A Report

On

Multibrowser Support for Oracle Financial Services Analytical Applications

By

Vibhor Joshi 2013A8PS511G

At

Oracle Financial Services Software Ltd.
Bangalore

A Report

On

Multibrowser Support for Oracle Financial Services Analytical Applications

By

Vibhor Joshi 2013A8PS511G

Prepared for

Partial fulfilment of the course

Practice School-II BITS-C412

At

Oracle Financial Services Software Ltd. Bangalore
A Practice School –II station of
BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE

Practice School Division

Station: Oracle Financial Services Software	**Centre:** Bangalore
Duration: 5 Months	**Date of Start:** July 13th, 2016
Name: Vibhor Joshi	**ID:** 2013A8PS511G

Title of Report: Multibrowser Support for Oracle Financial Services Analytical Applications

Name and Designation of Experts:	Renjith Ravindram Pankaj Changgwani
Names of PS Faculty:	Raja Vadhana.

Key Words: OFSAA, Web Development, OJET, PMF.

Abstract: OFSAA is a suite of applications (previously named OFSA) for the Financial Services. The application can easily incorporate changing trends and requirements in risk management with minimal effort in terms of cost and time. The project entails three major parts Enabling support for the OFSAA software’s Enterprise Modelling Feature in all popular browsers like Google Chrome, Mozilla Firefox and Internet Explorer, Updating the current Front End to a new Style using OJET and Integrating New features like PMF with the modelling framework.

The project entails three major parts Enabling support for the OFSAA software’s Enterprise Modelling Feature in all popular browsers like Google Chrome, Mozilla Firefox and Internet Explorer, Updating the current Front End to a new Style using OJET and Integrating New features like PMF with the modelling framework.

Signature of the Student	Signature of the PS Faculty

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE
Practice School Division
Response Option Sheet

Station: Oracle Financial Services Software **Center:** Bangalore

Name: Vibhor Joshi **ID:** 2013A8PS511G

Title of Report: Multibrowser Support for Oracle Financial Services Analytical Applications

Usefulness of the project to the on-campus courses of study in various disciplines: Project should be scrutinized keeping in view the following response options. Write Course No. and Course Name against the option under which the project comes.
Refer Bulletin for Course No. and Course Name.

Code No.	Response Option	Course No.(s) & Name
1	A new course can be designed out of this project.	No
2	The project can help modification of the course content of some of the existing Courses	No
3	The project can be used directly in some of the existing Compulsory Discipline Courses (CDC)/ Discipline Courses Other than Compulsory (DCOC)/ Emerging Area (EA), etc. Courses	No
4	The project can be used in preparatory courses like Analysis and Application Oriented Courses (AAOC)/ Engineering Science (ES)/ Technical Art (TA) and Core Courses.	No
5	This project cannot come under any of the above mentioned options as it relates to the professional work of the host organization.	Yes

Signature of the Student Signature of the PS Faculty

Acknowledgement

I would like to thank the **PS Division** for providing a course like PS-II which gives us professional exposure. I would like to thank **Oracle Financial Services Softwares** for providing an opportunity to intern here. I would like to thank Sanjay Bajaj, Renjith Ravindram and Pankaj Changgwani for their guidance and mentorship during the course of the project**.**

I would like to thank our instructors Ms Raja Vadhana for encouraging and guiding us during this course.

Table of Contents

Introduction

OFSAA is a suite of applications (previously named OFSA) for the Financial Services industry.

The application can easily incorporate changing trends and requirements in risk management with minimal effort in terms of cost and time. Using OFSAA, Banking and Financial Services institutions can measure and meet regulatory requirements and risk-adjusted performance objectives, cultivate a risk management culture through transparency, reduce costs spend to meet compliance and regulatory requirements, and get improved customer behaviour. The software development is done by several teams working on different features of the application. The data modelling feature is handled by the modelling team.

A major part of developing the platform for the OFSAA Enterprise Modelling Framework is developing the screens associated with the features provided by the framework, such as Model Creation, Model Execution and Model Deployment. In addition there is a need to develop the API with which the screens communicate with the rest of the framework.

Technologies Used

The platform to be worked on uses HTML, CSS, Java Server Pages and Javascript.

HyperText Markup Language (HTML):

HyperText Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS), and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a webserver or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects, such as interactive forms may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> introduce content into the page directly. Others such as <p>...</p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript which affect the behaviour and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

Cascading Style Sheets (CSS):

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. It can also display the web page differently depending on the screen size or viewing device. Readers can also specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author specified.

Changes to the graphic design of a document (or hundreds of documents) can be applied quickly and easily, by editing a few lines in the CSS file they use, rather than by changing markup in the documents.

The CSS specification describes a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities (or weights) are calculated and assigned to rules, so that the results are predictable.

Javascript:

Javascript is a high-level, dynamic, untyped, and interpreted programming language. It has been standardized in the ECMAScript language specification. Alongside HTML and CSS, it is one of the three core technologies of World Wide Web content production; the majority of websites employ it and it is supported by all modern Web browsers without plug-ins.

