

# Oracle Apps Task Automation – OATA

P. Prem Kumar and P. Srinivas Rao

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

February 26, 2020

# **ORACLE APPS TASK AUTOMATION – OATA**

<sup>1</sup>P Prem Kumar ASSISTANT PROFESSOR Prem.cse@jbiet.edu.in <sup>2</sup> Dr.P Srinivas Rao PROFESSOR hodcse@jbiet.edu.in

# 1:JB INISTITUTE OF ENGEENERNG & TECHNOLOGY , HYDERABAD, TELANGANA. 2:JB INISTITUTE OF ENGEENERNG & TECHNOLOGY , HYDERABAD, TELANGANA.

The OATA is an automation tool that gives a one stop point for all stakeholders (Managers, Team Leads and Team associates of OATA). The main objective of the OATA is to simplify the process as much as possible with minimum lines of code and to show the overall progress of tasks to managers and TLs. It helps to reduce the execution time and performs regular tasks like Patching, Script Execution, OTL tasks and Clone Execution. It also carries out the tasks like recycling Oracle ERP instance or individual components and starting services on multiple instances during unintended outages. OATA gives a quick view of all tasks. Role based access is provided to perform a specific task such as adding a new user, registering or editing instances in OATA Administration. The most important module in OATA is METADATA; it consists of sub modules such as Client, ERP, OBIEE, Hyperion, FMW and Generic Scripts. My current work is to summarize the METADATA state-of-the-art, learn how to resolve technical issue of OATA and to develop a new module called HEALTHCHECK REPORT in OATA TOOL.

Keywords: Oracle ERP, App DBA, Data Base Clone

#### **1. Introduction**

The Enterprise Resource Planning (ERP) is business management software, usually a suite of integrated applications that a company can use to collect, store, manage and interpret data from many business activities. The ERP system is considered a vital organizational tool because it integrates varied organizational systems and facilitates error-free transactions and production. However, ERP system development is different from traditional systems development [1]. ERP systems run on a variety of computer hardware and network configurations, typically using a database as an information repository [2].

The top 3 ERP vendors are Oracle, SAP and Microsoft. Out of all the ERPs, Oracle E-Business Suite (also known as Oracle Applications/Apps) is the most comprehensive suite of integrated, global business applications that enable organizations to make better decisions, reduce costs, and increase performance. This entire paper aims at Oracle E-Business Suite [3].

There are different types of users to use this ERP. They are, Project Manager, Functional Consultants, Techno-Functional Consultants, Technical Consultants, App DBA, User and Super User. Each and every user will have their own role to play. The project Manager, Functional Consultants, Techno-Functional Consultants, Technical Consultants roles is to deal with foreground work. The App DBA looks after the database. The user is the end user of the ERP and the Super user will look after the administration part. In the entire paper, App DBA is considered as an end user.

#### **1.1 Problem statement**

An App DBA's work involves a lot of time on the command prompt of UNIX environment. Primary duties involve performing clones of production environment, Patching, Service Recycle and Script Execution. During these activities the instructions needs to be adhered with utmost sincerity. A lot of manual errors creep in from time-to-time as majority of operations need human intervention. The errors cost many hours of labour and a significant loss of quality in deliverables. To reduce the mistakes made during redundant or routine activities the most affordable and accurate way is to automate them. A basic level automation has already been provided through shell scripting, which had the ease of being deployed over the respective command prompts in times of need. As this is again a character user interface, it hasn't decreased the error rate and human intervention.

# 1.1 Aim

To figure out all the problems of App DBA, the best solution is to "automate" all the tasks performed by App DBA using an easy to use information system that will assist App DBAs. The main purpose of automation is, it saves energy, labour and improves accuracy and quality. As we know that we cannot expect human to do work without errors, we need to make a computer to work more efficient than human.

## **1.2 Objectives**

The main objectives of this paper are mentioned below,

- > A sophisticated user interface.
- > Register all the Clients and instances with latest data.
- Automate all the important tasks like cloning, patching, script execution and service recycle with a single click.
- Set the hierarchy of the organization (Managers on top and employees at bottom).
- Limit user features based on their roles.
- > Send automatic mails to sender and their respective teams about the status of the task.
- > Generate reports based on different criteria's.
- > Export data to excel and can send it as a mail to their own user id.
- > Maintain a detailed log of all the tasks.

