

## **Oracle® Fusion Middleware**

Developer's Guide for Application Development Framework  
Mobile

11g Release 1 (11.1.1)

**E10140-01**

August 2008

E10140-01

Copyright © 2008 Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

---

---

# Contents

<b>Preface</b> .....	v
Audience .....	v
Documentation Accessibility .....	v
Related Documents .....	vi
Conventions .....	vi
<b>1 Overview</b>	
1.1 Introduction to ADF Mobile .....	1-1
1.1.1 About Java Server Faces and ADF .....	1-1
1.1.2 Developing Mobile Applications Using ADF Mobile .....	1-2
1.2 Supported Platforms .....	1-2
<b>2 Configuring the ADF Mobile Environment</b>	
2.1 Introduction to the ADF Mobile Development Environment .....	2-1
2.2 How to Configure the Environment for ADF Mobile Application Development .....	2-1
2.3 How to Create an ADF Mobile Application .....	2-4
2.3.1 Creating a Mobile JSF Page .....	2-5
2.4 Testing the Application .....	2-7
2.4.1 How to Test ADF Mobile Applications .....	2-8
2.4.2 Browser Settings .....	2-10
<b>3 Component Support</b>	
3.1 Introduction to Apache My Faces Trinidad Components .....	3-1
3.1.1 Supported Features .....	3-1
3.1.2 Partial Page Rendering for Microsoft Pocket Internet Explorer .....	3-1
3.1.3 Dialogs .....	3-2
3.1.4 Rendering Specific to the BlackBerry Browser 4.x .....	3-2
3.2 Input Components .....	3-2
3.2.1 Creating Input Text Fields .....	3-2
3.2.2 Creating Lists .....	3-2
3.3 Output Components .....	3-3
3.3.1 Displaying Text .....	3-3
3.3.2 Displaying Images .....	3-3
3.3.3 Showing (or Hiding) Components .....	3-4
3.4 Layout Components .....	3-4

3.4.1	Using Page Templates.....	3-4
3.4.2	Managing the Page .....	3-4
3.4.3	Laying Out Sections of the Page.....	3-5
3.4.4	Inserting Spaces .....	3-5
3.5	Navigation Components.....	3-5
3.5.1	Creating Buttons .....	3-6
3.5.2	Creating Links .....	3-6
3.5.3	Navigation Components.....	3-6
3.6	Tables and Trees.....	3-6
3.6.1	Creating Tables .....	3-7
3.6.2	Creating Trees .....	3-7
3.7	Unsupported Components and Attributes .....	3-7
3.7.1	Unsupported Components.....	3-7
3.7.2	Unsupported Attributes .....	3-8

---

---

# Preface

This guide describes the components available for Application Development Framework Mobile (ADF Mobile) and how to use them to develop applications.

## Audience

This document is intended for developers of browser applications.

## Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, see the Oracle Accessibility Program web site at

<http://www.oracle.com/accessibility/>

### Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

### Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these web sites.

### TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

## Related Documents

You can also find information on ADF in *Oracle Application Development Framework Developer's Guide*.

## Conventions

The following text conventions are used in this document:

<b>Convention</b>	<b>Meaning</b>
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

This chapter provides an overview of Oracle Application Development Framework Mobile (ADF Mobile). It includes the following sections:

- [Section 1.1, "Introduction to ADF Mobile"](#)
- [Section 1.2, "Supported Platforms"](#)

## 1.1 Introduction to ADF Mobile

Oracle Application Development Framework Mobile (ADF Mobile) is a standards-based framework that enables the rapid development of enterprise mobile applications. The current version of ADF Mobile extends Oracle ADF to browsers running on mobile devices. Because ADF Mobile is built upon the component model of Java Server Faces (JSF), you can quickly develop applications for mobile browsers. ADF Mobile's mobile-specific extensions to JSF enable you to develop mobile applications using the same methodologies for developing JSF applications for the desktop.

### 1.1.1 About Java Server Faces and ADF

Java Server Faces (JSF) is a standard specified by JSR-127 that enables developers to create applications using pre-built components that define functionality and behavior. JSF provides a clean Model-View Controller (MVC) mechanism that simplifies the development of Web applications through its renderkit, which converts components both to and from a specific markup language. The renderkit's renderers abstract the production of markup and responses to browser requests by generating the markup representations of components and the way in which these components should interpret browser requests.

JSF development focuses on components, not markup. Using JSF, you create a JSP page containing JSF component tags. When a user visits this page (through the FacesServlet) JSF uses the Renderkit specified by the user's device to encode the markup for the appropriate output. For example, if the user's device specifies HTML for a desktop browser, then the Renderkit's markup encoding results in an HTML page. In addition to rendering appropriate content, JSF supports user interaction.

Application Development Framework (ADF) is built on the standard JSF technology and provides the following:

- A large component set (since JSF provides only basic components).
- Renderers that support these components in HTML browsers, including a rich renderkit for applications using AJAX (Asynchronous JavaScript and XML) technologies.

