Oracle® Fusion Applications

Extensibility Guide for Business Analysts

11*g* Release 7 (11.1.7)

E16691-12

September 2013

Documentation for business analysts and administrators that describes how to use runtime tools to customize and extend the standard functionality provided by Oracle Fusion Applications.

Oracle Fusion Applications Extensibility Guide for Business Analysts, 11*g* Release 7 (11.1.7)

E16691-12

Primary Author: Chris Kutler (lead), Sarah Bernau, Shelly Butcher, Ralph Gordon, Peter Jew, Mark Kennedy, Steven Leslie, Landon Ott, Leslie Studdard

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Part II Runtime Customizations and Extensions

3 Customizing Existing Pages

4 Using Flexfields for Custom Attributes

5 Customizing Menus

6 Customizing and Extending BPMN Processes

7 Customizing Reports and Analytics

Part III Appendixes

A Troubleshooting Customizations

Preface

Welcome to *Oracle Fusion Applications Extensibility Guide for Business Analysts.*

Audience

This document is intended for business analysts and administrators who want to use runtime tools to customize and extend the standard functionality provided by Oracle Fusion Applications. Business analysts and administrators should have a basic understanding of Oracle Fusion Applications. This book gives an overview of the customization and extension tasks and provides references to the books that contain more detailed documentation.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Documents

You can also find information about Oracle Fusion Middleware and extending and customizing Oracle Fusion Applications in the following documents:

- *Oracle Database Security Guide*
- *Oracle Fusion Applications Administrator's Guide*
- *Oracle Fusion Applications Administrator's Troubleshooting Guide*
- *Oracle Fusion Applications Administrator and Implementor Roadmap*
- *Oracle Fusion Applications Common Implementation Guide*
- *Oracle Fusion Applications Concepts Guide*
- *Oracle Fusion Applications CRM Extensibility Guide*

- *Oracle Fusion Applications Enterprise Deployment Guide for Customer Relationship Management*
- *Oracle Fusion Applications Extensibility Guide for Developers*
- *Oracle Fusion Applications Developer's Guide*
- *Oracle Fusion Functional Setup Manager User's Guide*
- *Oracle Fusion Functional Setup Manager Developer's Guide*
- *Oracle Fusion Applications Installation Guide*
- *Oracle Fusion Applications Post-Installation Guide*
- *Oracle Fusion Applications Master Glossary*
- *Oracle Fusion Applications Patching Guide*
- *Oracle Fusion Applications Security Guide*
- *Oracle Fusion Applications Security Hardening Guide*
- *Oracle Fusion Middleware Administrator's Guide*
- *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite*
- *Oracle Fusion Middleware Application Security Guide*
- *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management*
- *Oracle Fusion Middleware Data Modeling Guide for Oracle Business Intelligence Publisher (Oracle Fusion Applications Edition)*
- *Oracle Fusion Middleware Developer's Guide for Oracle Enterprise Scheduler*
- *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter Portal*
- *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*
- *Oracle Fusion Middleware Error Messages Reference*
- *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*
- *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition)*
- *Oracle Fusion Middleware XML Schema Reference for Oracle Business Intelligence Enterprise Edition*
- *Oracle Fusion Middleware Modeling and Implementation Guide for Oracle Business Process Management*
- *Oracle Fusion Middleware Oracle Authorization Policy Manager Administrator's Guide (Oracle Fusion Applications Edition)*
- *Oracle Fusion Middleware Patching Guide*
- *Oracle Fusion Middleware Developer's Guide for Oracle Business Intelligence Publisher (Oracle Fusion Applications Edition)*
- *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*
- *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition)*

- *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*
- *Oracle Fusion Middleware Reference Guide for Oracle Business Intelligence Applications*
- *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*
- *Oracle Fusion Middleware User's Guide for Oracle Business Rules*
- *Oracle Fusion Middleware User Guide for Oracle Enterprise Repository*
- *Oracle Fusion Middleware User's Guide for Oracle Identity Manager*
- *Oracle Fusion Middleware User's Guide for Oracle WebCenter Portal: Spaces*
- *Oracle Fusion Middleware User's Guide for Technology Adapters*
- *Oracle Fusion Middleware Web User Interface Developer's Guide for Oracle Application Development Framework*
- *Oracle Fusion Middleware Oracle WebLogic Scripting Tool*
- *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	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.