JavaScript is prototype-based with first-class functions, making it a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles. It has an API for working with text, arrays, dates and regular expressions, but does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

JavaServer Pages (JSP)

JavaServer Pages (JSP) is a technology that helps software developers create dynamically generated web pages based on HTML, XML, or other document types. To deploy and run JavaServer Pages, a compatible web server with a servlet container, such as Apache Tomcat or Jetty, is required. JSPs are translated into servlets at runtime; each JSP servlet is cached and re-used until the original JSP is modified. JSP can be used independently or as the view component of a server-side model–view–controller design, normally with JavaBeans as the model and Java servlets (or a framework such as Apache Struts) as the controller. JSP allows Java code and certain pre-defined actions to be interleaved with static web markup content, such as HTML, with the resulting page being compiled and executed on the server to deliver a document. The compiled pages, as well as any dependent Java libraries, contain Java bytecode rather than machine code. Like any other Java program, they must be executed within a Java virtual machine (JVM) that interacts with the server's host operating system to provide an abstract, platform-neutral environment.

JSPs are usually used to deliver HTML and XML documents, but through the use of Output Stream, they can deliver other types of data as well.

Extensible Markup Language (XML):

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The W3C's XML 1.0 Specification and several other related specifications,—all of them free open standards—define XML.

The design goals of XML emphasize simplicity, generality, and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures such as those used in web services.

Several schema systems exist to aid in the definition of XML-based languages, while programmers have developed many application programming interfaces (APIs) to aid the processing of XML data.

JSON:

JavaScript Object Notation) is an open-standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs. It is the most common data format used for asynchronous browser/server communication, largely replacing XML which is used by AJAX.

JSON is a language-independent data format. It derives from JavaScript, but as of 2016, code to generate and parse JSON-format data is available in many programming languages. The official Internet media type for JSON is application/json. The JSON filename extension is .json.

Douglas Crockford originally specified the JSON format; two competing standards, RFC 7159 and ECMA-404, define it. The ECMA standard describes only the allowed syntax, whereas the RFC also provides some semantic and security considerations.

Problem Statement

There are two major aspects to the problem statement.

1. Updating existing code to HTML5 to allow support in all browsers.

2. Integrating new features like PMF into the Modelling framework.

OFSAA HTML5 and MultiBrowser Enabling

IE* has 2 rendering modes

a) Quirks mode (the older rendering mode)
b) Standards Mode (the newer standards which imposes stricter rules)

IE or any Browser cannot render both of the above modes. Only one mode can be adopted for a page.

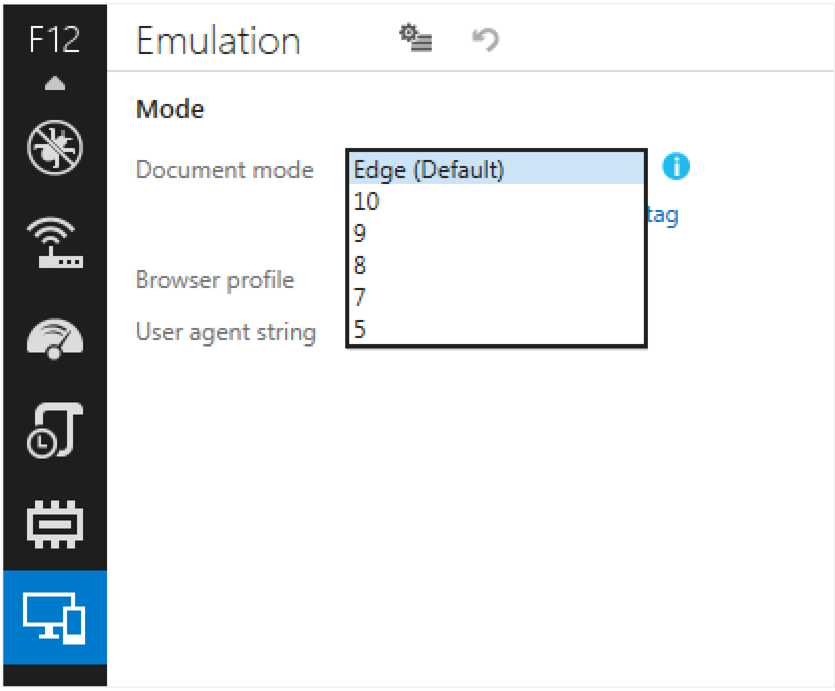
DOCTYPE Declaration

This directs the browser to render the page in standards mode.

<!DOCTYPE html> goes at the top of every **HTML5** page.

The below line will force the browser (IE) to render in Standards Mode even if compatibility is turned on
<meta http-equiv="X-UA-Compatible" content="IE=edge">

Use IE Developer tools (F12) to check the Document mode in IE Browser



Emulation

Mode

Document mode: Edge (Default)

Browser Support

The following four browsers are supported:

- Google Chrome: Support for the most current major stable channel release only

- Mozilla Firefox: Support for the most current major ESR version and above, in production only

- Apple Safari: Support for most current major production release and one prior release

- Microsoft IE Browser: Support for the most current major production release on a supported operating system

	Chrome	**Firefox**	**Microsoft Browser**	**Safari**
Android	Supported *	Not Supported	N/A	N/A
iOS	Not Supported	Not Supported	N/A	Supported
Mac OS X	Supported	Supported	N/A	Supported
Windows	Supported	Supported	Supported	Not Supported

document.getElementById

"document.getElementById" was introduced in IE7+ and for backward compatibility Browser used to identify by the "Name" attribute in the absence of "ID" attribute. This was allowed only for Quirks mode. This workaround existed only as an interim arrangement to allow applications time to switch to standards.
With Standards Mode the work around (Name Attribute used in the absence of ID Attribute) has been de-supported.