- Converters, validators, and events.

### 1.1.2 Developing Mobile Applications Using ADF Mobile

You can use the same programming model and component set for developing desktop browser applications to develop mobile browser applications for mobile devices. ADF Mobile application development is almost identical to ADF Web application development, except that ADF Mobile application development uses only mobile JSF pages that consist of Trinidad components.

---

---

**Note:** You cannot use ADF Faces components to develop an ADF Mobile application. You must use Apache MyFaces Trinidad components.

---

---

Developing mobile browser applications for mobile devices with ADF Mobile leverages the same methodologies used in developing JSF applications for the desktop with a few specific mobile extensions. With support for over 60 Apache MyFaces Trinidad components, you can build applications with the rich component set, each of which renders appropriately for small-screen mobile devices. In this way, you can reuse the desktop browser application's model and controller layers to assemble a new view layer for mobile devices by using similar Apache MyFaces Trinidad components.

#### Improving Performance

The PDA component renderers have also been optimized to minimize the payload of the Web page sent to the device for improved performance over wireless networks. In mobile environments with high-latency and low-bandwidth wireless networks, partial-page rendering is essential in providing end-users with an efficient application. For Microsoft Windows Mobile 6, ADF Mobile supports partial-page rendering for certain components to minimize the amount of data requested from the server and improve the responsiveness of the applications.

## 1.2 Supported Platforms

ADF Mobile supports Apache MyFaces Trinidad components for the following devices:

- Microsoft Windows Mobile 5 and 6, Microsoft Pocket Internet Explorer
- RIM BlackBerry Browser 4.x



---

---

## Configuring the ADF Mobile Environment

This chapter describes how to configure the environment for ADF Mobile applications as well as how to build and test ADF Mobile applications.

This chapter includes the following sections:

- [Section 2.1, "Introduction to the ADF Mobile Development Environment"](#)
- [Section 2.2, "How to Configure the Environment for ADF Mobile Application Development"](#)
- [Section 2.3, "How to Create an ADF Mobile Application"](#)
- [Section 2.4, "Testing the Application"](#)

### 2.1 Introduction to the ADF Mobile Development Environment

ADF Mobile application development is almost identical to ADF Web application development, except that ADF Mobile application development uses only mobile JSF pages that consist of Apache MyFaces Trinidad components.

To create an ADF Mobile application:

- Configure the environment by creating an application and project.
- Add a Web project
- Add the JSF pages using Apache MyFaces Trinidad components

**Read this chapter to understand:**

- How to create a Web project that contains technologies that support mobile applications.
- How to implement an ADF Mobile application by adding JSF pages to the Web project that are built with Apache My Faces Trinidad components.
- How to select the appropriate testing environment for an ADF Mobile application.

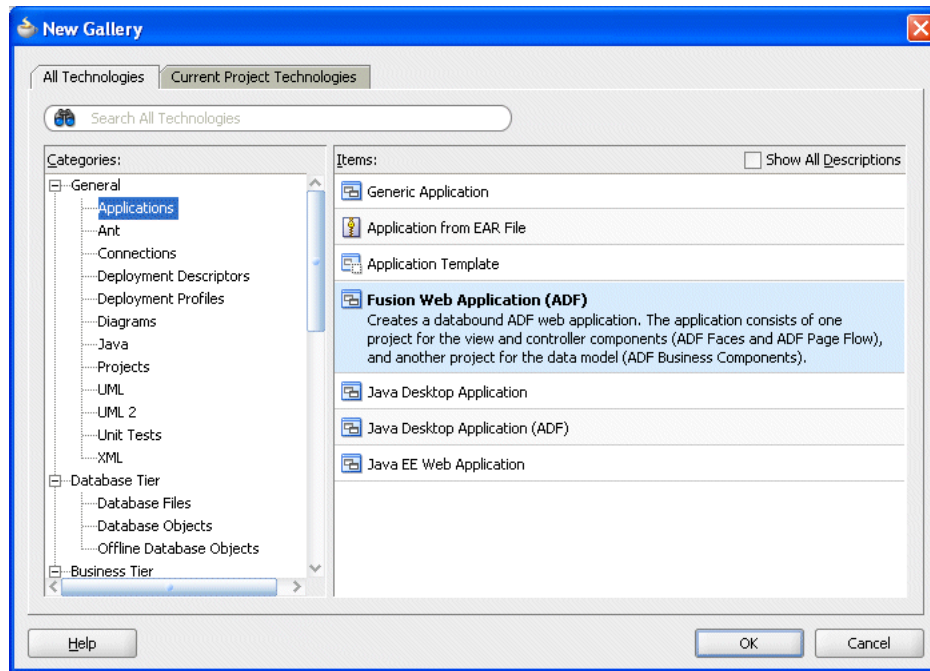
### 2.2 How to Configure the Environment for ADF Mobile Application Development

ADF Mobile application development for desktop browsers differs only from ADF Web application development for desktop browsers in the creation of the Web project. For more information, refer to [Section 2.3, "How to Create an ADF Mobile Application"](#).