What's New in This Guide

The following topics introduce the new and changed customization and extensibility features of Oracle Fusion Applications and other significant changes that are described in this guide, and provides pointers to additional information.

New and Changed Features for 11*g* Release 7 (11.1.7)

Oracle Fusion Applications 11*g* Release 7 (11.1.7) includes the following new and changed customization and extensibility features for this document.

- Customization Set Migration tool, which enables the movement of customizations and extensions from test environments to production environments. See Section 2.3.1, "Using Customization Set Migration to Move Customizations."
- Flexfield enhancements:
  - **Highlight Flexfields** option on the **Administration** menu, which enables you to see information about the descriptive flexfields on a page and to access the Quick Create dialog to add context values, global segments, and context-sensitive segments. See Section 4.2, "Finding the Flexfields on a Page."
  - **Show Value Description** option on the Manage Contexts page in the Manage Extensible Flexfields task, which enables the display of a value description next to an extensible flexfield segment. See Task: Configure the Context in Section 4.5.2, "Configuring Extensible Flexfields."
  - **Create Value Sets** button in the Manage Descriptive Flexfields and Manage Extensible Flexfields tasks, which enables you to create a value set at the time that you create a segment. See Section 4.5, "Configuring Flexfields."
  - Ability to add an additional attribute to an independent, dependent, or subset value set, which can be used to capture additional information about each valid value, such as its purpose. See Task: Add Attributes to a Value Set in Section 4.4, "Creating Custom Value Sets."
  - Ability to use a segment value from a different context in the WHERE clause for a table value set. See Task: Create a Table Value Set in Section 4.4, "Creating Custom Value Sets."
  - Ability to use SQL statements to set the initial value of an extensible flexfield segment. You can also set the initial value to the value of another segment in the same extensible flexfield. See Task: Define Initial Values for Extensible Flexfield Custom Attributes in Section 4.3.2, "Planning Extensible Flexfields."
  - Non-deployment of extensible flexfield segments if their category is not enabled. See Section 4.7, "Deploying Flexfield Configurations."

  - Ability to enable flexfields for Oracle Social Network. See Section 4.8, "Integrating Custom Attributes."

- Home page menu customization. See Section 5.1.2, "About Home Page Menu Configuration."
- Oracle Business Intelligence Publisher customize feature, which simplifies adding custom layouts to Oracle BI Publisher reports. See Section 7.2.1.4, "Using the Customize Feature."

Other Significant Changes in this Document for 11*g* Release 7 (11.1.7)

For 11*g* Release 7 (11.1.7), this guide has been updated in several ways. Following are the sections that have been added or changed.

- Corrected information about what lookups are included in a customization set that is created by the Customization Set Migration dialog. Only standard lookups are migrated. See Section 2.3.1, "Using Customization Set Migration to Move Customizations."
- Added steps to Section 2.3.1, "Using Customization Set Migration to Move Customizations" for enabling attachments on runtime pages in the target environment after a migration.
- Moved information about Oracle Fusion CRM Application Composer to the *Oracle Fusion Applications CRM Extensibility Guide*.
- Moved information about using design time tools to customize and extend Oracle Fusion Applications to *Oracle Fusion Applications Extensibility Guide for Developers*.
- Revised sections throughout the book to note features and functionality that are not available in Oracle Cloud implementations.
- Revised tasks in Chapter 3, "Customizing Existing Pages" to account for Design view being the only mode available for personalization.
- Added instructions for setting search options for site-level search. See Task: Create a Site-Level Saved Search for All Users in Chapter 3, "Customizing Existing Pages."
- Added instructions for implementing role-based page customizations using Page Composer. See "Task: Customize a Component Based on Role" in Chapter 3, "Customizing Existing Pages."

New and Changed Features for 11*g* Release 6 (11.1.6)

Oracle Fusion Applications 11*g* Release 6 (11.1.6) includes the following new and changed customization and extensibility features for this document.

- Site-level saved searches, which allow you to create searches that can be used by all users. See Task: Create a Site-Level Saved Search for All Users in Chapter 3, "Customizing Existing Pages."
- Custom folder seeded under Shared Folders provides a standard location to maintain your customized reports. Default permission settings are enhanced to support user interaction with the new folder. See Section 7.2.1.4, "Using the Customize Feature."