Example:

Quirks Mode :

```
<input type="hidden" name="endRow" value="">
document.getElementById(“endRow”);
```

Standards Mode:

```
<input type="hidden" name="endRow" id="endRow"  value="">
document.getElementById(“endRow”);
```

Case Sensitive ID

Example:

Quirks Mode:

```
<input type="hidden" name="endRow"  id="endRow"  value="">
document.getElementById("endrow")
```

Standards Mode:

```
<input type="hidden" name="endRow" id="endRow"  value="">
document.getElementById("endRow")
```

Alt and Title

It has been observed that the “alt” attribute no longer displays the image tooltip in browser with IE Standards mode.

The “alt” attribute is required attribute and read by screen readers.

Example:

Quirks Mode:

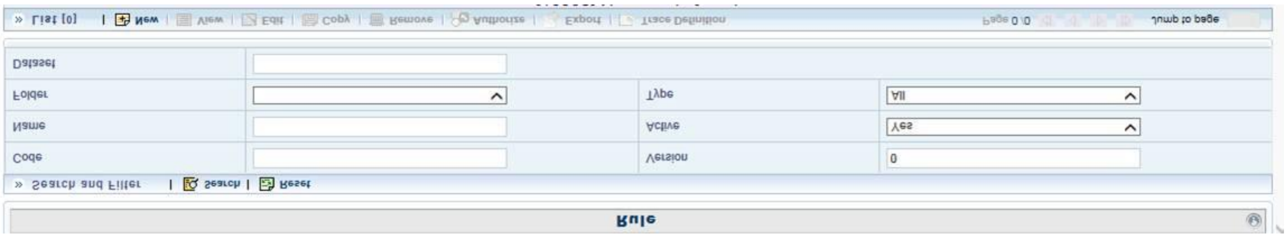
```
<img alt=”tooltip string” />
```

Standards Mode:

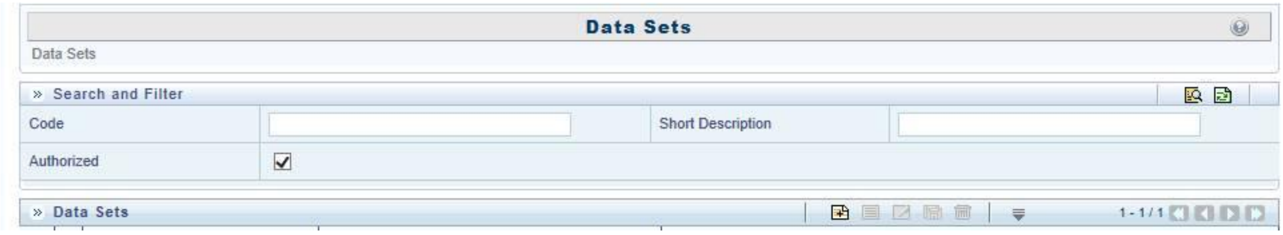
```
<img alt=”tooltip string” title=” tooltip string”  />
```

“title” attribute added to display the tooltip on screen.

If an icon has got a text besides it then the tooltip is not required. As seen in the example below



If the Icon does not have a text besides it, then, in addition to the “alt” attribute, we will also need a “title” attribute to display the tooltip for the image.



Data Sets

Data Sets

» Search and Filter

Code		Short Description	
Authorized	☑		

» Data Sets

1 - 1 / 1

Definition of units

In Quirks mode the default is .px.

In Standards mode, Left, top, width and height parameters must have suffix as the unit type. The unit can be either, px, em, rem, %, etc has to be suffixed after the numeric value.

Otherwise no width / height / left / top will be applied.

Example:

Quirks Mode:

<img src=”images/add.gif” alt=”add” title=”add” width=”20” height=”20”/>

Standards Mode:

<img src=”images/add.gif” alt=”add” title=”add” width=”20px” height=”20px”/>

Trailing comma in a JSON object

JSON object ignores trailing comma.

Example:

Quirks Mode:

[0,1,2,3,4,5,]

Standards Mode:

[0,1,2,3,4,5]

[0,1,2,3,4,5,] - In Quirks mode the object count will be 7.

[0,1,2,3,4,5,] - In Standards mode the extra comma is ignored and the count will be 6.

The JSON specification does not allow a trailing comma.

UI Distortion with span tags.

In Standards Mode:

For an inline element such as a <span> , the CSS attributes such as Width, Height do not get applied.
The span element has to explicitly declare inline-block level for elements to apply these attributes.

Example:

Quirks Mode:

<span style="width:200px;height:20px"></span>

Standards Mode:

<span style="width:200px;height:20px;display:block"></span>

Javascript display = "block" HTML Attribute

In a few cases, you may want to retain "Div" or other elements to be a "Block Element".

In most tags like Table, TR, TD, Span.. etc. it is noticed that if the "display:block" attribute is applied, the elements gets converted to a Block Element.

To avoid this, instead of using .display = 'block' it should be made either as, .display = '' or .display = 'inline-block'.

Quirks Mode:

```
        Obj.style.display = "block"; //converts the element to a block element
```

Standards Mode:

```
        Obj.style.display = "inline-block";
                         Or
        Obj.style.display = "";
```

CSS expressions

The header freeze feature implemented by FormsFramework uses CSS expressions in Quirks Mode, which has been discontinued in IE11.

Quirks Mode

```
.rowLine{position:relative;left:expression(this.offsetParent.scrollLeft)}
```

Javascript expressions are not allowed in CSS anymore because of security reasons.

Standards Mode:

Remove the Javascript expressions from css file. Handle the position code in your .JS file.

Use CSS style Position:Fixed with Javascripts in your JS file.