To configure the environment, first create an ADF Mobile application:

1. Click **File** and then **New**. The *New Gallery* appears, with *General* selected in the *Categories* pane and *Application* selected in the *Items* pane (Figure 2–1).

**Figure 2–1** The New Gallery



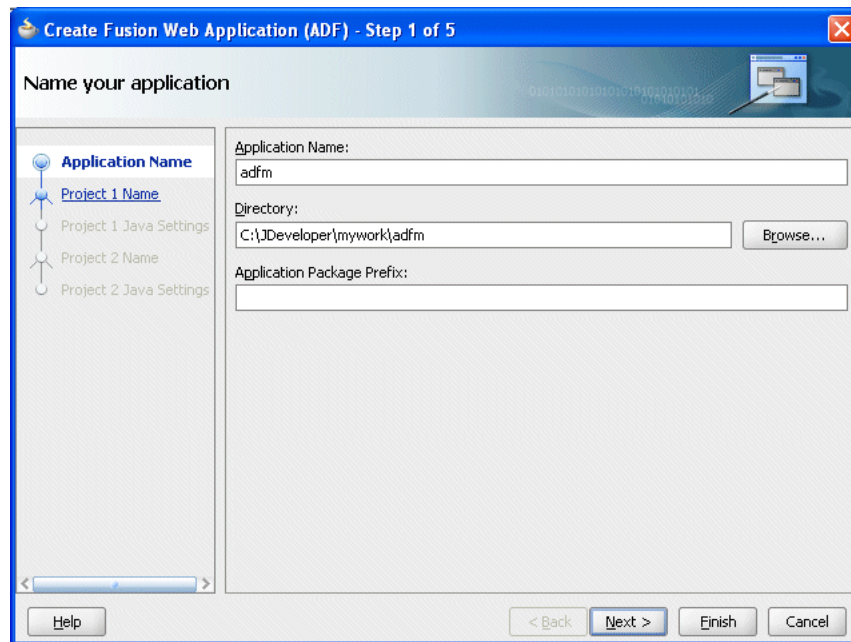
2. Select **Applications** in the *Categories* pane and then **Fusion Web Application** in the *Items* pane.

---

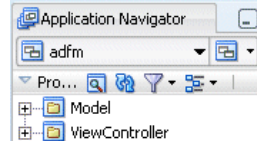
**Note:** **Fusion Web Application** must be selected as the application template.

---

3. Click **OK**. The Create Fusion Web Application (ADF) wizard opens to its first page, *Step 1: Name your application*. (Figure 2–2).
4. Complete this page by entering a name and, if needed, a location for the application in the *Directory* field.

**Figure 2–2 The Name your application Page**

5. Click **Finish**. The application's Model and its generated Model-View Controller projects appear in the Application Navigator (Figure 2–3).

**Figure 2–3 The Fusion Web Application and its Projects in the Application Navigator**


---

**Note:** Define the business logic the Model project, but do not use the generated Model-View Controller project. Instead, create a mobile Model-View Controller project as described in the following steps.

---

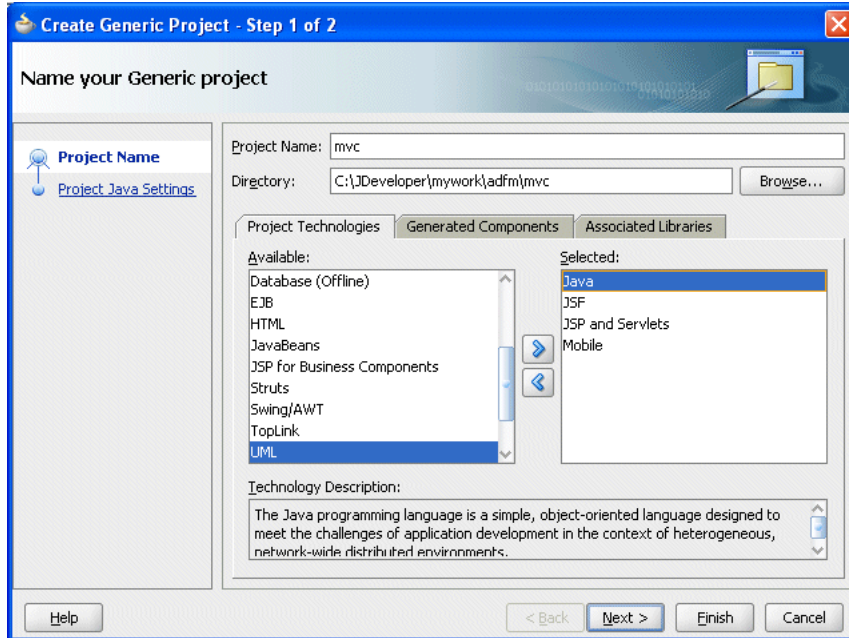
6. Click **File** and then **New**. The *New Gallery* appears.
7. Select **Projects** in the *Categories* pane and then **Generic Project** in the *Items* pane.
8. Click **OK**. The Create Generic Project wizard appears.
9. Complete the wizard by first entering a name for the project. For example, enter ViewController-Mobile.

