ADF Code Corner

015. How-to configure a custom splash screen in ADF Faces

Abstract:

Splash screens, or enduser entertainment as I like to put it, show at the beginning of initial page loading until all resources become available for the user to start his work. ADF Faces RC components provide a splash screen that by default shows a rotating "O" for Oracle with a message letting the user know that application initialization is in progress. For your production applications, you may want to change the splash screen to show a custom image that fits to your corporate identity and logo. This how-to explains how to do this.

twitter.com/adfcodecorner

Author: Frank Nimphius, Oracle Corporation
twitter.com/fnimphiu
07-MAY-2008

Oracle ADF Code Corner is a loose blog-style series of how-to documents that provide solutions to real world coding problems.

Disclaimer: All samples are provided as is with no guarantee for future upgrades or error correction. No support can be given through Oracle customer support.

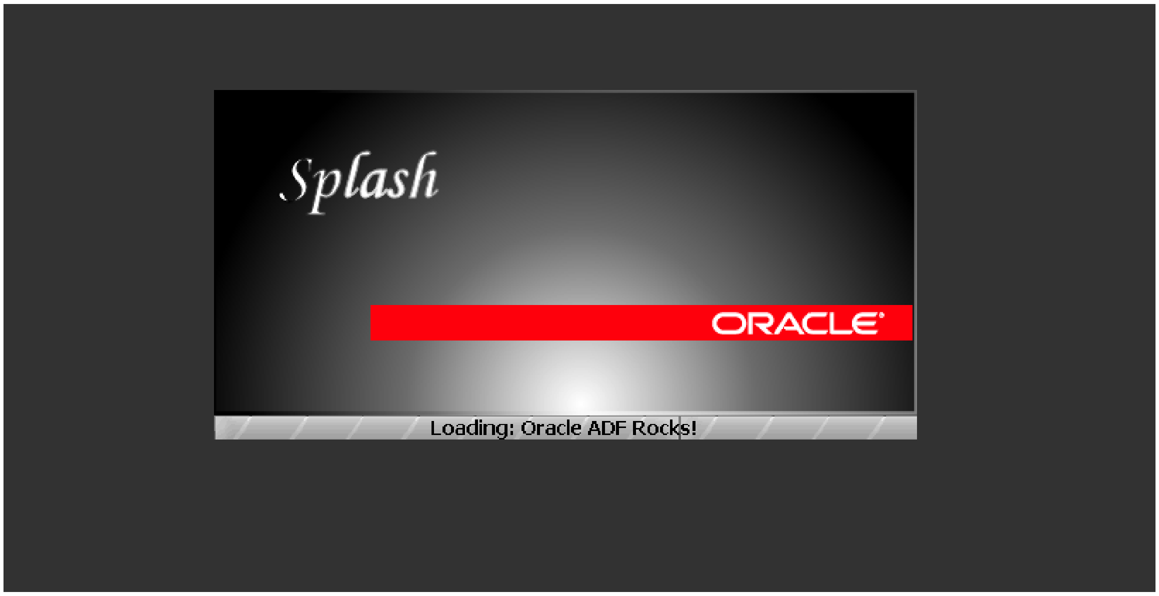
Please post questions or report problems related to the samples in this series on the OTN forum for Oracle JDeveloper: http://forums.oracle.com/forums/forum.jspa?forumID=83

Introduction

The splash screen in ADF Faces RC is configured within the skin definition. This means that to change the splash screen you need to define a custom skin definition. While this sounds like a lot of work for a simple change it is not. JDeveloper 11 has simplified custom skin development by allowing you to inherit styles from an existing skin, for example the default Oracle fusion look and feel.

In this example I explain how you change the splash screen only, plus the message that gets displayed upon loading. The workspace that you can download at the end of this document was developed with a JDeveloper 11 build later than TP4, which is available on OTN to the time of writing. However, I expect this sample to work in TP4 as well.

Following this how-to document, or running the sample workspace, will make the Splash screen render as shown below:



The following changes are applied, compared to the default splash screen:

- New image
- New loading message and other location
- Animated loading gif as the background of the loading text

Configuring the Skin

Skins are configured in the trinidad-skins.xml file and referenced in the trinidad-config.xml file. While the trinidad-config.xml file exist by default, you need to manually create the trinidad-skins.xml file in the WEB-INF directory of the view layer project, which can be done from within Oracle JDeveloper.

