

# Oracle ALTA UI : Smashing UI for Web and Mobile

**Luc Bors**  
**eProseed**  
**Utrecht - Netherlands**

## **Keywords**

ALTA UI, Web, Mobile, CSS, ADF, Templates

## **Introduction**

Oracle uses the Oracle Alta UI to develop the most recent Oracle Cloud products, the latest versions of Oracle Fusion Applications, a number of innovative mobile applications. With fewer elements, cleaner designs and minimal UI chrome, the Oracle ALTA UI adds to the faster loading of critical content and helps users to keep focus on data. This session discusses the key concepts you should consider for creating modern user interfaces for your web applications as well as your mobile applications. It then demonstrates, based on a real life scenario, how to achieve these user interfaces with the Oracle ADF Faces feature of Oracle Application Framework (Oracle ADF) and Oracle MAF. In this article, you will first learn about the key concepts of Mobile and Web UI's. After that, I will introduce Oracle ALTA UI and how Oracle ALTA UI can be used to implement these concepts. First, I will show how to apply the ALTA UI concepts to an Oracle ADF Browser Application. Next, I will do the same for an Oracle MAF mobile app. Finally, the audience will learn a way to embed the ADF Browser application to the Oracle MAF app and have a smooth transition between both UI's.

## **Oracle ALTA UI**

Oracle ALTA UI was introduced by Oracle in 2014. There has been a lot of noise on ALTA UI since, and many designers and developers are adopting the ALTA UI in their applications. Now what if you are new to ALTA UI, and want to start using it? What is ALTA UI, where do you start and how can you use your tools to create ALTA UI applications?

There is only that much I can tell in this short article, so I will give you an overview of ALTA UI and introduce you to the basic and common techniques to start using Oracle ALTA UI to create modern looking Oracle ADF Web and Oracle MAF Mobile applications. The examples in this article are based on Oracle's WorkBetter reference application.

## **What is Oracle ALTA UI?**

Oracle ALTA UI is a design system and set of guidelines that enables developers to deliver modern, simple, and more engaging UIs across web-based and on-device mobile applications. It is the new standard for Oracle Applications and Oracle Cloud-based services and is supported by Oracle's development tools and frameworks for both mobile and web-based applications. ALTA UI brings a fresh style and modern clean layouts with minimal UI chrome.

From a design and end user perspective Oracle ALTA UI is based on four key principals. This starts with "Mobile first design". As a consequence, buttons, menus, text, and other UI components are larger. Larger components offer more generous touch targets and easier to read displays. Next, ALTA UI stands for simple and uncluttered layouts. This enables users to focus on data instead of having to figure out where to even find the relevant data on a page. Then, the principal of clear information hierarchy adds to usability of web applications so the user can easily find where they want to go. Finally Oracle ALTA UI aims to engage users with visual content. When I look at myself and how I use my mobile device, I need instant information. I rather skim web pages than fully read them. This

means we need to approach modern application design differently. Compared to other UI designs, Oracle ALTA UI is more graphical in nature, moving away from a traditionally text based UI. A picture paints a thousand words.

From a technical and developer perspective, Oracle ALTA UI is based on the following foundations. First, ALTA UI uses a Simplified component structure and more open space keep focus on data. Besides that, the use of less chrome means faster loading of critical content. Further, ALTA UI supports both browser and mobile technology, and finally it adds flexibility with simpler, modular, components.

In the next sections I will go into the technical details and explain how the Oracle tools and frameworks support Oracle ALTA UI.

## Oracle ALTA UI for Web

Oracle ALTA UI can be used to create modern looking web applications. In contrast with the Oracle Simplified UI and the applications that use the previous skins such as *Skyros*, the ALTA UI stands for a light UI with clear and concise colors. To build ADF web applications that use ALTA UI there are several tools and tips that you can use.

