

Oracle Application Framework (OAF) Personalizations Done Quicker, Better & Easier in eBS Release 12

By Austin Davis, Chain-Sys Corporation

Objective of this Paper

Oracle Application Framework (OAF) is the Oracle Applications self-service and deployment platform. Oracle e-Business Suite (eBS) Release 12 uses OAF in most of their modules. OA Framework is built on a flexible architecture with metadata-driven UI components, flows and business logic. This makes the UI and business logic extensible. Using this technology and framework, we will show you how to improve functionality of the new OAF pages in Oracle's e-Business Suite Release 12.

Intended Audiences:

(i) Individual contributor (ii) Project team member (iii) Project Manager

High Level Overview

During any business process improvement, change management, user experience improvement or enforcement of a Data Quality Management (DQM) initiative, one of the places to start is the User Interface (UI) of the application. In this case, that would be the forms and OAF pages of eBS Release 12. Many of the old forms have been replaced which creates a dilemma for some as the new OAF forms look nothing like the previous traditional forms. Both the IT and business users must work together to come up with new ways to improve how they work in the UI, to follow business process and comply with new corporate laws and regulations.

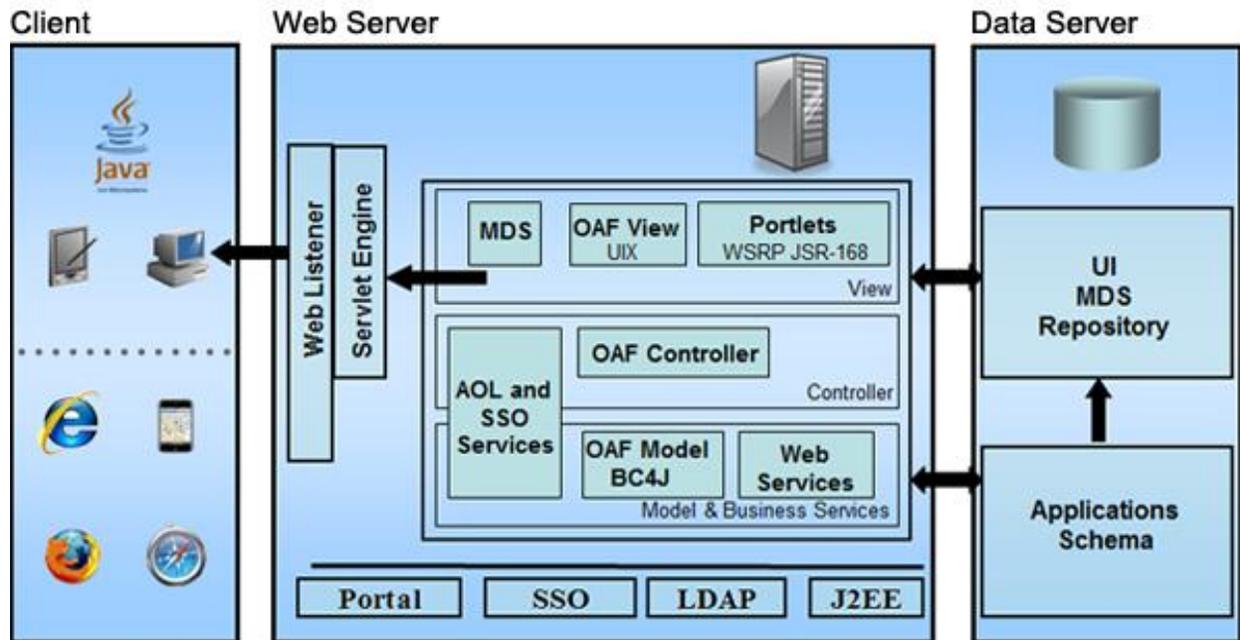
What is OAF?

OAF is the Oracle Applications self-service development and deployment platform described as:

- 100% Java & XML, middle-tier application framework and services for the rapid development & deployment of HTML based applications.
- Follows the Model-View-Controller (MVC) design pattern.
- OA Framework is built on a flexible architecture.
- Metadata-driven UI components, flows and business logic.

- Extensible UI and business logic.