Other Significant Changes in this Document for 11*g* Release 6 (11.1.6)

For 11*g* Release 6 (11.1.6), this guide has been updated in several ways. Following are the sections that have been added or changed.

- Revised report customization procedures:
  - Section 7.2.1.1, "What You Can Customize"
  - Section 7.2.1.4, "Using the Customize Feature"
  - Section 7.2.2, "Customizing Layouts"
- Revised analytics customization procedures:
  - Section 7.3.1.3, "Understanding Catalog Security and Allowed Customizations"
  - Section 7.3.2, "Customizing Analytics"

Part I

Introduction to Customizing and Extending Oracle Fusion Applications

Part I contains the following chapters:

1

Customizing and Extending Oracle Fusion Applications

This chapter provides an overview of how to customize and extend Oracle Fusion applications and introduces Page Composer, Oracle Fusion CRM Application Composer (Application Composer), Business Process Composer, and Oracle Business Intelligence Publisher (Oracle BI Publisher).

Within this guide, the term **customize** means to change a standard (existing) artifact. The term **extend** means to create a completely new artifact, such as a custom object.

This chapter includes the following sections:

- Section 1.1, "Understanding Customizing and Extending Oracle Fusion Applications"
- Section 1.2, "Understanding Customization Layers"
- Section 1.3, "Understanding the Runtime Tools"

Note: For Oracle Cloud implementations, check the Oracle Cloud documentation to determine which Oracle Fusion Applications versions are supported by that Oracle Cloud implementation.

1.1 Understanding Customizing and Extending Oracle Fusion Applications

While Oracle Fusion applications provide robust out-of-the-box functionality, there may be areas of an application that you must change to meet your business needs. This book guides you through the process of using runtime tools to extend and customize Oracle Fusion applications.

Oracle Fusion applications are built using a common data model. Because of this commonality, when you make a customization in one area, that customization is available to all objects in the application. For example, if you add an attribute to an object, you can easily add that attribute to the web-based view page, to an associated mobile page, and to any associated reports.

You can make the customizations in a sandbox so that the changes are isolated. After validating the changes, you can publish them into a full test environment. For more information about sandboxes, see Section 2.2, "Using the Sandbox Manager."

For customizations and extensions, there are three basic scenarios:

- Personalization

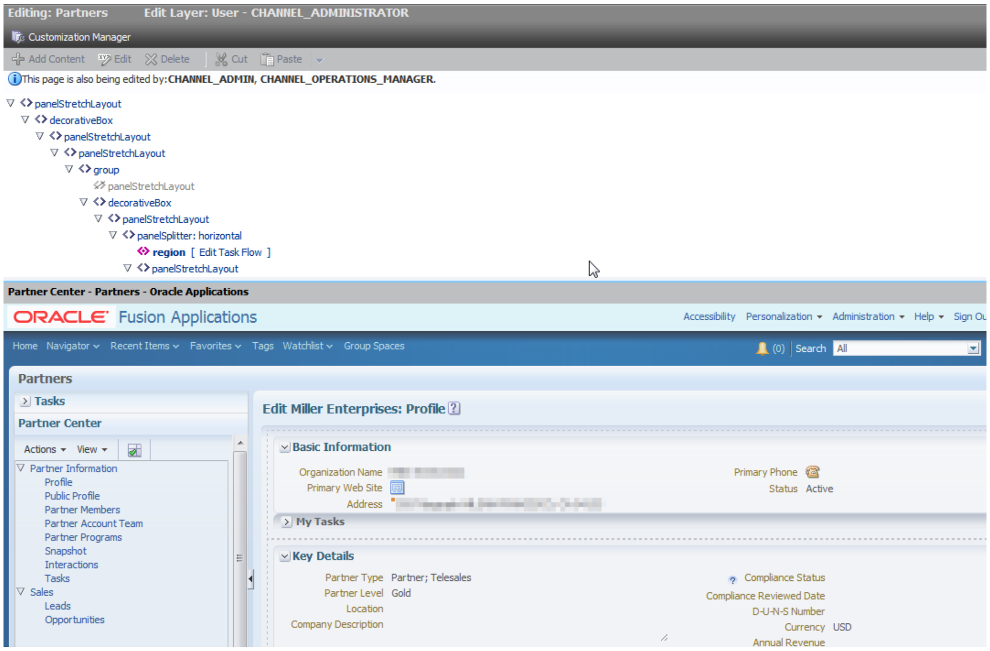
- Runtime customizations and extensions
- Design time customizations and extensions