# **1.2 Applications**

Some of the interesting applications are mentioned below,

- ✓ The architecture has a flexibility to inherit its architecture to automate other products like Hyperion CPM (Corporate Performance Management) System and PeopleSoft HRM (Human Resource Management) System.
- $\checkmark$  It makes huge profits with less human power.
- $\checkmark$  It has a capability to generate a graphical report which makes everyone to understand clearly.
- ✓ Reduces the human errors and increases reliability and robustness.

## 1.3 Purpose

The Companies those are a suite of internal tools that assist organization and employees to communicate and collaborate more effectively. The determination of this document are to give a clear sketch of the technical design of the Oracle COE Tool Automation and provide an overview for the OATATool implementation.

The main purpose of OATA tool is to –

- Reduce the total time of the manual tasks done by App DBA team by automating the tasks and execute them through OATAtool
- Generate a detailed report of tasks showing success rate, no. of tasks done through OATA and no. of tasks done by individual App DBA.
- Provide accurate data of the tasks

This document is not projected on installation and configuration details of the OATAtool.

# 1.4 Scope

The scope of this document will be focused on the use case view, logical view, and implementation view of the architectural design of the OATA tool. We will identify important aspects of each view as well as the architectural significance each presents to the success of the OATA tool.

# 2. Design Overview

2.1 Approach This document is prepared and extended in multiple phases over the course of the project -

- Requirements Phase- During the Requirements Phase, the initial version of this document is created, describing the candidate architecture to be validated in the System Design Phase.
- System Design Phase- During the System Design phase, the Evolutionary Prototype is created and this document is finalized by establishing a sound architectural foundation for the Construction Phase.
- Construction Phase- During the Construction Phase, this document is not expected to change radically; it is mainly updated to reflect changes in any interface definitions.

Transition / Training Phase- During the Transition/Training Phase, no further additions or modifications are made to this document.

## 2.2 Architectural Goals and Constraints

The Architectural Goals and Constraints of OATAtool are:

- A sophisticated user interface.
- Register all the Clients and instances with latest data.
- Automate all the important tasks like cloning, patching, script execution and service recycle with a single click.
- Set the hierarchy of the organization (Managers on top and employees at bottom).
- Limit user features based on their roles.
- Send automatic mails to sender and their respective teams about the status of the task.
- Generate reports based on different criteria's.
- Export data to excel and can send it as a mail to their own user id.
- Maintain a detailed log of all the tasks.

## 2.3 Guiding Principles

Guiding principles provide a foundation upon which to develop the target architecture for the OATA tool, in part by setting the standards and measures that the tool must satisfy. These in turn drive design principles that can be used to validate the design and ensure that it is aligned with OATA's overall Architecture, Design Principles and Standards. Some of the guiding principles that will be followed during the OATA tool design and development are outlined below.

## 2.3.1 Scalable

Scalability is the ability of the platform to scale both up and down to support varying numbers of users or transaction volumes. The application should be able to scale horizontally (by adding more servers) or vertically (by increasing hardware capacity or software efficiency).

### 2.3.2 Flexible

Flexibility is the ability of the application to adapt and evolve to accommodate new requirements without affecting the existing operations. This relies on a modular architecture, which isolates the complexity of integration, presentation, and business logic from each other in order to allow for the easy integration of new technologies and processes within the application.

### 2.3.3 Standards-Based

Portal services will comply with established industry standards. The standards-compliance will not only apply to application development but also to design, platform/infrastructure and other parts of the OATAapplication. Examples of standards include HTML, XML, J2EE, and JSP.

### 2.4 Design Patterns

Design patterns are elements of reusable object oriented software. A design pattern catalog is a repository of design patterns. Use of such patterns makes the design of an application transparent. These patterns have been used successfully by developers in their respective fields, and therefore, the pros and cons of the pattern (as well as implementation issues) are known beforehand. All design patterns are reusable and can be adapted to particular contexts.

The design patterns used in the design and development of the OATA tool is-MVC Architecture

- > Apache Tomcat is used as the servlet container.
- Spring MVC [4] framework is selected to be the web Model-View-Controller (MVC) framework. It provides tight integration with the underlying business services layer. It is also very flexible to integrate different web view technologies.