Create Popup

CreatePopup was IE Browser specific. It is a window used to launch a popup-up that overlap all iframes. This will NOT work in IE Standards Mode or any other Browser.

window.showModalDialog

window.showModalDialog is not supported in Chrome from the recent update of Chrome 37

Quirks Mode

```
window.showModalDialog('url', 'Title', 'win properties');
window. showModelessDialog('url', 'Title', 'win properties');
```

Standards Mode:

```
Window.open('url', 'Title', 'win properties')
```

Radio button obj.getAttribute("checked")

It is observed that checkboxes having obj.getAttribute("checked") is not supported in standards mode.

Quirks Mode

```
obj.getAttribute("checked")=true;
```

Standards Mode:

```
obj.checked =true;
```

parent.frames

Using parent.frames without referring to the window object results in undefined error.

Quirks Mode

```
parent.frames["fr1"].document.getElementById("cField ");
```

Standards Mode:

```
window.parent.frames["fr1"].document.getElementById("cField");
```

HTML Tags

Make sure that all HTML tags are closed properly.

If the html has got any orphaned or extra <td><tr> or any <Tag> which is not closed, it can result in an error.

Integration of Process Modelling Framework

In order to monitor and automate the process of Model Deployment we integrated the Process Modelling Framework to our model deployment procedure.

Initially if a mathematical model created by a user was to be deployed, the user would have to request for deployment and manually ask a model authorizer to allow the deployment of the model. In addition there was no method to go check the progress of the request other than to check the model directly from the model screen.

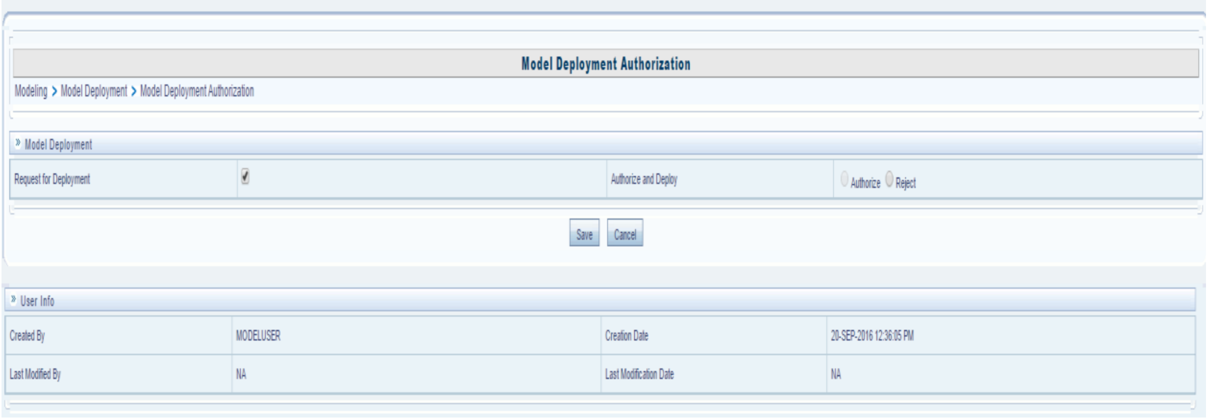
There are two types of users model creators and model administrators.

Model Creators can create and test models but cannot deploy them. In order to deploy the model they can request deployment, which can only be authorized by a model administrator.

Model Administrators can Approve or Reject the deployment of a model.