1.1.1 Personalization

The term personalization refers to the changes that every end user of the Oracle Fusion Applications product suite makes to certain artifacts in the user interface (UI) at runtime. These changes remain for that user each time that user logs in to the application. Personalization includes changes based on user behavior (such as changing the width of a column in a table), changes the user elects to save, such as search parameters, or composer-based personalizations, where an end user can redesign aspects of a page.

For composer-based personalizations, Oracle Fusion Applications includes Page Composer, which allows end users to change certain UI pages to suit their needs. For example, they can rearrange certain objects on a page, add and remove designated content, and save queries. Figure 1–1 shows the Partner Profile page in Page Composer. An end user can add other content to this page, or change the order of the current content.

Figure 1–1 End Users Can Personalize UIs with Page Composer



This page is also being edited by:CHANNEL_ADMIN, CHANNEL_OPERATIONS_MANAGER.

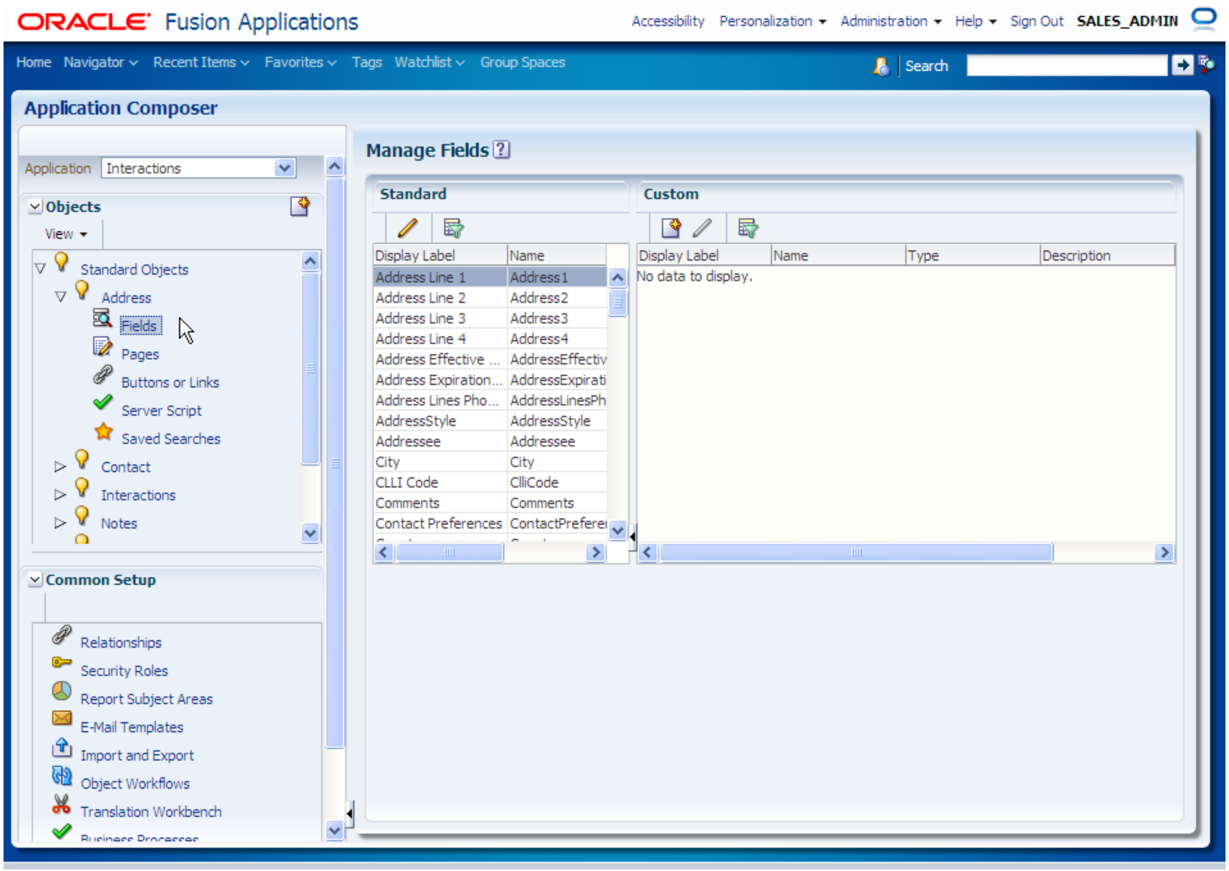
Organization Name [redacted]
Primary Web Site
Address [redacted]
Primary Phone ☎
Status Active