The "OpenSessionInViewFilter"[12] pattern in the Spring framework is used by the web controllers to open a Hibernate session for each incoming HTTP request.

## **2.5 Design Principles**

Design principles are mentioned below,

- The architecture has a flexibility to inherit its architecture to automate other products like Hyperion CPM (Corporate Performance Management) System and PeopleSoft HRM (Human Resource Management) System.
- It makes huge profits with less human power.
- It has a capability to generate a graphical report which makes everyone to understand clearly.
- Reduces the human errors and increases reliability and robustness.

## 3. Topology Diagram

The below diagram illustrates the OATA architecture and communication between different components.



### Fig 2: Topology Diagram

The main components of the OATAare,

Apache tomcat web server, External server ,MySQL database, APPJUMP, Client Servers

### 3.1 Apache tomcat web server

The OATAweb application is hosted using Apache Tomcat Webserver. Tomcat executes Servlets and JSPs specifications from Oracle, and provides a "pure Java" HTTPweb server environment for Java code to run in. This server holds all the required APIs to run the web application efficiently and securely [5][6].

### 3.2 External server

The external server plays a key role to access the web application securely from any remote machine. This smoothly balances the load when there is huge number of active sessions [7][8].

### 3.3 MySQLdatabase

The entire data is store in MySQL database. This is reserved as a backend unit for OATA. This database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. There is a special exclusive encryption technique applied on sensitive tables to protect them from security threat [9]. A crontab is dedicated to take a backup of the database on daily basis.

# **3.4 APPJUMP**

The APPJUMP is an internal Authentication tool. This adds an additional security layer to the OATA. Using this APPJUMP, only authorized users can connect to external server. After successful connection with external server, the user can access the OATA. All the company employees have their own dedicated username and password to access this APPJUMP [10].

### **3.5 Client Servers**

There are almost 100 clients using Oracle Apps. All the clients have multiple servers. All the servers are located in different Data Centers. The location of the client server and other external details are stored in local repository (MySQL). By using local repository, the client servers are accessed. The OATA connects to the client server using SSH. After connecting to the server, it performs the required task and then returns the control back [11].

## 4. Application Architecture

Application architecture defines the various components and their interactions in context of a whole system. Application architecture is the critical software that bridges the architectural gap between the application server and the application's business logic, thereby eliminating the complexities and excessive costs of constructing, deploying and managing distributed enterprise applications.

The OATATool will have a layered application architecture which provides some of the key features below -

- STRUCTURE: Organizing applications along business-level boundaries and not technical boundaries.
- SPEED & FLEXIBILITY: Making application changes through configuration and not programming.
- CONTROL: Modifying, extending or overwriting any architectural element.

• REUSE: Achieving greater reusability and integration by loosely coupling application logic to infrastructure. At a *conceptual* level, they represent distinct and cohesive aggregations of functionality. The OATATool design is based on a tiered approach. "A tier is a logical partition of the separation of concerns of the system. Each tier is assigned its unique responsibility in the system. We view each tier as logically separated from one another. Each tier is loosely coupled with the adjacent tier." The OATATool architecture can be represented in the following layers illustrated by the diagrambelow:



Fig 3: Application Architecture Overview

### 5. OATA Tool Literature

The conventional system is a Character User Interface (CUI) where the App DBA needs to remember all the set of instructions and execute them sequentially which makes their role very tedious and prone to errors. One of the paramount tasks of App DBA is cloning. This is the most crucial task usually leads to errors. And also it is quite hard to track employee's performance and to generate reports using this conventional system. Though shell scripting gives a better result, it is not the best solution. Moreover it is again a character user interface (CUI). The best possible solution to overcome these problems is to automate all the App DBA tasks using a sophisticated Graphical Interface.

The App DBA will have a right to login to specific server and have access to do a particular task from remote location using secure shell (SSH). Some of the tasks taken care by App DBAs are describe below,

- Patching.
- Database Clone.
- Service recycles / bounces.
- Script execution.

### 6. Application Implementation

The Application Deployment Structure is shown below. Screening application will be deployed as EAR (Enterprise Archive) file using ANT build scripts.



