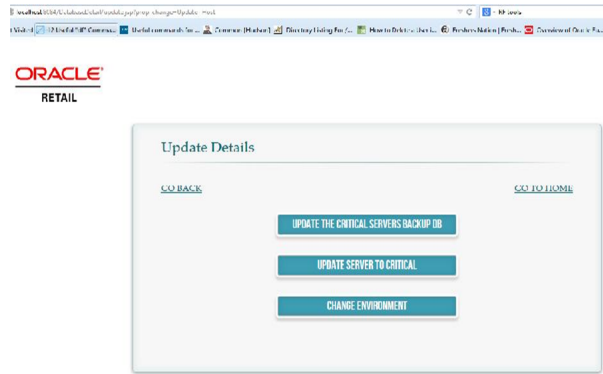


Figure 5.14: Update Host Details Page

- Generate Backup report for the critical servers.
- Generate current day's Backup report for the critical servers.
- Generate Backup report for the critical servers after 8AM IST.
- Mail the generated reports to respective members for monitoring.

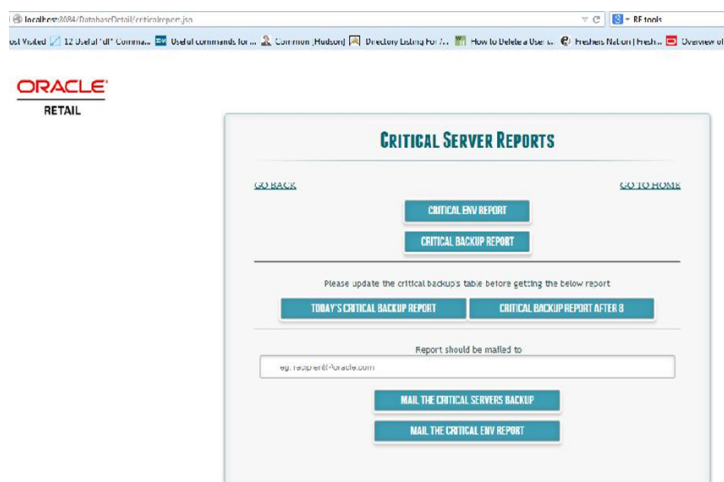


Figure 5.15: Critical Reports Page

#### Decommissioning Server:

Here the user can decommission the database/server based on two options i.e. directly the server can be decommissioned or delete specific instance on the server as shown in the below figures.

#### Generation of the Server Reports:

Here the user can generate reports based on the type of server. As shown in figure the different reports are:

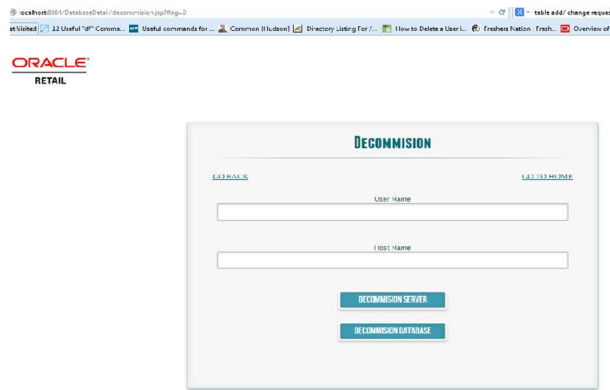


Figure 5.16: Decommissioning Host Page

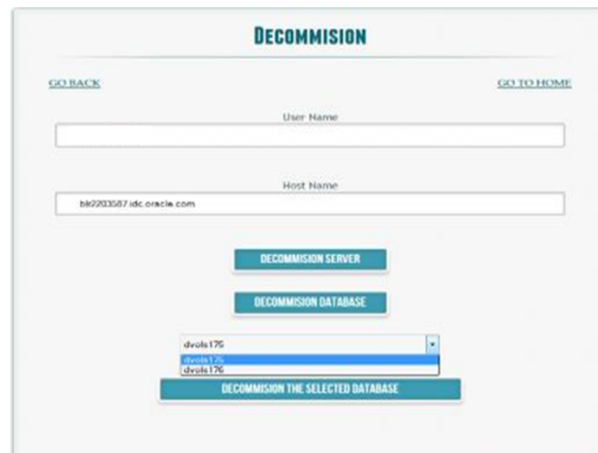


Figure 5.17: Decommissioning the Database Instance On a particular Server

- DB(Database Server)
- APP(Application Server)
- Critical Server Report
- Other Server Report

In DB and APP type the user can generate the reports based on the operating system and regions options as shown in figure.