### Skins and templates

The typical ALTA UI look and feel for an ADF application is implemented by a skin that ships with Oracle ADF. That skin is the *alta* skin. When you want to build a new ADF application in JDeveloper (in version 12.1.3) the skin of that application defaults to Skyros as can be found in the applications' trinidad-config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<trinidad-config xmlns="http://myfaces.apache.org/trinidad/config">
  <skin-family>skyros</skin-family>
  <skin-version>v1</skin-version>
</trinidad-config>
```

To use the Oracle ALTA UI in your application, simply change *skyros* to *alta*. If there is need to, you can easily create your own extensions on top of the alta skin. The mechanism is the same as extending any other skin.

**NOTE:** You can turn on the Alta skin for existing applications. However, to get the full UI experience it is better to start with a re-design of the UI. Oracle ALTA UI is not just about the colors, it is about the interaction!

ALTA UI applications typically use the tablet first approach. In order to use the tablet first approach, you can use the *tablet first template* that ships with Oracle JDeveloper and Oracle ADF. The Tablet First template will have your ADF Faces page follow the Oracle Alta UI design principles. When you create a new page, simply base that new page on the "*table first template*" as shown in figure 1.

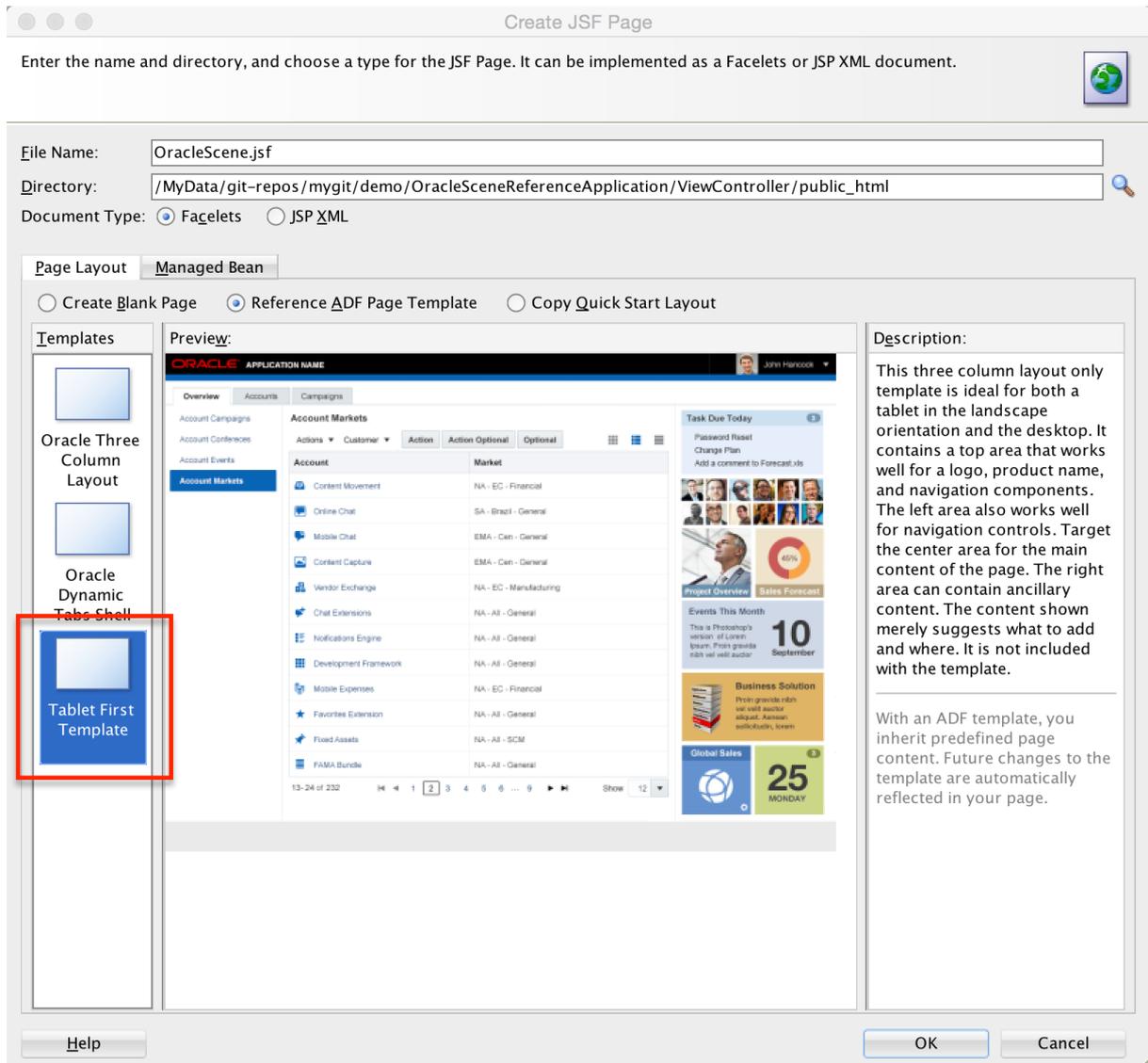


Figure 1: Selecting the Tablet First Template

This template consist of several predefined area's (figure 2) such as **branding title**, **application navigation bar** and a **content** facet, the latter which is typically used to put the real functional content of your new ALTA UI application.

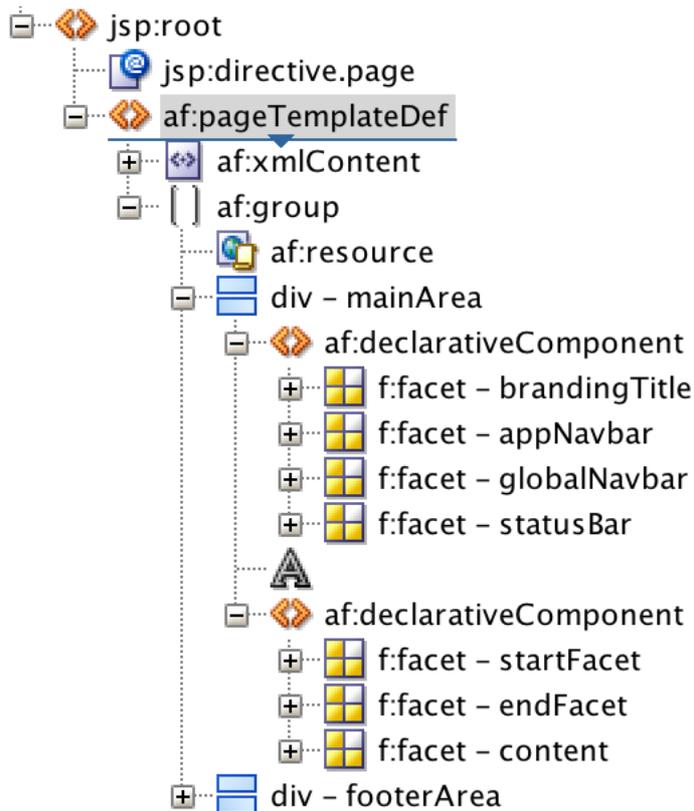


Figure 2: The Tablet First Template

You can now construct your ALTA UI page by adding the desired components to the predefined facets. Start by adding buttons to the appNavBar to control your application and after that add content to the content facet.