Model Deployment Process with PMF

1. The Modelers create the model and request to deploy.



Model Deployment Authorization

Modeling > Model Deployment > Model Deployment Authorization

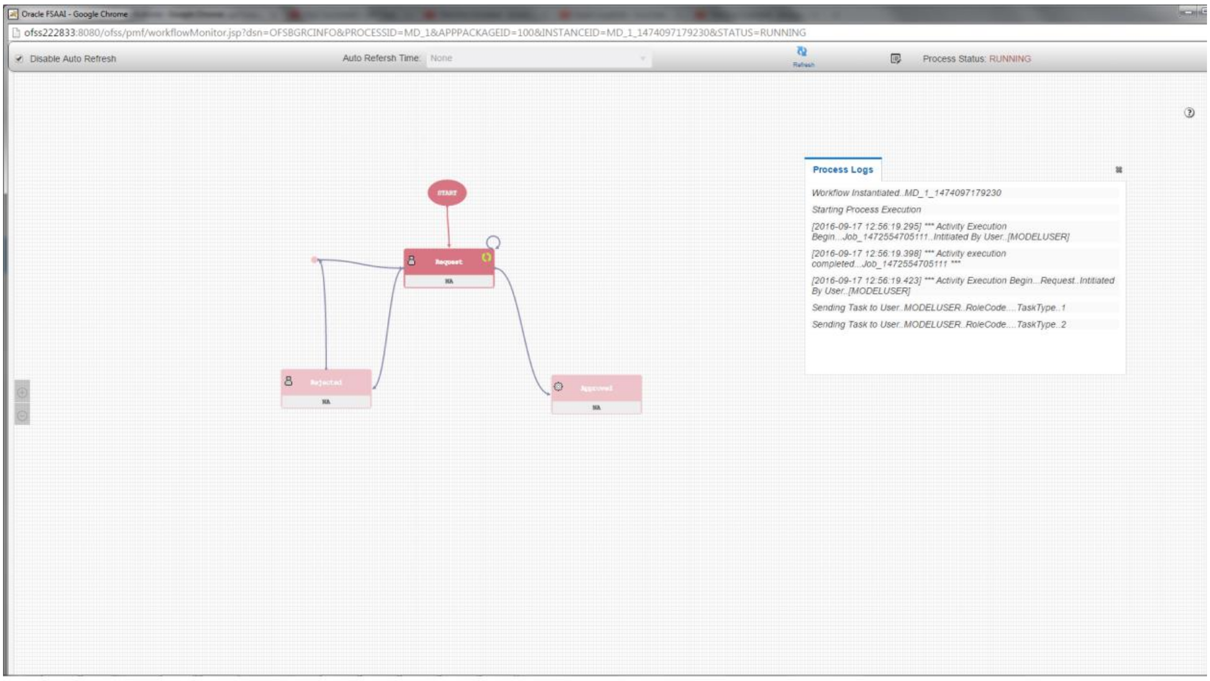
» Model Deployment

Request for Deployment	☑	Authorize and Deploy	○ Authorize ○ Reject

» User Info

Created By	MODELUSER	Creation Date	20-SEP-2016 12:36:05 PM
Last Modified By	NA	Last Modification Date	NA

2. The progress of the Model Deployment can be viewed with the process monitor.



Oracle FSAAI - Google Chrome

ofss222833:8080/ofss/pmf/workflowMonitor.jsp?dsn=OFSBGRCINFO&PROCESSID=MD_1&APPPACKAGEID=100&INSTANCEID=MD_1_1474097179230&STATUS=RUNNING

☑ Disable Auto Refresh

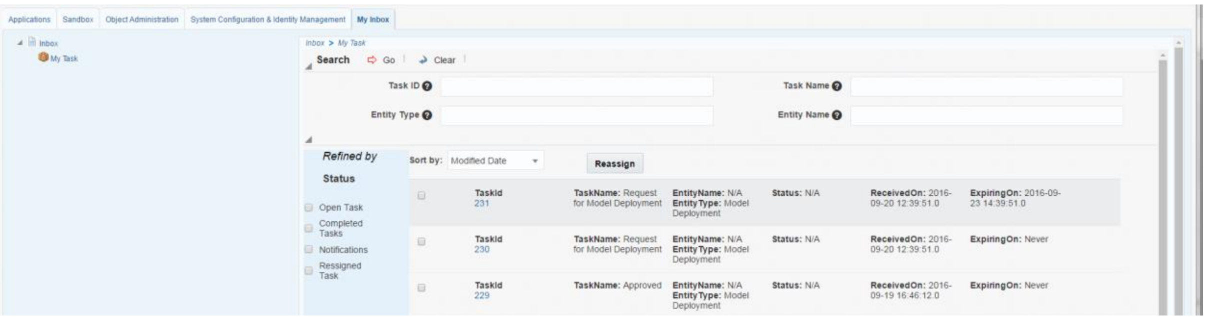
Auto Refersh Time: None

Process Status: RUNNING

Process Logs

```
Workflow Instantiated..MD_1_1474097179230
Starting Process Execution
[2016-09-17 12:56:19.295] *** Activity Execution Begin...Job_1472554705111..Intitiated By User..[MODELUSER]
[2016-09-17 12:56:19.398] *** Activity execution completed...Job_1472554705111 ***
[2016-09-17 12:56:19.423] *** Activity Execution Begin...Request..Intitiated By User..[MODELUSER]
Sending Task to User..MODELUSER..RoleCode....TaskType..1
Sending Task to User..MODELUSER..RoleCode....TaskType..2
```

3. A notification is sent to the Model Administrators, who can then access the model directly from the Notification they received.



Task ID
Task Name
Entity Type
Entity Name

Refined by

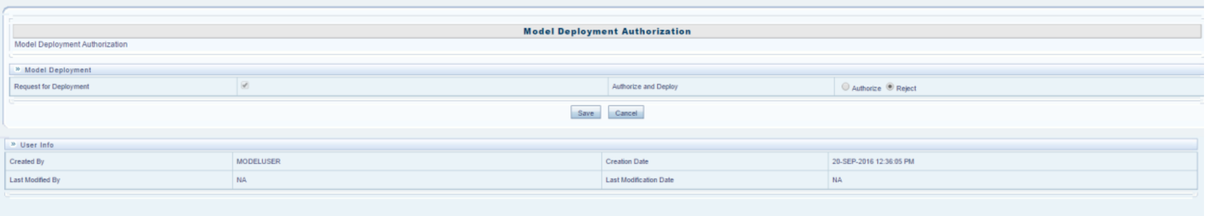
Status

□ Open Task
□ Completed Tasks
□ Notifications
□ Ressigned Task

Sort by: Modified Date

□	**TaskId** 231	**TaskName:** Request for Model Deployment	**EntityName:** N/A **EntityType:** Model Deployment	**Status:** N/A	**ReceivedOn:** 2016-09-20 12:39:51.0	**ExpiringOn:** 2016-09-23 14:39:51.0
□	**TaskId** 230	**TaskName:** Request for Model Deployment	**EntityName:** N/A **EntityType:** Model Deployment	**Status:** N/A	**ReceivedOn:** 2016-09-20 12:39:51.0	**ExpiringOn:** Never
□	**TaskId** 229	**TaskName:** Approved	**EntityName:** N/A **EntityType:** Model Deployment	**Status:** N/A	**ReceivedOn:** 2016-09-19 16:46:12.0	**ExpiringOn:** Never