As stated in the beginning this tool is a database tool and uses a database which will consists of all the tables created for the functioning of the tool. Following tables are involved in these tools:

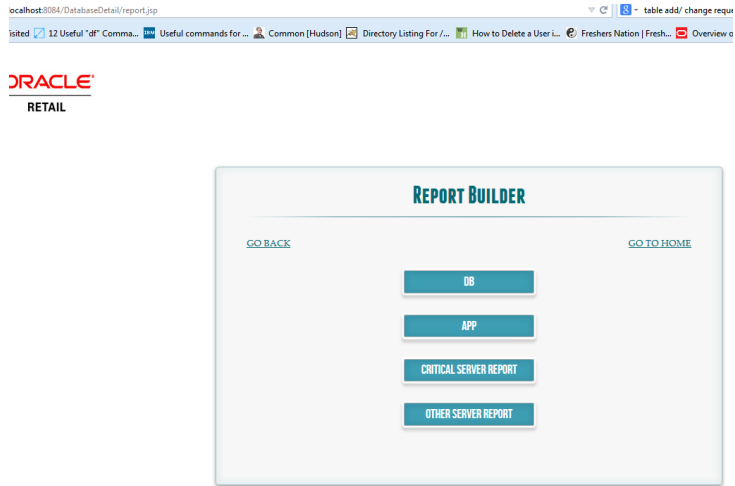


Figure 5.18: Report Builder Page

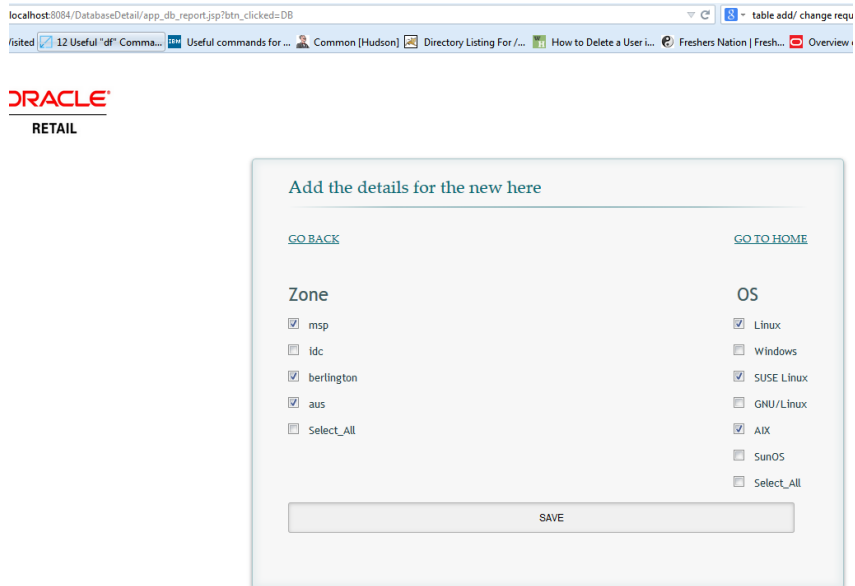


Figure 5.19: DB/APP Report Builder Options page

### CRITICAL SERVER ENV STATS

HOST_NAME	OS	CPU	MEMORY	HDD_U00	HDD_ROOT	TYPE	SUB_TYPE	REGION
Dlrprdvm001.idc.oracle.com	Linux	6	31371464	146	null	DD	null	idc
ISCV07.idc.oracle.com	Linux	2	4153608	19	19	DD	null	idc
ISCGRD01.idc.oracle.com	Linux	2	7987200	191	191	DB	null	idc
Blrdev02.idc.oracle.com	Linux	6	31948976	580	580	DB	null	idc
blr2202939.idc.oracle.com	Linux	2	11689092	171	171	DB	null	idc
blrdev09.idc.oracle.com	Linux	8	32615552	34	48	DB	null	idc