```
<f:facet name="appNav">
  <af:button text="Dashboard" id="b1" styleClass="AFAppNavbarButton"/>
  <af:button text="People" id="b2" styleClass="AFAppNavbarButton" />
<af:button text="Organizations" id="b3" styleClass="AFAppNavbarButton"/>
</f:facet>
```

Note the use of the “*AFAppNavbarButton*” StyleClass. This StyleClass is part of the *alta* skin and is used for a styling the buttons that appear on and application navigation bar. With the skin and template in place, your new ADF application will already look like ALTA UI.

### ADF UI Components

There are several ADF UI components that enable you to create the ALTA UI application as it was meant to be. I will give you two examples. First there is the use of responsive components. In a responsive web application there is one layout that always changes, based on the device it is rendered in. When you flip you tablet from landscape to portrait, you want to make sure the UI behaves accordingly and adjust to the available real estate. In order to have your ADF application behave in a

way that it adjusts to the screen size, you need to know what ADF components you should use. The key to this is the use of

Also one simple entry in a stylesheet can help you implement his responsive behavior. The value **“inline-block”** for the **“display”** instructs the browser to fill the boxes fitting the page width and wraps them nicely to the next line.

```
.LayoutStyleDisplayBlock {
    display: inline-block;
}
```

This style class can then be used in the panelGroupLayout components or other components.

```
<af:panelGroupLayout id="pb2"
    styleClass=" LayoutStyleDisplayBlock">
```

Then there is the use of cards. These are “flip-able” components that give you twice the amount of real estate. From an ADF perspective, these components do not exist. This however does not mean that they cannot be created. An excellent example of this can be found in Oracle’s reference application “WorkBetter”, where cleverly a deck component with two panelboxes are use to create a flipcard.