4. The Model Administrator can then authorize or reject the model.



Model Deployment Authorization

Model Deployment Authorization

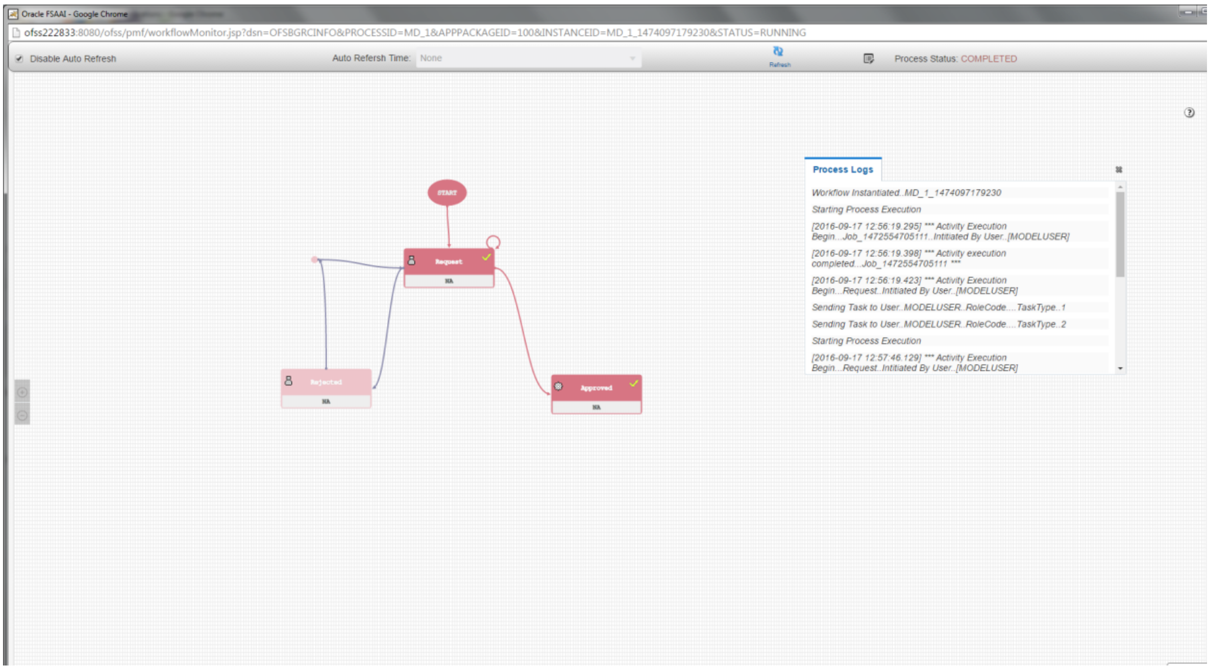
» Model Deployment

Request for Deployment	☑	Authorize and Deploy	○ Authorize ● Reject

» User Info

Created By	MODELUSER	Creation Date	20-SEP-2016 12:36:05 PM
Last Modified By	NA	Last Modification Date	NA

5. In the Process Monitor you can view the status of the model.



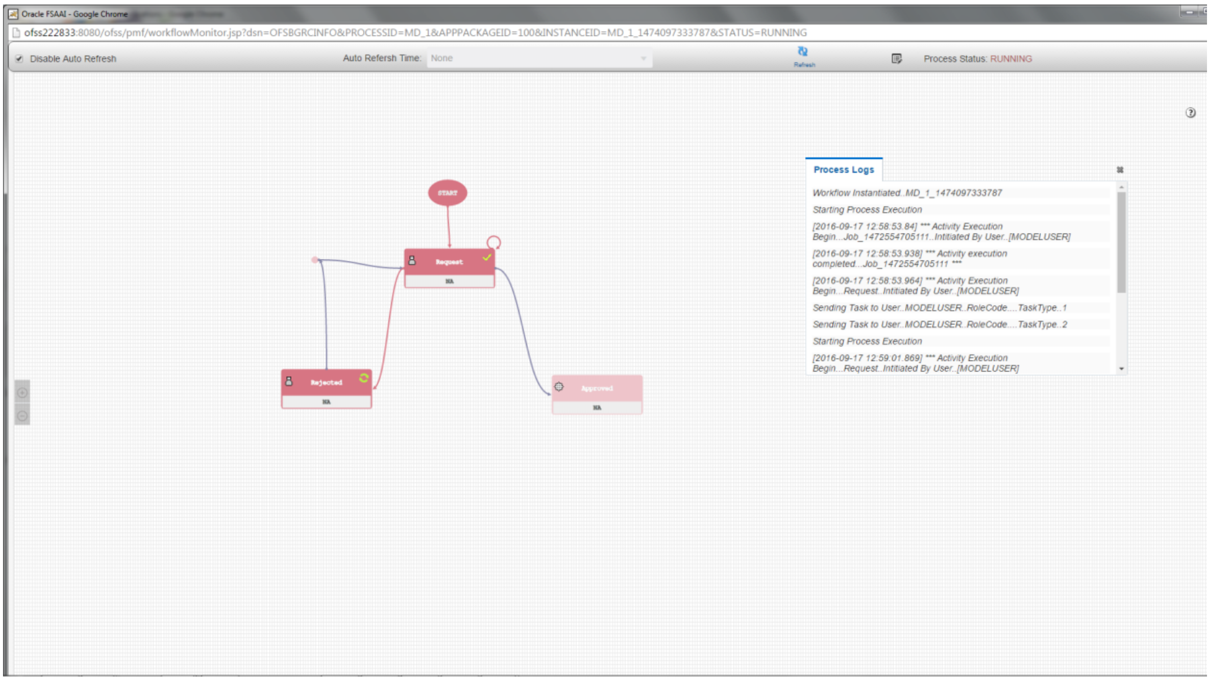
ofss222833:8080/ofss/pmf/workflowMonitor.jsp?dsn=OFSBGRCINFO&PROCESSID=MD_1&APPPACKAGEID=100&INSTANCEID=MD_1_1474097179230&STATUS=RUNNING