Figure 5.20: User Requirement Report based on server type



Similarly the results output sheet format generated along with the results after the script execution is as shown below:

Application	Environment	Uri	Username	Password	Facility	Version	Result
							Passed
							Passed
							Passed
							passed
							passed
							passed
							passed
							passed
							Passed
							Yet to do
							Passed
							Yet to do
							Passed
							Unknown Error
							App is Down
							Authentication Eirc
							passed
							Yet to do
							Yet to do
							Yet to do
							Passed
							passed
							passed
							App is Down
							App is Down
							App is Down
							App is down
							passed
							passed
							App is Down
							App is Down
							App is Down
							App is Down
							passed
							Yet to do
							App is Down
							App is Down
							passed

Figure 5.24: Output Sheet Structure

The input and output sheets path in the tool is present at a specific location as depicted in figure 5.23 and 5.24. If the user wishes to modify the sheet name he can do it over here. For the modification in the sheet details the user has to go to the specific path.

```

517
518 // for s
519 System.out.println("SheetList = " + sheetlist.toString());
520 datatable.exportSheets("C:\\OATSworkspace\\Data\\sheet_3_may_5.xls",sheetlist);
521
522
523 )
524
525
526 public String resu4(String url, String user, String pass) throws Exception{
527     browser.setBrowserType(BrowserType.InternetExplorer);
528     browser.launch();
529     String result = "";

```

Figure 5.25: Output sheet Path Details

## 5.4 Host Monitoring Tool Results:

Host monitoring tool is an application servers monitoring tool to Restart Servers and manage them during outage time. The various operations carried out by the tool are as follows:

- The tool URL is launched in order to load the login page of the application. Here the user is given options for logging into the existing host with the help of the LOGIN button and ADDHOST button for adding the new host. The user can login to

the host if it is already present in the database or else the new host can be added to the database by inputting the required details.

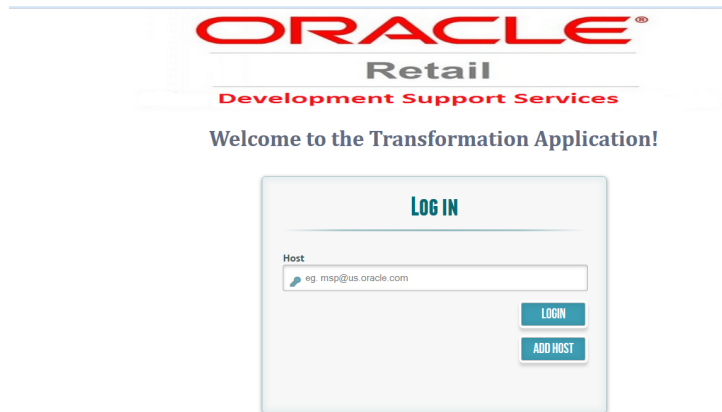


Figure 5.26: Host Monitoring Tool Login Page

After logging into the specified host the user can perform few of the operations like list various domains, view server logs and modify application properties file as shown in figure 5.27. Here on click of DOMAINS button user gets the list of all the available domains in the particular host and on click of CHANGE APPLICATION PROPERTIES button properties file of the weblogic installation can be modified with the necessary details by maintaining the backup of the previous version.

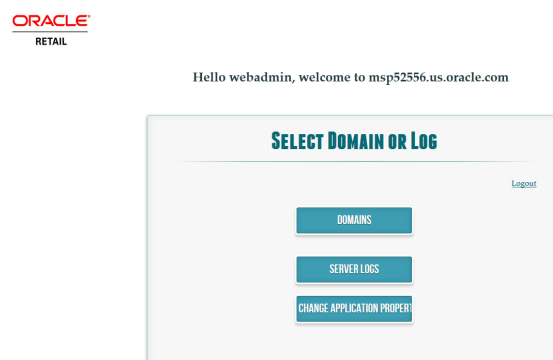


Figure 5.27: Login Page Operation

Here the user can perform various operations on the selected domains and view its status, logs, start and stop the servers in it and view the entire domain log or the managed server logs too.