```
<af:deck id="d1"
    displayedChild="#{item.bindings.active.inputValue==true?'back':'front'}">
    <af:transition triggerType="forwardNavigate" transition="flipStart"/>
    <af:transition triggerType="backNavigate" transition="flipStart"/>
    <af:panelBox text="FrontCard" id="front" showHeader="never"
        showDisclosure="false">
        ... → content
    </af:panelBox>
    <af:panelBox text="BackCard" id="back" showDisclosure="false"
        showHeader="never">
        ... → content
    </af:panelBox>
</af:deck>
```

When a user clicks the arrow (in the lower right corner of the boxes in figure 2) the box flips and the “backside” is shown. The deck component has the built-in transitions to mimic the flip effect.

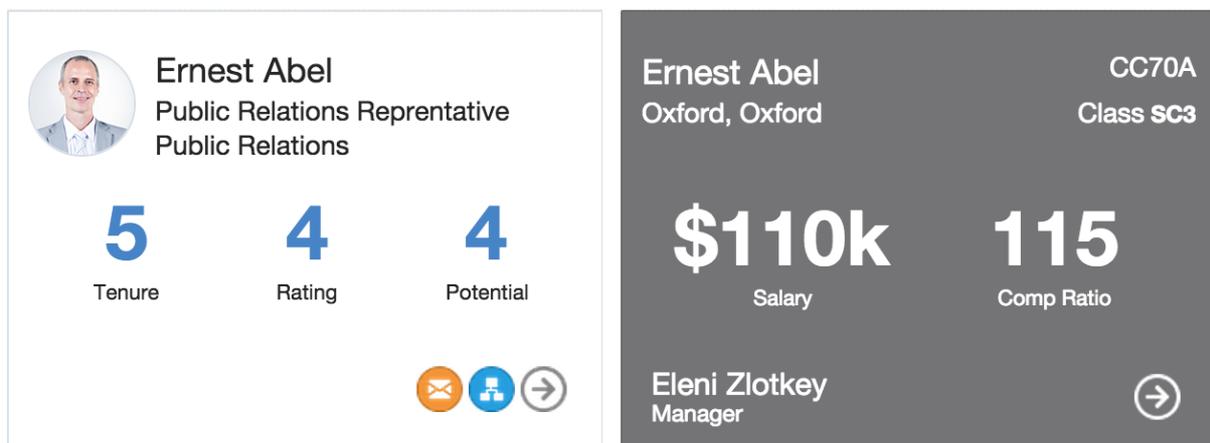


Figure 3: The Flipcard

## Skin extensions

The Oracle ALTA UI design guidelines describe a very concise use of colors and fonts. In order to use these guidelines and implement them in your ADF application, you would typically create an extension to the *alta* skin. Lets look at the blue numbers in figure 3. Those numbers are big and blue. A skin-extension definition and StyleClasses in the accompanying Stylesheet accomplish this. The skin-extension is defined in the Trinidad-skins.xml

```
<skins xmlns="http://myfaces.apache.org/trinidad/skin">
  <skin>
    <id>YourSkin.desktop</id>
    <family>YourSkin</family>
    <extends>alta-v1.desktop</extends>
    <render-kit-id>org.apache.myfaces.trinidad.desktop</render-kit-id>
    <style-sheet-name>resources/skins/YourSkin.css</style-sheet-name>
  </skin>
</skins>
```

This skin extension makes sure that the application uses the *alta* skin plus everything that you define in a new css file, in this case named YourSkin.css. For the example in figure 3 we use a big and bold font, and a typical blue color, both defined in separate Style classes. This is good practice so the Style classes can be reused separately.

```
. BigBold {
  font-size: 40px;
  font-weight: bold;
}
. BlueColor {
  color: #4a85c5;
}
```

Finally use those style classes in your ADF components and you are all set.