**Tip:** To make entering a URL on mobile device easier, enter short, lower-case names for both the application and the project.

10. Select the ADF Mobile technology scope for the project by moving **Mobile** from the *Available* pane to the *Selected* pane. As depicted in Figure 2–4, the following technologies appear in the *Selected* pane:
  - Java

- JSF (JavaServer Faces)
- JSP and Servlets
- Mobile

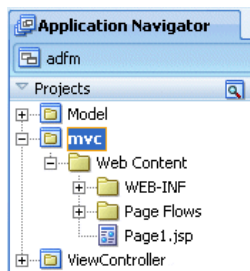
**Figure 2-4** *Selecting the Mobile Technology Scope*



Because you added the *Mobile* technology scope, the Apache My Faces Trinidad library is automatically loaded to the workspace and the *Trinidad* component palette will be loaded when you create mobile JSF pages (Figure 2-9).

11. Click **Next**. The *Configure Java settings for the project* page appears.
12. Click **Finish** to complete the project. The mobile Model-View Controller project (*mvc*) appears in the Application Navigator (Figure 2-5) within the Fusion Web application.

**Figure 2-5** *The Mobile ADF Model-View Controller Project in the Application Navigator*

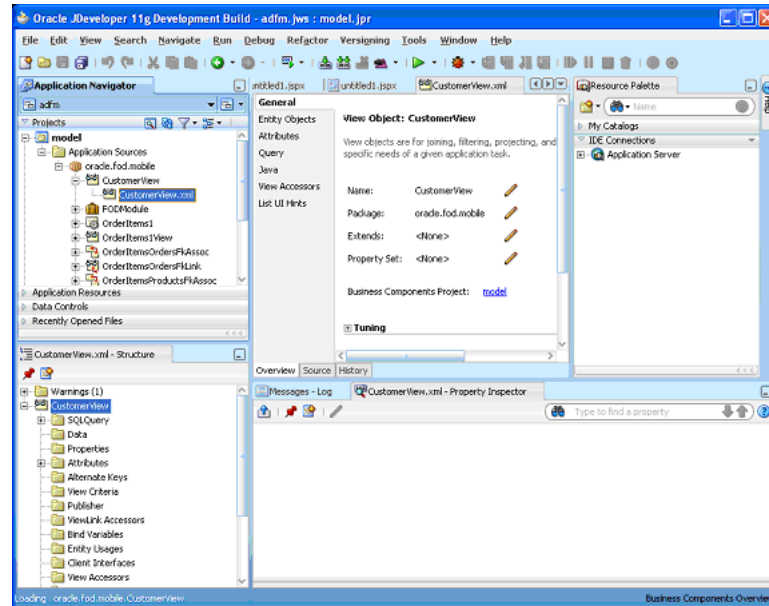


## 2.3 How to Create an ADF Mobile Application

ADF Mobile application development for desktop browsers is nearly identical to ADF Web application development for desktop browsers; the two only differ in how you create the Web project. For ADF Mobile applications, you develop an ADF Mobile application by creating Web pages in the Web project. Otherwise, you develop an ADF

Mobile application the same way that you develop an ADF Web application for a desktop browser. Typically, you create a Web project within the application to implement a user interface and ADF business components or a TopLink project to implement a business layer. [Figure 2–6](#) illustrates an ADF Mobile application (*adfm*) that contains *model*, a business components project.

**Figure 2–6** Creating a Business Components Project within an ADF Mobile Application



### 2.3.1 Creating a Mobile JSF Page

To create a mobile JSF page:

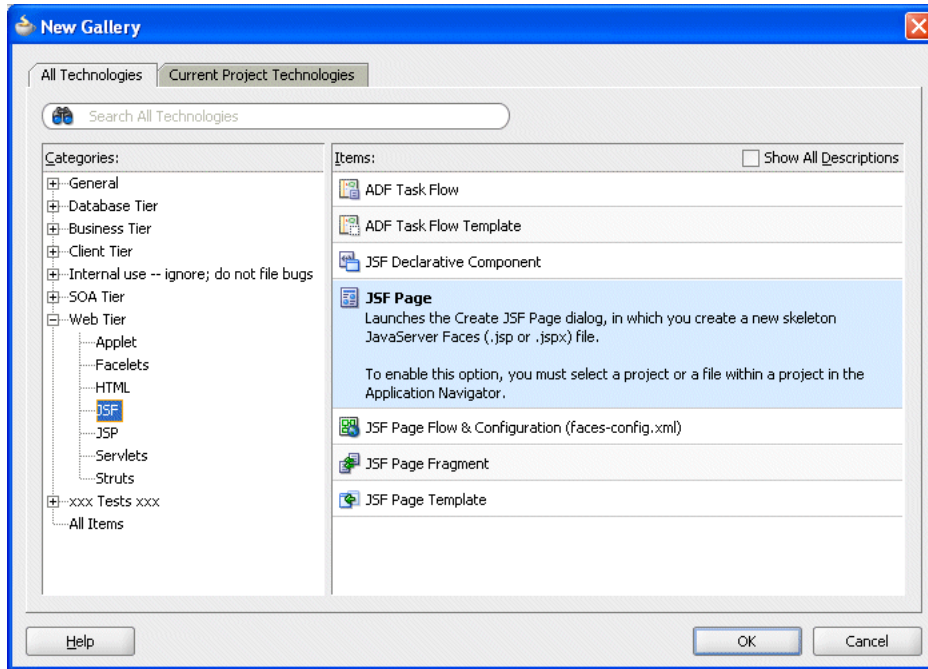
1. Click **File** and then **New**. The *New Gallery* appears.
2. Select **JSF** from the *Web Tier* node in the *Categories* pane.