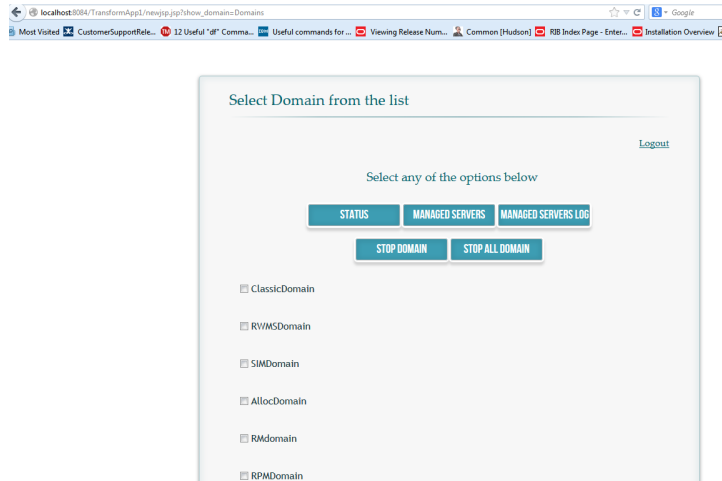


Figure 5.28: List of Domains in a Host

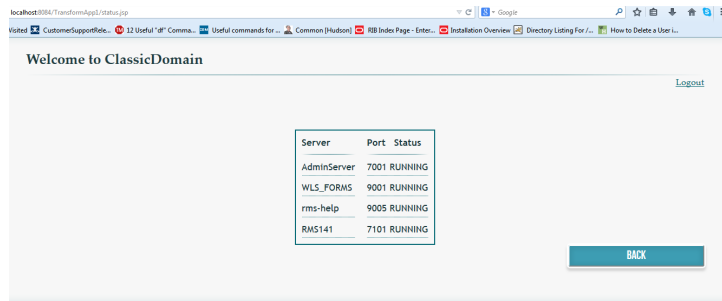


Figure 5.29: Status of the selected Domain

In the above figure if the user has selected a particular domain and clicked on the status button then the result displayed consists the list of the Admin server and all the managed servers within the selected domain.

Here in the properties page the user can modify the JDK home, DB/APP user details, wiki link details as per the requirement.