```
<af:outputFormatted value="5" styleClass="BigBold BlueColor">
```

With these tips and tricks you now know everything you need to start building your own ALTA UI ADF web application. Now what about Mobile?

## Oracle ALTA UI for Mobile

Oracle Mobile Application Framework (MAF) has the ALTA UI look and feel built-in. There is no need to do any configuration in order to use the ALTA UI. Also the responsiveness is integrated in the MAF Framework. An MAF app will automatically adapt to the device it is running on thus enabling a good UX on both phone and tablet.

Knowing the MAF components enables you to create MAF apps that follow the ALTA UI guidelines. As with ADF applications, you can combine existing components to create the desired behavior. For instance ALTA UI requires large “touch” areas for optimal user interaction. The List of employees used in the WorkBetter demo is an excellent example of this. Here all list items in the listview are displayed in a tablelayout component with multiple rows. By using the height, width and rowspan properties for the cells you can achieve the nice layout as displayed in figure 4.

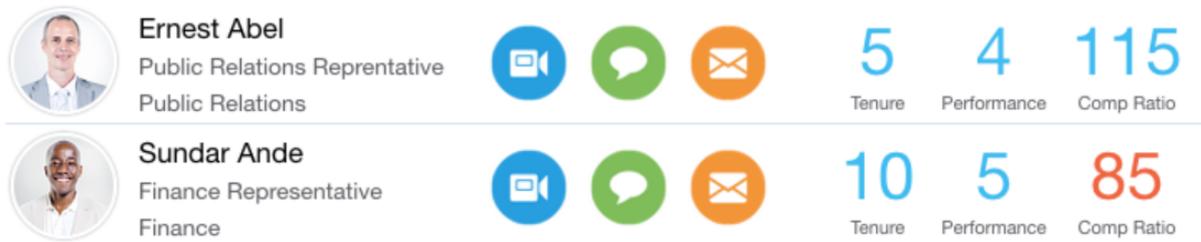


Figure 4: Listview

```

<amx:listView id="list" var="row">
  <amx:listItem id="li1">
    <amx:tableLayout width="100%" id="tl2" shortDesc="Table">
      <amx:rowLayout id="rl2">
        <amx:cellFormat height="44px" shortDesc="Cell">
          <amx:image source="pics/#{row[Image]}" id="i1"
            shortDesc="List Image"/>
        </amx:cellFormat>
        <amx:cellFormat>
          -> content
        </amx:cellFormat>
        <amx:cellFormat id="cf2" width="50%">
          <amx:panelGroupLayout id="pgl2" layout="horizontal">
            all logos for Skype, FaceTime etc
          </amx:panelGroupLayout>
        </amx:cellFormat>
        <amx:cellFormat id="cf14" rowspan="3" >
          <amx:panelGroupLayout id="pgl3" layout="horizontal">
            all numbers and cfi's
          </amx:panelGroupLayout>
        </amx:cellFormat>
      </amx:rowLayout>
      <amx:rowLayout id="rl3">
        <amx:cellFormat>
          title
        </amx:cellFormat>
      </amx:rowLayout>
      <amx:rowLayout id="rl4" >
        <amx:cellFormat>
          -> subtitle
        </amx:cellFormat>
      </amx:rowLayout>
    </amx:tableLayout>
  </amx:listItem>
</amx:listView>

```

Adding extra styles on top of **alta** is achieved in more or less the same way as in ADF applications. You can create your own skin extensions and add extra styles to that, or assign a stylesheet to an MAF feature. There is always a, not preferred but working solution to add styles directly to your components, by using the inlineStyle attribute. This is how Oracle implemented it in the WorkBetter demo sample for MAF.

```

<amx:outputText value="5" styleClass="BigBold"
  inlineStyle="color:#47BDEF"/>

```

More details on how to work with skins and skin additions can be found in the Oracle Mobile Application Framework Developer Guide.

## Summary

When you are about to design or build a new Oracle ADF application you should definitely consider using Oracle Alta UI. This ensures a uniform look and feel and also a uniform User Experience for both Oracle MAF applications and Oracle ADF Browser based web applications. The key to success is to use the Oracle ALTA UI website in order to understand the principals on which ALTA UI is based and how to adopt these principals in your new ADF and MAF applications. From designers to developers and from color settings to navigation patterns, all is described on the ALTA UI website. When following these guidelines of how to build efficient and usable ALTA applications you are on your road to success.

## Resources

Oracle ALTA UI website:

<http://www.oracle.com/webfolder/ux/middleware/alta/index.html>

Oracle Work Better Application Live Demo:

<http://jdevadf.oracle.com/workbetter/faces/index.jsf>

Download Oracle Work Better Application:

<http://download.oracle.com/otn/java/jdeveloper/12.1.3.0.0/WorkBetterFaces.zip>

The Work Better demo for Oracle MAF is shipped with MAF and can be found in the public-samples folder: <JDEVELOPER\_HOME>/jdeveloper/jdev/extensions/oracle.maf/Samples

## Contact:

Luc Bors  
eProseed  
Dijkgraafpolder, 3  
3991 ZC Houten, Netherlands

Telefon: +31 (0) 6 52 32 72 40  
E-Mail [luc.bors@eproseed.com](mailto:luc.bors@eproseed.com)  
Internet: [www.eproseed.com](http://www.eproseed.com)