WEB-INF/trinidad-skins.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<skins xmlns="http://myfaces.apache.org/trinidad/skin">
    <skin>
        <id>adcs.desktop</id>
        <family>adcs</family>
        <render-kit-id>
            org.apache.myfaces.trinidad.desktop
        </render-kit-id>
        <style-sheet-name>skins/adcs.css</style-sheet-name>
        <extends>blafplus-rich.desktop</extends>
        <bundle-name>com.oracle.adcs.AdcsResourceBundle</bundle-name>
    </skin>
</skins>
```

The name of the skin ID and the skin family can be freely chosen and needs to be referenced in the trinidad-config.xml file. The following settings are used within this example

- **id** - the skin name identifier with a .desktop extension
- **family** - the name of the skin used at runtime to reference the skin css file
- **render-kit-id** - the Trinidad desktop renderkit. This value doesn't change and is the same for all custom skins
- **style-sheet-name** - the reference to the style sheet that contains the CSS file with the skin information. The CSS file is relative to the public_html directory
- **extends** - identifies the skin to inherit the settings from. In this case I reference Oracle default look and feel, which is blafplus-rich.
- **bundle** - to override the loading message, I need to reference a message bundle that contains the string key. The message bundle is a Java class that needs to be available in the project or classpath and that extends ListResourceBundle. For your convenience, the bundle class provided with this example contains all message keys and their default message strings. This way it becomes easy for you to use this example as a starting point for your own skin development.

The trinidad-config.xml file reference for this skin looks as follows:

```
<?xml version="1.0" encoding="windows-1252"?>
<trinidad-config xmlns="http://myfaces.apache.org/trinidad/config">
  <skin-family>adcs</skin-family>
</trinidad-config>
```

Defining the Skin

The skin definition is in a CSS file located in a directory located below the public_html folder of the view project. It uses the skin selectors defined for the splash screen for all aspects of the required modification:

Note: Skins can be deployed in JAR files, which is explained in one of the referenced documents mentioned at the end of this how-to.

```
/* Customize the Splash Screen*/

af|document::splash-screen-content{
 background-color:transparent;
}
af|document::splash-screen{
 background-color:transparent;
}

af|document::splash-screen-content{
 -tr-inhibit:background-image;
 -tr-inhibit:background-repeat;
 border:transparent;
 background-color:transparent;
}


af|document::splash-screen-message{
 display:table-row;
 text-align: center;
 color:Black;
 background-image:url("images/progress_bar.gif");
}

af|document::splash-screen-cell{
 background-color:transparent;
}
af|document::splash-screen-icon{
  content:url("images/splash.gif");
}
```

Skinning explained

- **background-color:transparent** - the default skin contains colored background areas that you need to override explicitly or inhibit inheritance. I decided for overriding the style because it is a quick change
- **-tr-inhibit:background-image;** - the tr-inhibit option allows you to prevent inheritance for a specific selector or even all settings defined by the parent skin. In this case I don't want the background image to repeat. The image is located in a folder relative to the skin CSS file

- **content:url("images/splash.gif");** - Set my image to be the icon loaded upon splash display. In JDeveloper 11 builds later than TP4, you have the ability to enable a new skin extension (Tools-->Preferences-->CSS) that then provides you with syntax help and code completion; plus an option to select the image from the filesystem, which is less error prone than typing in the required image reference
- **display:table-row;** .- CSS code that changes the layout of the location in which the loading message is show. By default the load message is on the right end of the splash image. Changing the display option renders it at the bottom. In addition I added an animated background image because its all about end user entertainment, isn't it ?

Download the Example

You can download the JDeveloper 11 sample workspace from the ADF Code Corner site:

http://www.oracle.com/technetwork/developer-tools/adf/learnmore/index-101235.html

It also contains the resource bundle class that I mentioned and that I didn't further explain in this document.

RELATED DOCOMENTATION

☒	Skin selectors - http://download.oracle.com/docs/cd/E15523_01/apirefs.1111/e15862/toc.htm
☒	Skinning in the Oracle product documentation - http://download.oracle.com/docs/cd/E15523_01/web.1111/b31973/af_skin.htm#BAJFEFCJ
☒	Oracle Fusion Developer Guide – McGraw Hill Oracle Press, Frank Nimphius, Lynn Munsinger http://www.mhprofessional.com/product.php?cat=112&isbn=0071622543