---

**Note:** **Project Technologies** (the default) must be selected from the *Filter By* list.

---

**Figure 2–7 The New Gallery for JSF Pages**



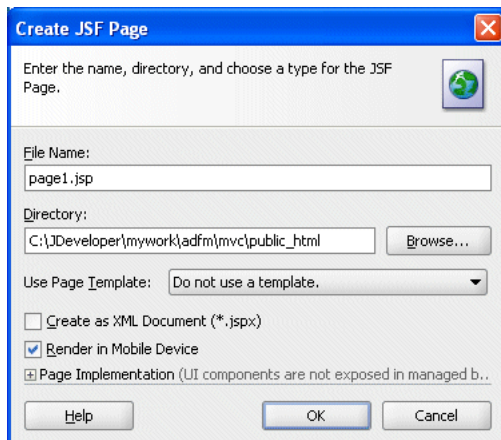
3. Select **JSF Page** in the *Items* pane and then click **OK**. The *Create JSF Page* dialog appears (Figure 2–8). A page designer for a mobile JSF page appears.

---

**Note:** Because you added the Mobile technology scope for the application, the **Render in Mobile Device** option is selected by default.

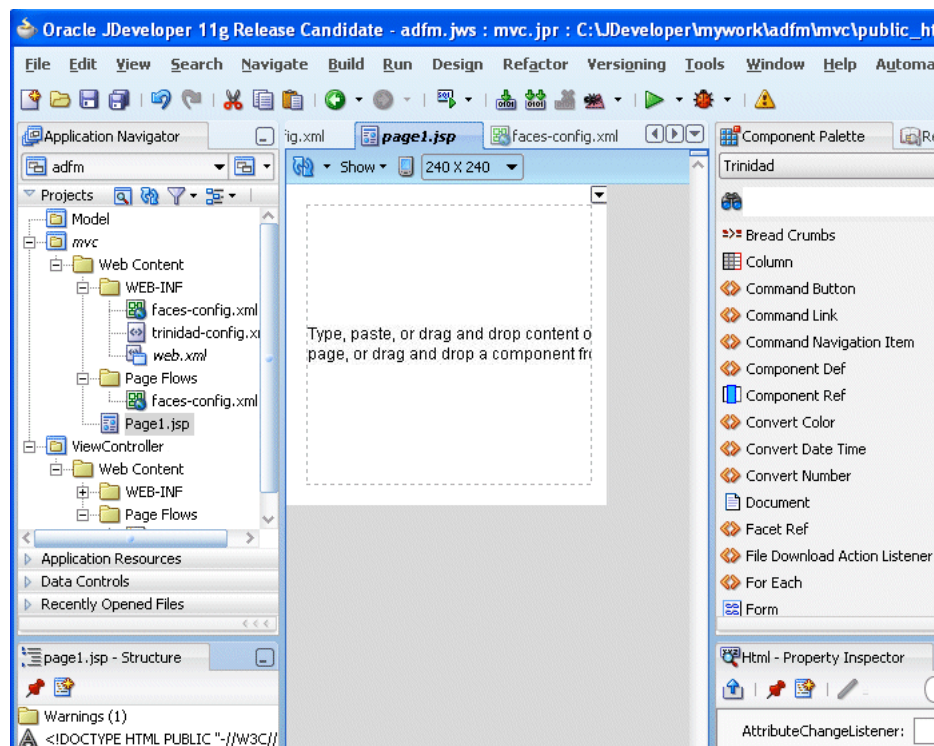
---

**Figure 2–8 The Create JSF Page Dialog**



4. Select the *Trinidad* component palette and then create the page using the Apache My Faces Trinidad components. You can create the page in the same manner as a desktop ADF Web page.

Figure 2–9 Using the Trinidad Component Palette



## 2.4 Testing the Application

You can test an ADF Mobile application on a mobile device, a mobile device emulator, or a desktop browser. Testing on an actual mobile device or mobile device emulator provides more accurate results than does testing on a desktop browser.