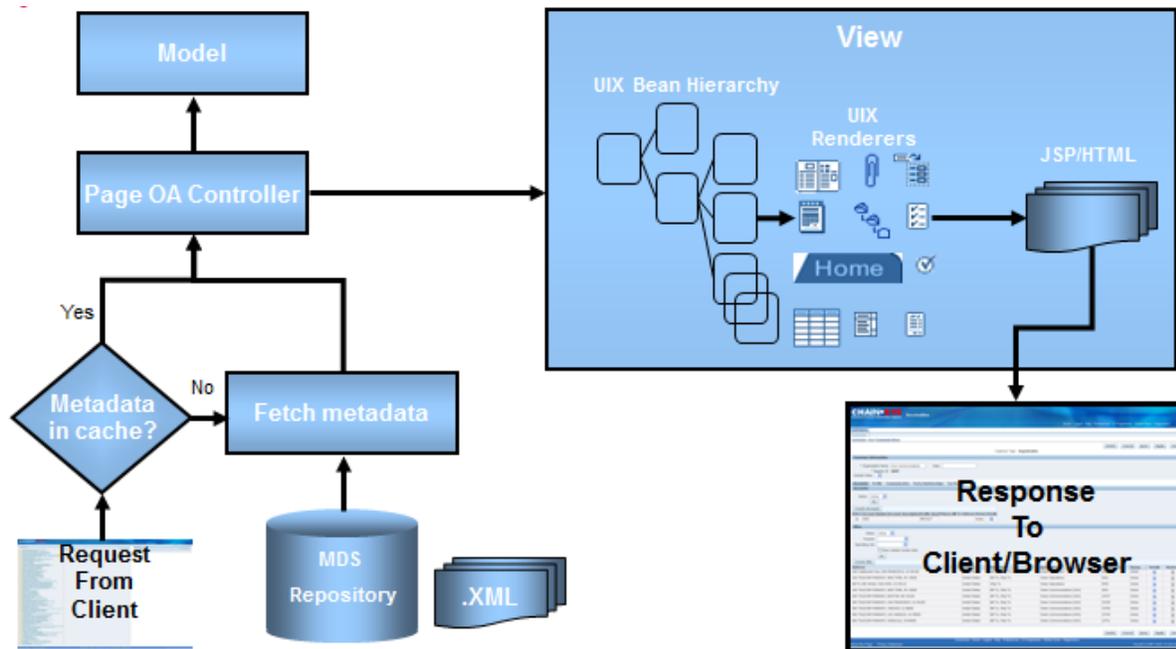
OA Framework Overview



As with any UI, how to extend or personalize is the first question asked by the business. The resulting question is whether to make the changes in OAF or to extend using Fusion technologies such as Oracle Application Development Framework (ADF). The answer depends on the use of Fusion, complexity of extensions and number of extensions. If using Oracle eBS, simple extensions should be done using OAF while more complex extension can be made in ADF and be kept “future proof”—however, there is a level of integration lost in ADF that one must be aware. OAF has no such limitations and for the time until Fusion is well adopted by your organization, it makes sense to maintain use of OAF’s advantages. Further, with adoption of tools and improvements, many of the complex “customization” capabilities can be made as simple configurations which can be performed by IT or the business itself.

OAF Run Time

How does the OAF page execute from a run time perspective? When a user makes a request from the browser, the activity initiates as below:



- Request is received by the JSP (OA.jsp)
- JSP invokes OAPageBean
- OAPageBean creates OAPageContext
- Provides access to the state of the page
- Provides hooks into OA Framework services
- The OAPageContext instance is passed to the controller

Here the OAF Controller takes over where it:

- Delegates some events to the Model
- Performs some data binding
- Determines next page
- Invokes View to present next page

The resulting view is basically a HTML page and would be described as:

- A page would contain one or more region
- A region can contain normal HTML components and other regions.
- The JSP/HTML page is rendered from an UIX.

The UIX is formed from Meta Data Service (MDS = Metadata Repository + Metadata cache). MDS is a dictionary containing data about the UI components to be displayed in a page. The MDS has utilities to convert the metadata into XML documents with each component as nodes.

How to make it Better

What does all this mean? The typical user only wants to be able to eliminate some keystrokes from their UI or the business may want to add special form validations to comply with the business process. With a better understanding of OAF from an architectural and run-time perspective, we can quickly see the need to improve the OAF pages to gain the customization capability from the traditional Forms environment without having to perform programming. Given the meta-data driven nature of OAF and its highly extensible business logic capabilities, some re-thinking will allow much more advanced customizations (“augmentations”) and extensions than previously thought possible. Here are some high-level capabilities the new framework engine should have:

- Collects information about the components and its attributes of the page to be “augmented” from MDS
- Stores the “augmentations” in a data schema.
- Creates dynamic controller classes by extending the Controller of the page to be “augmented”
- Binds this controller information into MDS as OA Extension.
- Notifies the application cache about the state of the page has been modified.

This would result in an improved run-time with the following flow:

- When a user invokes augmented OAF page, OA Page context invokes NEW Controller from MDS
- Invokes Parent Controller followed by Java Augmenter engine to fetch augmented data
- New Engine evaluates the augmentations to the page based on the events and conditions
- Other steps would be handled by the OAF to reflect the augmentations

The achievement is additional capabilities previously left only in the hands of capable technical developers. Now with simple configurations, business users or IT staff can efficiently manage and perform the following types of extensions:

- Rules: To change the UI behavior
- Zoom: To zoom from OAF page to any OAF Page or Form or external application
- List of Values: To change an input field to LOV field or Combo Box
- To restrict the List of Values in a Combo Box
- Table Region: To add a tooltip to a column of table
- To sort a column of the table
- Add Actions to buttons

These “augmentations” will allow for the quick adoption of many enterprise objectives for process improvement, business process change, data validation, data quality management and improved user interface experience. Additionally having proper management capability with the right productivity tools is important. All extensions (“augmentations”) should be easily manageable:

- All “augmentation” types can be done on a condition basis

- Search all the augmentations done on the instance
- Export and Import all (or) any augmentations
- Easily enabled or disabled

Conclusion

From this paper you have learned several key concepts regarding OAF from the architecture to the ability to extend it further. These augmentations are readily available for use in the application appAUGMENTER. Leveraging this appAUGMENTER framework can greatly improve your organization's initiatives to:

- ✓ **Extend eBS OAF Pages to Meet Business Requirements.**
- ✓ **Meets SOX compliance needs for OAF pages.**
- ✓ **Management of Workflow Process**
- ✓ **Continuous Improvement for efficiency**
- ✓ **Secure access to data**

For now, until fusion is fully adopted, OAF with appAUGMENTER can be complete integrated framework for extending OA pages. With ease of use, significant business process benefits along with manageability from an IT perspective, this type of solution can provide the complete framework for extending and maintaining your OA pages while minimizing any development and deployment timelines.