Diagrammatic layout of OATA deployment structure is shown below -

Fig 4: OATA Directory Structure for Deployment

#### 7. Observations and Result Analysis

This chapter deals with the observations and result analysis of this tool. According to the observations, this tool is being used as a powerful weapon by all the App DBA's. Day by day the usage of this tool continues to proliferate. And the statistics clearly shows that this tool is playing a key role in the industry.

#### 7.1 Observations

This tool has a very sophisticated user interface, which made everyone to use this tool frequently. Day by day there is a lot of exponential growth in the profit ratio because every task is being performed using this application. The real taste of the tool looks too good after analysing the report. The below Table 1 has stats, it makes everyone to clearly understand the importance and the usage of this application in the organization.

Total	Time Manual Exec.	Total time(Manual)	Time in OCTA	Total time(Auto)
2757	10	27570	3	8271
1192	25	29800	15	17880
923	15	13845	5	4615
849	30	25470	10	8490
752	20	15040	9	6768
431	20	8620	10	4310
217	10	2170	5	1085
203	960	194880	480	97440
203	25	5075	15	3045
127	15	1905	5	635
119	10	1190	2	238
102	б	612	3	306
102	40	4080	20	2040
38	5	190	1	38
28	20	560	10	280
28	5	140	1	28
23	15	345	9	207
22	12	264	6	132
8116		331756		155808
		5529.27		2596.8
	Total 2757 1192 923 849 752 431 217 203 203 127 119 102 102 38 28 28 28 28 22 8116	Total Time Manual Exec.   2757 10   1192 25   923 15   849 30   752 20   431 20   217 10   203 960   203 255   127 15   119 0   102 6   102 40   38 5   22 12   23 15   22 12   8116	Total Time Manual Exec. Total time(Manual)   2757 10 27570   1192 25 29800   923 15 13845   849 30 25470   752 20 1500   203 960 19480   203 960 194880   203 25 5075   1127 15 1190   102 6 612   102 40 4080   38 5 1990   22 12 264   8 5 1490   22 12 264   8116 331756	$\begin{tabular}{ c c c c c } \hline Total time(Manual) & Time in OCTA \\ \hline Time Manual Exec. & Total time(Manual) & Time in OCTA \\ \hline 1192 & 25 & 29800 & 15 \\ \hline 923 & 15 & 13845 & 5 \\ \hline 849 & 30 & 25470 & 10 \\ \hline 752 & 20 & 15040 & 9 \\ \hline 431 & 20 & 8620 & 10 \\ 217 & 10 & 2170 & 5 \\ 203 & 960 & 19480 & 480 \\ \hline 203 & 25 & 5075 & 15 \\ \hline 127 & 15 & 1095 & 55 \\ \hline 119 & 10 & 1190 & 22 \\ 102 & 6 & 612 & 3 \\ 102 & 40 & 4080 & 200 \\ \hline 38 & 5 & 190 & 10 \\ \hline 28 & 20 & 5 & 140 & 10 \\ \hline 28 & 20 & 5 & 140 & 10 \\ \hline 22 & 115 & 13945 & 9 \\ \hline 22 & 12 & 264 & 6 \\ \hline 8116 & 331756 & \\ \hline \end{tabular}$

Table 1: Time Savings Stats (Jan2014 onwards).

To be precise, the following points are the More than 50% of the time is saved.

- More than 70% of clones are done through OATA(till date).
- 86% success rate is attained.
- Time consumption is decreased exponentially.
- Clone execution time through OATAgot reduced by 50% compared to manual clone.
- Human error rate is very minimal.

#### 8.2 Results

The results of the OATA was analysed with several test cases on several clients. Each and every module in the OATA was perfectly analysed and the final results were amazing. Some of the important modules which were tested with the best test cases are mentioned below, important attributes to this paper. Are App Jump login page, External server desktop OATAlogin page ,Dashboards ,Service bounce , Patch analysis, AdPatch and OPatch, Script execution ,Clones,Reports

## **8.3 Assumptions and Constraints**

- To access this tool, users need to connect via VPN
- The web browser will be the primary client used by employees and public users.
- Technology interoperability won't hinder the project's progress
- The project team will deliver work as scheduled
- The project team will have access to required servers
- Changes to the project scope will follow a predefined change control system.
- Need to follow legal or industry-specific regulations that require extra communication and analysis among team members.