Partner Type Partner; Telesales
Partner Level Gold
Location
Company Description
Compliance Status
Compliance Reviewed Date
D-U-N-S Number
Currency USD
Annual Revenue

1.1.2 Runtime Customizations and Extensions

Runtime customizations and extensions include those that a business analyst makes to an Oracle Fusion application at runtime using browser-based composers. These customizations and extensions are visible and usable by all or by a subset of Oracle Fusion Applications users. The types of **runtime customizations and extensions** range from changing the look and feel of a page, to customizing standard objects, adding a new object and associated pages and application functionality, changing workflows, defining security for new objects, and customizing reports. Figure 1–2 shows how you can customize the fields on a standard object using Application Composer, which is a runtime tool used to customize and extend certain Oracle Fusion Customer Relationship Management (Oracle Fusion CRM) applications.

Figure 1–2 Application Composer Allows You to Customize Objects at Runtime



Standard

Display Label	Name
Address Line 1	Address1
Address Line 2	Address2
Address Line 3	Address3
Address Line 4	Address4
Address Effective ...	AddressEffectiv
Address Expiration...	AddressExpirati
Address Lines Pho...	AddressLinesPh
AddressStyle	AddressStyle
Addressee	Addressee
City	City
CLLI Code	ClliCode
Comments	Comments
Contact Preferences	ContactPreferei

Custom

Display Label	Name	Type	Description
No data to display.			

Customizing and extending Oracle Fusion applications using runtime tools, with the exception of Application Composer, is covered in this book. For information about Application Composer, see the "Application Composer: Introduction" chapter in the *Oracle Fusion Applications CRM Extensibility Guide*.

1.1.3 Design Time Customizations and Extensions

Note: This functionality is not available in Oracle Cloud implementations.

Design time customizations and extensions include more complex changes, such as creating a **SOA composite application** or creating a new batch job, and they require deployment into the runtime environment. These **design time customizations and extensions** are most often done by Java developers using Oracle JDeveloper (a comprehensive integrated development environment), or they may be done in other tools, such as Oracle SOA Composer. The customizations are then uploaded or deployed to a running instance of Oracle Fusion Applications.

Design time customizations and extensions are covered in the *Oracle Fusion Applications Extensibility Guide for Developers*.

1.2 Understanding Customization Layers

Oracle Fusion applications contain built-in customization layers that allow you to make customizations that affect only certain instances of an application. For example, the Sales application has a layer for job **role**. When you customize an artifact, you can choose to make that customization available only to users of a specific job role, for example, a sales representative.

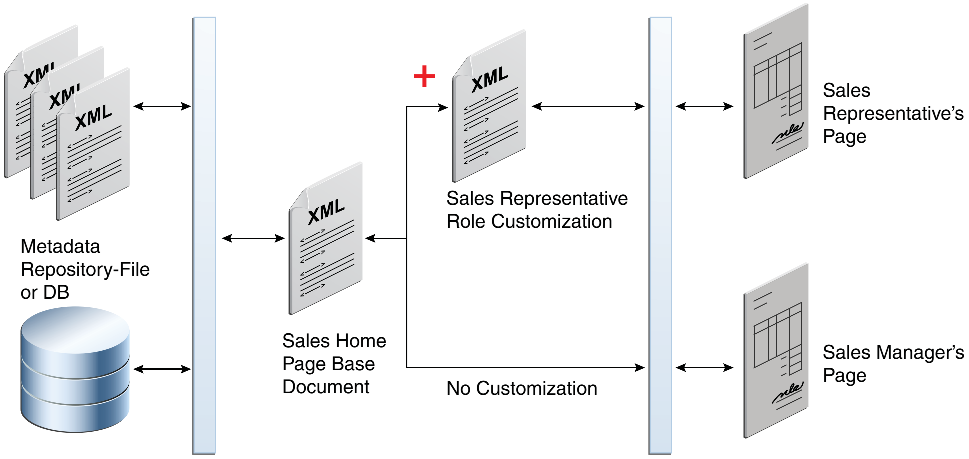
Customizations you make are not saved to the base standard artifact. Instead, they are saved to an Extensible Markup Language (XML) file that is stored in an Oracle Metadata Services (MDS) repository. This XML file acts like a list of instructions that determines how the artifact looks or behaves in the application, based on the layer that is controlling the current context. The customization engine in MDS manages this process.