Using a desktop browser to test an ADF Mobile application produces only approximate results. Desktop browsers provide a fairly uniform testing environment: Web pages appear and behave similarly and business logic executes identically in any type of browser.

Testing an application on an actual mobile device, however, produces more accurate results, because the capabilities of mobile browsers may cause controls to behave differently than they do on a desktop browser. In addition, mobile browsers are usually smaller than desktop browsers. They also render pages differently than desktop browsers because Web servers optimize the look and feel by generating pages that are specific to the mobile browser.

Testing application on mobile devices has limitations as well, in that you may not have access to all of the devices that you want to test. Further, development environments behind firewalls can interfere with testing on mobile devices that access the Internet. In such cases, mobile device emulators provide an alternative testing method. For example, to test applications on BlackBerry or Windows Mobile emulators (Figure 2–10 and Figure 2–12, respectively), first download device emulators from the RIM developer site (<http://na.blackberry.com>) and the Microsoft developer site (<http://www.microsoft.com>). Before you test applications on the emulator, you first configure the emulator and connect it to the Web server. Information on downloading and configuring simulators for ADF Mobile is available at the Oracle Technology Network at

[http://www.oracle.com/technology/tech/wireless/ADFMobile\\_Emulator\\_howtos.html](http://www.oracle.com/technology/tech/wireless/ADFMobile_Emulator_howtos.html).

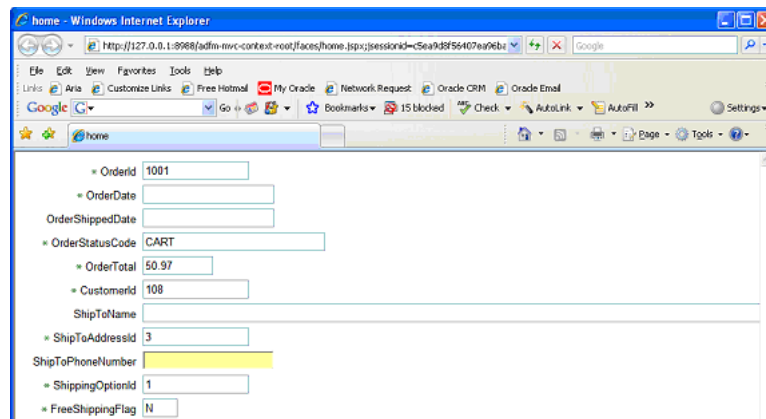
**Figure 2–10** Testing an ADF Mobile Application on a BlackBerry Emulator



## 2.4.1 How to Test ADF Mobile Applications

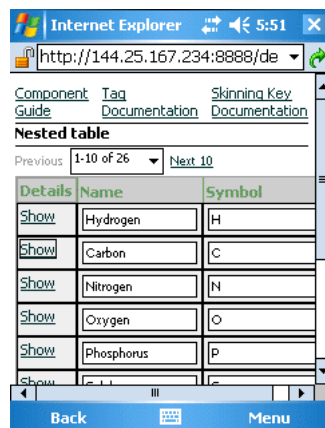
After you test an application on a desktop browser, you can then test it on an emulator. You can use the URL displayed in the desktop browser, but you must replace the local IP address with the IP address of the server that hosts the emulator. For example, to test an application using a Windows Mobile 6 emulator, change the address from the local IP used by the desktop to that of the server hosting the Windows Mobile 6 emulator.



**Figure 2–11 Testing an ADF Mobile Application on a Desktop Browser**

For example, in [Figure 2–11](#), the desktop browser displays the IP address of the local host, *127.0.0.1*. In [Figure 2–12](#), the IP address has been changed to that of the server hosting the Windows Mobile 6 emulator (*144.25.167.234*).

**Tip:** To obtain the network IP address, use the `ipconfig` command interface on Windows systems and the `ifconfig` command on Linux/UNIX systems.

**Figure 2–12 Testing an ADF Mobile Application on a Windows Mobile Emulator**

In addition, you must remove the session specification that follows the page name. The page name is typically appended with either `.jspx` or `.jsp`. In [Figure 2–11](#), the page name, *home*, is appended with `.jspx`.

In general, you debug an application by repeating cycles of code and then by testing the application. When you test an application that has been modified, you must do one or both of the following:

- Refresh the page.
- Clear the browser's cache.

You do not need to re-enter the URL, as it does not change as long as you develop the same application.

## 2.4.2 Browser Settings

This section describes the browser setting requirements for Windows Mobile and BlackBerry browsers.

### **Microsoft Windows Mobile 5 and 6, Microsoft Pocket Internet Explorer**

For optimal viewing, select the *Fit to Screen* view (accessed by selecting **Menu, View** and then **Fit to Screen**). Do not select the *One Column* view, as it causes layout problems.