## 9. Conclusion and Future Work

An Oracle apps DBA's spends lot of time on the command prompt of UNIX environment. Majority of operations need human intervention, this leads to creep lot of errors. To reduce the mistakes made during redundant or routine activities the most affordable and accurate way is to automate them. To provide a unified interface and to provide a one-stop approach to all scripts a 3-tier structure need to been developed as a tool. This tool purpose is to contain all errors made through human intervention and non-adherence to instructions.

The OATA- Oracle Apps Task Automation is an efficient tool for Oracle App DBAs to deliver their tasks in less time. Loads of time is being saved by using this automation tool. It is producing huge profits for the company and App DBAs as well. Rich interface with secured validation is provided to the users (App DBAs) using various state of the art web technologies to attract users and to eliminate the human errors.

Using this tool clone execution time got reduced by 50% compared to manual clone. Other tasks like patching, script execution and service recycle execution time got reduced by more than 60% compared to manual process. Human error rate is very minimal. And finally, the success rate of all the tasks is 86%.

Without a doubt, this project makes Oracle App DBAs work simple. As this tool is still in its first version, it has a wide scope for the future enhancements such as

- Providing chatting and discussion forums.
- Optimizing the existing queries to get the results/reports in fraction of seconds
- Extended to a mobile application.
- SSL protocol for secure login.

### **10. References**

- [1] SHAUL, L. AND TAUBER, D. 2012. CSFs along ERP life-cycle in SMEs: a field study. Industrial Management & Data Systems, 112(3), 360-384.
- [2] Khosrow–Puor, Mehdi. (2006). Emerging Trends and Challenges in Information Technology Management. Idea Group, Inc. p. 865.
- [3] Oracle E-Business Suit Architecture<u>http://www.oracleerpappsguide.com/2011-/04/oracle-applications-r12-architecture.html</u>
- [4] Diana M. Selfa, Maya Carrillo, Ma. DelRocío Boone, "A Database and Web Application Based on MVC Architecture". Electronics, Communications and Computers, CONIELECOMP 2006, 16th International Conference, pp. 48 (2006).
- [5] CC.kun, G., Kakilli, A., (2011), Analysis of Tendencies in WEB Based Technology Education of Undergraduate Students inTurkey, Technics Technologies Education Management-TTEM, ISSN 1840-1503, Vol.6 No. 2, pp. 1008-1015.
- [6] S. Kounev, B. Weis, and A. Buchmann, "Performance Tuning and Optimization of J2EE Applications on the JBossPlatform," Journal of Computer Resource Management, vol. 113, 2004.
- [7] D. Menasće, V. Almeida, R. Fonseca, and M. Mendes, "A Methodologyfor Workload Characterization of E-commerce Sites," inProceedingsof the 1st ACM conference on Electronic commerce, Denver, Colorado, United States, Nov. 1999, pp. 119–128.

- [8] R. Bhatti, E. Bertino, A. Ghafoor, and J. Joshi, "XML-BasedSpecification for Web Services Document Security," Computer, vol. 37, no. 4, pp. 41-49, 2004.
- [9] E. Ferrari and B.M. Thuraisingham, "Security and Privacy for Web Databases and Services,"Advances in Database Technology—EDBT2004, Proc. Ninth Int'l Conf. Extending Database Technology, Mar.2004.
- [10] M. Bellare and P. Rogaway.Optimal Asymmetric Encryption–How to Encryptwith RSA. In A. De Santis, editor, Advances in Cryptology – Eurocrypt '94, volume 950 of Lecture Notes in Computer Science, pp. 92 – 111. Springer Verlag, 1995.
- [11] Shan, T.C., Hua, W.W., (2006) Taxonomyof Java Web Application Frameworks, e- Business Engineering, ICEBE '06. IEEE International Conference on, pp.378 – 385.
- [12] Stoica, I., Morris, R., Liben-Nowell, D., Karger, D.R., Kaashoek, M.F., Dabek, F., Balakrishnan, H.: Chord: a scalable peerto-peer lookup protocol for internet applications. IEEE/ACM Trans. Netw.11 (1), 17–32 (2003).
- [13] M. Elkoutbi, I. Khriss, and R. K. Keller. Generating userinterface prototypes from scenarios. InProcs of RE '99, page150. IEEE CS, 1999.