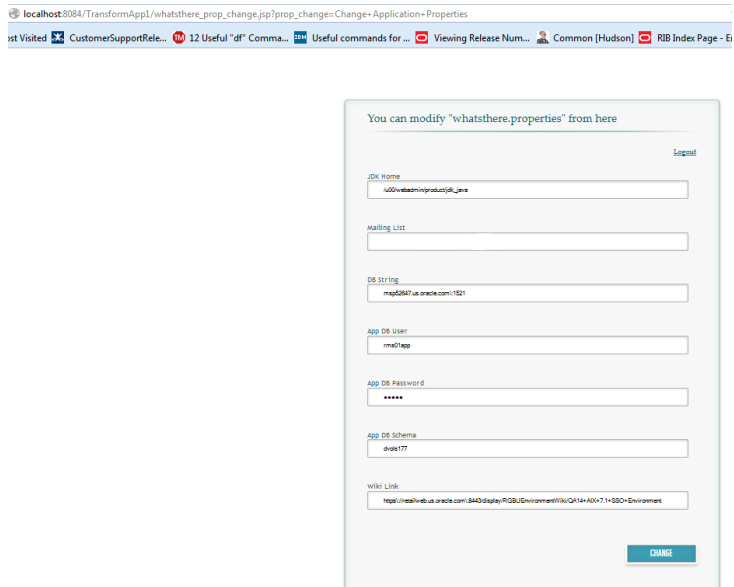


Figure 5.30: Properties Page Contents

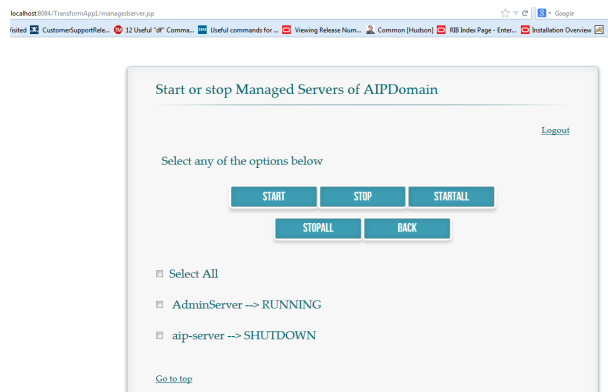


Figure 5.31: Start/Stop the select managed server

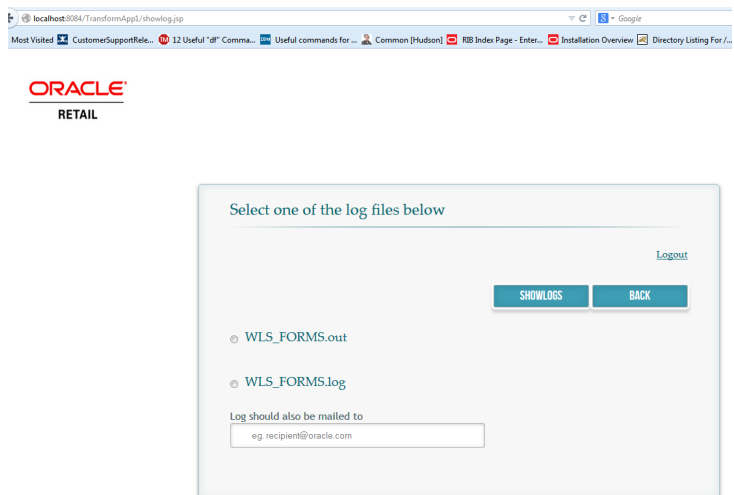


Figure 5.32: Managed Server Logs Page



# Chapter 6

## Conclusion and Future Work

This project accomplishes requirement that the work done till now complies with the goal of the project. The proposed automation has been implemented and found beneficiary in comparison with existing methodology. Product Deployment is a lengthy and time consuming process, but it is a very important process as well. Therefore, it is highly imperative that fully developed and built products first be deployed on internal servers and check if all the steps are correctly documented, to ensure that the client does not face any issues.

Automation can significantly reduce problems in computer-based systems, especially in lengthy, repetitive, boring, combinatorial and non-innovative tasks where human beings are likely to make more mistakes as compared to machines. Automation not only carries out long lists of mechanical steps efficiently but can also help verify the correctness of system components before they go into field

Future work comes with the Chef Automation for Development of environment and Machine learning technique for the Host monitoring tool. so that if there is any outage, software or hardware failure or we can say hyper visor failure at a time it is helpful for monitoring and predicting the status and health of the servers and is used for automatic recapitulated of servers in case of a breakdown. In events such as technical fault or accidental breakdown of servers, all the applications running on them come down. These servers and applications must be brought up one by one.

# Bibliography

- [1] Moinuddin Qadir, “Role of Automation in Computer-based Systems.”
- [2] Oracle, “Oracle Internal Files and Training Manuals,” *Oracle*, 2015.
- [3] Steve Vinoski, “Server-Sent Events with Yaws.”
- [4] Hostinger Tutorials, “<https://www.hostinger.com/tutorials/ssh-tutorial-how-does-ssh-work>.”
- [5] Slideshare.net, “<https://www.slideshare.net/nagainquisitive/hp-fortify-source-code-analyzersca>.”
- [6] iTMethods, “<http://electric-cloud.com/wiki/display/releasemanagement/Deployment+Automation>”
- [7] “<https://docs.oracle.com/cd/E12461-01/140/rsb-installation-guide/rsb-140-ig.pdf>.”
- [8] Tutorials Point, “<https://www.tutorialspoint.com/svn/>.”
- [9] Kiran Nagaraja, Fabio Oliveira and Ricardo Bianchini, Richard P. Martin and Thu D.Nguyen, “Understanding and Dealing with Operator Mistakes in Internet Services.”