### **RIM BlackBerry Browser 4.x**

ADF Mobile only works if JavaScript support is enabled. To ensure that JavaScript support is enabled:

1. Select **Options** and then **Browser Configuration**.
2. Ensure that the following tables are selected:
  - Support JavaScript
  - Allow JavaScript Popup
  - Support HTML Tables

---

---

## Component Support

This chapter describes the ADF Components that are supported by ADF Mobile. This chapter includes the following topics:

- [Section 3.1, "Introduction to Apache My Faces Trinidad Components"](#)
- [Section 3.2, "Input Components"](#)
- [Section 3.3, "Output Components"](#)
- [Section 3.4, "Layout Components"](#)
- [Section 3.5, "Navigation Components"](#)
- [Section 3.6, "Tables and Trees"](#)
- [Section 3.7, "Unsupported Components and Attributes"](#)

### 3.1 Introduction to Apache My Faces Trinidad Components

ADF Mobile supports more than 60 of Apache My Faces Trinidad components, enabling you to build applications with a rich component set that renders appropriately to the screens of mobile devices. For more information, refer to the Apache My Faces Trinidad site (<http://myfaces.apache.org/trinidad/>).

#### 3.1.1 Supported Features

ADF Mobile supports the following renderer-specific features for the supported browsers:

- [Partial Page Rendering for Microsoft Pocket Internet Explorer](#)
- [Dialogs](#)

#### 3.1.2 Partial Page Rendering for Microsoft Pocket Internet Explorer

The high latency and low bandwidth of networks in mobile environments diminish application responsiveness for mobile users. Screens refresh slowly, lessening the mobile user experience. ADF Mobile's support of Partial Page Rendering (PPR) compensates for the negative impact that slow connections have on screen updates by minimizing the amount of data requested from the server; using PPR, mobile device screen updates do not require a full refresh. ADF Mobile supports PPR for Microsoft Pocket Internet Explorer. For this browser, PPR support is limited to the `showDetail` and `outputText` components as partial targets (that is, these components specify a `partialTriggers` attribute) and for the `table` component for internal changes, such as sorting.

---

---

**Note:** BlackBerry Browser 4.x does not support PPR. Specifying the `autosubmit` attribute on certain form components results in the submission of the page once the user exits the field. A full, not partial, refresh of the page then follows.

---

---

### 3.1.3 Dialogs

ADF Mobile supports dialogs, pages used by applications to obtain user input. Because mobile browsers cannot open a new window that contains a dialog (a pop-up window), dialogs appear as new pages within the main browser window after automatically preserving the state of the current page.

### 3.1.4 Rendering Specific to the BlackBerry Browser 4.x

On the BlackBerry Browser 4.x, the bullets in a list sublevel (such as those from a `tr:panelList`) appear large and are not indented. The BlackBerry browser's table handling may affect complex layouts; the BlackBerry Browser does not allow horizontal scrolling. Instead, it wraps a table row onto multiple display lines, which may disturb the layout.

## 3.2 Input Components

This section describes the ADF Faces core components that support user input. This section includes the following topics:

- [Section 3.2.1, "Creating Input Text Fields"](#)
- [Section 3.2.2, "Creating Lists"](#)

### 3.2.1 Creating Input Text Fields

You can create input fields using the following components:

- `tr:inputColor`

---

---

**Note:** Mobile browsers do not support an inline `chooseColor` or a `color` dialog.

---

---

- `tr:inputDate`
- `tr:inputHidden`
- `tr:inputText`

### 3.2.2 Creating Lists

You can create lists using the following components:

- `tr:panelChoice`
- `tr:panelList`
- `tr:selectBooleanCheckBox`
- `tr:selectBooleanRadio`
- `tr:selectItem`

---

---

**Note:** Mobile browsers do not support the `disabled` attribute.

---

---

- `tr:selectManyCheckBox`
- `tr:selectManyListBox`
- `tr:selectOneChoice`
- `tr:selectOneListBox`
- `tr:selectOneRadio`
- `tr:resetButton`

## 3.3 Output Components

This section describes the Apache MyFacesTrinidad core components that support output on mobile device applications. This section includes the following topics:

- [Section 3.3.1, "Displaying Text"](#)
- [Section 3.3.2, "Displaying Images"](#)
- [Section 3.3.3, "Showing \(or Hiding\) Components"](#)

### 3.3.1 Displaying Text

The following components enable you to display text:

- `tr:iterator`
- `tr:message`
- `tr:messages`

---

---

**Note:** When using the `tr:message` and `tr:messages` components, the component-specific messages do not pop up as they do in a desktop browser. Instead, they display in the region where the message component is placed on the Web page.

---

---

- `tr:outputDocument`
- `tr:outputForwarded`
- `tr:outputLabel`
- `tr:outputText`

### 3.3.2 Displaying Images

The following components enable you to display images:

- `tr:icon`
- `tr:image`
- `tr:panelTip`

### 3.3.3 Showing (or Hiding) Components

The following components enable showing or hiding items:

- `tr:panelAccordion`

---

---

**Note:** Mobile browsers only support a full-page update; they do not support the `partialTriggers` attribute of the `tr:panelAccordion`.