☑ Disable Auto Refresh

Auto Refersh Time: None

Process Status: COMPLETED

```
Workflow Instantiated..MD_1_1474097179230
Starting Process Execution
[2016-09-17 12:56:19.295] *** Activity Execution
Begin...Job_1472554705111..Intitiated By User..[MODELUSER]
[2016-09-17 12:56:19.398] *** Activity execution
completed...Job_1472554705111 ***
[2016-09-17 12:56:19.423] *** Activity Execution
Begin...Request..Intitiated By User..[MODELUSER]
Sending Task to User..MODELUSER..RoleCode....TaskType..1
Sending Task to User..MODELUSER..RoleCode....TaskType..2
Starting Process Execution
[2016-09-17 12:57:46.129] *** Activity Execution
Begin...Request..Intitiated By User..[MODELUSER]
```



Oracle FSAAI - Google Chrome
ofss222833:8080/ofss/pmf/workflowMonitor.jsp?dsn=OFSBGRCINFO&PROCESSID=MD_1&APPPACKAGEID=100&INSTANCEID=MD_1_1474097333787&STATUS=RUNNING
Disable Auto Refresh
Refresh
START
Request
NA
Rejected
NA
Approved
NA
Process Logs

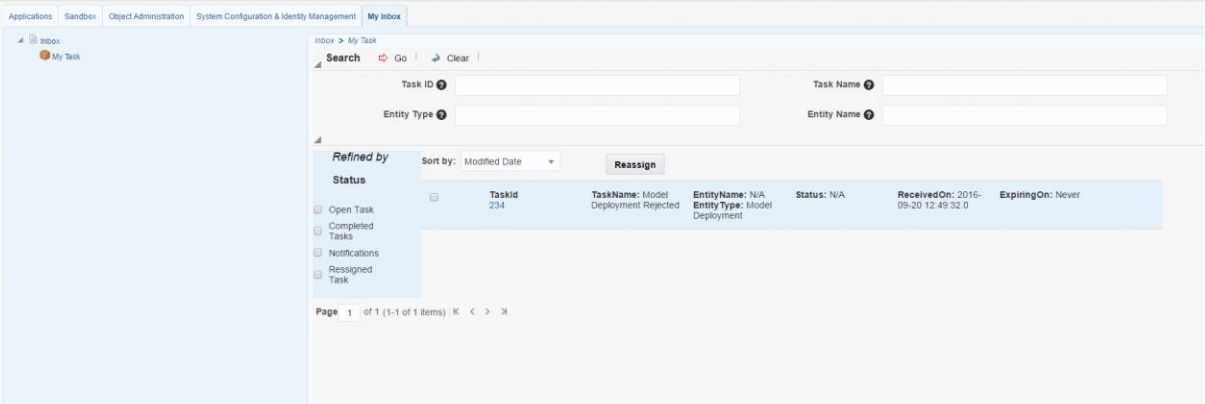
☑ Disable Auto Refresh

Auto Refersh Time: None

Process Status: RUNNING

```
Workflow Instantiated..MD_1_1474097333787
Starting Process Execution
[2016-09-17 12:58:53.84] *** Activity Execution Begin...Job_1472554705111..Intitiated By User..[MODELUSER]
[2016-09-17 12:58:53.938] *** Activity execution completed...Job_1472554705111 ***
[2016-09-17 12:58:53.964] *** Activity Execution Begin...Request..Intitiated By User..[MODELUSER]
Sending Task to User..MODELUSER..RoleCode....TaskType..1
Sending Task to User..MODELUSER..RoleCode....TaskType..2
Starting Process Execution
[2016-09-17 12:59:01.869] *** Activity Execution Begin...Request..Intitiated By User..[MODELUSER]
```

6. The model group will receive a notification that the model was approved or rejected.



Task ID

Task Name

Entity Type

Entity Name

Refined by

Status

□ Open Task

□ Completed Tasks

□ Notifications

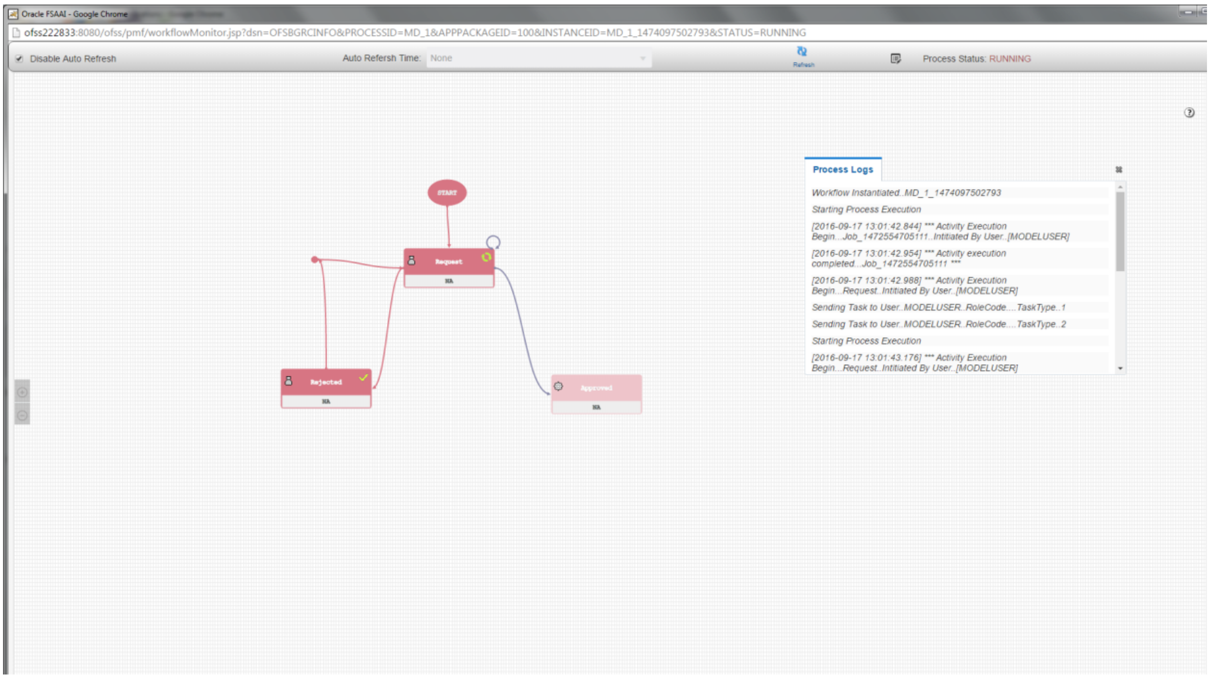
□ Ressigned Task

Sort by: Modified Date

□	**TaskId** 234	**TaskName:** Model Deployment Rejected	**EntityName:** N/A **Entity Type:** Model Deployment	**Status:** N/A	**ReceivedOn:** 2016-09-20 12:49:32.0	**ExpiringOn:** Never

Page 1 of 1 (1-1 of 1 items)

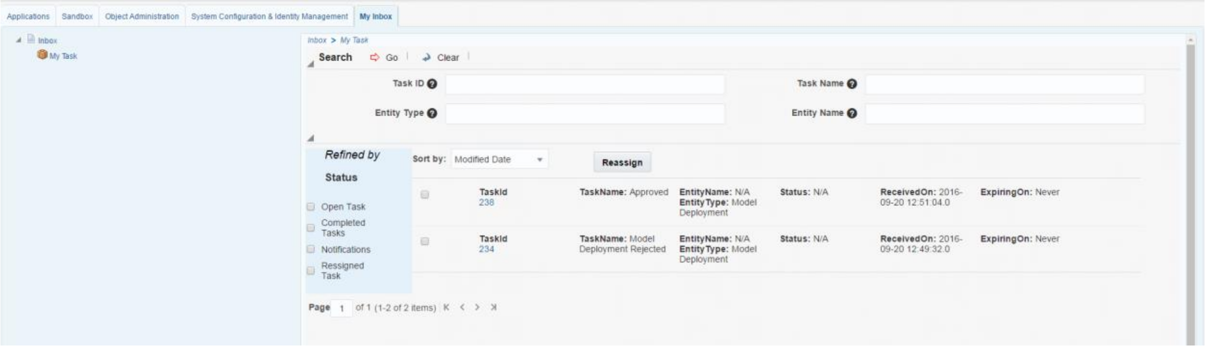
7. In case of rejection, the user can apply for Approval request again by opening the model from the Task ID. It will update the PMF Diagram of the model as can be viewed and send notifications again.



Oracle FSAAI - Google Chrome
ofss222833:8080/ofss/pmf/workflowMonitor.jsp?dsn=OFSBGRCINFO&PROCESSID=MD_1&APPPACKAGEID=100&INSTANCEID=MD_1_1474097502793&STATUS=RUNNING
☑ Disable Auto Refresh
Auto Refersh Time: None
Refresh
Process Status: RUNNING
START
Request
NA
Rejected
NA
Approved
NA
Process Logs

```
Workflow Instantiated..MD_1_1474097502793
Starting Process Execution
[2016-09-17 13:01:42.844] *** Activity Execution
Begin....Job_1472554705111..Intitiated By User..[MODELUSER]
[2016-09-17 13:01:42.954] *** Activity execution
completed....Job_1472554705111 ***
[2016-09-17 13:01:42.988] *** Activity Execution
Begin....Request..Intitiated By User..[MODELUSER]
Sending Task to User..MODELUSER..RoleCode.....TaskType..1
Sending Task to User..MODELUSER..RoleCode.....TaskType..2
Starting Process Execution
[2016-09-17 13:01:43.176] *** Activity Execution
Begin....Request..Intitiated By User..[MODELUSER]
```

8. The model user group will receive the notification of the model being approved or rejected.



Task ID:

Task Name:

Entity Type:

Entity Name:

Refined by

Status

□ Open Task

□ Completed Tasks

□ Notifications

□ Ressigned Task

Sort by: Modified Date

□	**TaskId** 238	**TaskName:** Approved	**EntityName:** N/A **EntityType:** Model Deployment	**Status:** N/A	**ReceivedOn:** 2016-09-20 12:51:04.0	**ExpiringOn:** Never
□	**TaskId** 234	**TaskName:** Model Deployment Rejected	**EntityName:** N/A **EntityType:** Model Deployment	**Status:** N/A	**ReceivedOn:** 2016-09-20 12:49:32.0	**ExpiringOn:** Never

Page 1 of 1 (1-2 of 2 items)

Conclusion

As the project is still in progress a lot of the components of the project are yet to be implemented, like using Oracle Jet and ALTA theme, which will be implemented in the coming weeks. Since OFSAA is a large scale software, rigorous testing will be a part of the development process. The code committed so far has been tested and works with the rest of the product.