---

---

- `tr:panelTabbed`
- `tr:showDetail`

---

---

**Note:** The disclosure arrow does not display; instead `[+]` and `[-]` display.

---

---

- `tr:showDetailHeader`

---

---

**Note:** The disclosure arrow does not appear on mobile browsers.

---

---

- `tr:showDetailItem`

---

---

**Note:** The disclosure arrow does not appear on mobile browsers and `flex` is not supported.

---

---

## 3.4 Layout Components

The following section describes the layout components supported by ADF Mobile:

- [Section 3.4.1, "Using Page Templates"](#)
- [Section 3.4.2, "Managing the Page"](#)
- [Section 3.4.3, "Laying Out Sections of the Page"](#)
- [Section 3.4.4, "Inserting Spaces"](#)

### 3.4.1 Using Page Templates

ADF Mobile supports the `tr:pageTemplate` component.

### 3.4.2 Managing the Page

The following components enable you to manage the page:

- `tr:document`
- `tr:form`

---

---

**Note:** Mobile browsers do not support the `defaultCommand` attribute.

---

---

- `tr:page`

### 3.4.3 Laying Out Sections of the Page

The following ADF Faces core tags support page layout for mobile device applications:

- `tr:group`
- `tr:panelBorderLayout`

---

---

**Note:** Mobile browsers do not support the following facets:

left  
right  
start  
end  
innerLeft  
innerRight  
innerStart  
innerEnd

---

---

- `tr:panelBox`
- `tr:panelFormLayout`
- `tr:panelGroupLayout`
- `tr:panelHeader`
- `tr:panelHorizontalLayout`
- `tr:panelLabelAndMessage`
- `tr:panelPage`
- `tr:panelPageHeader`
- `tr:panelRadio`

### 3.4.4 Inserting Spaces

The following components control the space allocation on pages:

- `tr:separator`
- `tr:spacer`
- `tr:subform`

## 3.5 Navigation Components

The following sections describe the components that enable users to navigate to other pages of the application or to external locations.

- [Section 3.5.1, "Creating Buttons"](#)
- [Section 3.5.2, "Creating Links"](#)
- [Section 3.5.3, "Navigation Components"](#)

### 3.5.1 Creating Buttons

ADF Mobile supports the following button types:

- `tr:commandButton`

---

---

**Note:** Because the `text` attribute cannot display if the `icon` attribute has been set, buttons on mobile devices can have either text or an image, but not both. If you set the `disabled` attribute to `true`, then the `commandButton` component with an `icon` attribute renders as a static image with no links.

---

---

- `tr:goButton`

### 3.5.2 Creating Links

ADF Mobile supports the following components for creating hyper-links:

- `tr:commandLink`
- `tr:goLink`

### 3.5.3 Navigation Components

ADF mobile supports the following navigation components:

- `tr:breadcrumbs`
- `tr:commandNavigationItem`

---

---

**Note:** `tr:commandNavigationItem` -- The component does not render when you set the `disabled` attribute to `true` for the following components:

```
tr:selectOneListBox  
tr:selectOneChoice  
tr:processChoiceBar  
tr:navigationPane with hint, "choice"  
tr:selectRangeChoiceBar
```

---

---

- `tr:navigationPane`
- `tr:train`

---

---

**Note:** Appears as *x of y* instead of listing each item.

---

---

## 3.6 Tables and Trees

This section includes the following topics:

- [Section 3.6.1, "Creating Tables"](#)
- [Section 3.6.2, "Creating Trees"](#)



### 3.6.1 Creating Tables

ADF Mobile supports tables comprised of the following components:

- `tr:table`
- `tr:column`

### 3.6.2 Creating Trees

ADF mobile supports the `tr:tree` component.

## 3.7 Unsupported Components and Attributes

This section lists the components and attributes that ADF Mobile 11g does not support and includes the following sections:

- [Section 3.7.1, "Unsupported Components"](#)
- [Section 3.7.2, "Unsupported Attributes"](#)

### 3.7.1 Unsupported Components

ADF Mobile 11g does not support the following:

- `tr:chart`
- `tr:chooseColor`
- `tr:chooseDate`
- `tr:inputFile`
- `tr:inputListOFVariables`
- `tr:inputNumberSpinbox`
- `tr:legend`
- `tr:media`
- `tr:navigationTree`
- `tr:panelButtonBar`
- `tr:panelPopup`
- `tr:panelSideBar`
- `tr:poll`
- `tr:progressIndicator`
- `tr:selectManyShuttle`
- `tr:selectOrderShuttle`
- `tr:singleStepButtonBar`
- `tr:statusIndicator`
- `tr:switcher`
- `tr:treeTable`

### 3.7.2 Unsupported Attributes

ADF Mobile 11g does not support the following component attributes on any component.

- `accessKey`
- `shortDesc` (tooltip)
- `onXXX` script handlers