For example, say you want to customize the Sales home page by removing the Quick Create panel, but only for users with the Sales Representative role. Before you make your customization, you first select the layer in which to make your customization, in this case the role layer whose value is `Sales Representative`. When you make your customization by removing that pane from the page, an XML file is generated with the instructions to remove the pane, but only in the role layer, and only when the value is `Sales Representative`. The original page file remains untouched. The customization engine in MDS then stores the XML file in an MDS repository.

Now, whenever someone logs in to the application and requests an artifact, the customization engine in MDS checks the repository for XML files that match the requested artifact and the given context, and if there is a match, it layers the instructions on top of the base artifact. In this example, whenever the Sales home page is requested (the artifact) by someone who is assigned the role of Sales Representative (the context), before the page is rendered, the customization engine in MDS pulls the corresponding XML file from the repository, layers it on top of the standard Sales home page, and removes that pane. Whenever someone who is not a Sales Representative logs in (for example, someone with the role of Sales Manager), the XML file with your changes is not layered on top, and so the Quick Create panel is displayed.

Figure 1–3 shows how the customization XML file is applied to the base document and is visible only to a sales representative.

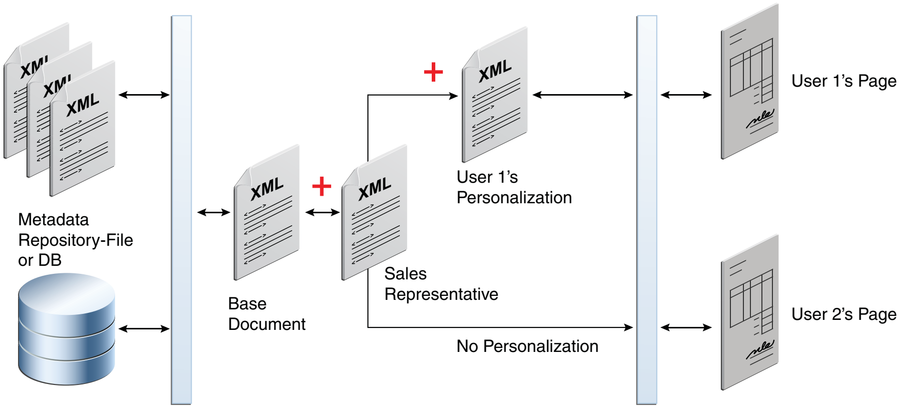
Figure 1–3 One Customization Layer Handled by the Customization Engine in MDS



All users of Oracle Fusion applications can personalize certain pages using the Personalization menu. Users can move elements around on a page, hide elements, and even add available elements to their page. When they do this personalization, the customization engine in MDS creates an XML file specific to that user.

For example, say User 1 (who has the role of Sales Representative) personalizes the Sales home page. There will then be an XML file stored in the repository, noting the changes that user made. When User 1 logs in, as in the previous example, the customization engine in MDS pulls the XML file with the sales representative customizations from the repository and layers it on top of the standard Sales home page. In addition, the engine pulls the XML file with the User 1personalizations, allowing the user to see the personalization changes along with the Sales Representative changes. When other Sales Representatives log in, they do not see the User 1 personalization changes, as shown in Figure 1–4.

Figure 1–4 Personalizations Are Also Handled by the Customization Engine in MDS



The exact customization layers available for an application depend on that application family (see the product-specific documentation from Oracle Enterprise Repository for Oracle Fusion Applications for details). However, all Oracle Fusion applications have the following customization layers:

- Global layer: When customizations are made in the **global layer**, they affect all users of the application. This layer's XML files are added for everyone, whenever the artifact is requested. Customizations made to ADF Business Components in JDeveloper must be made in the global layer.
- Site layer: Customizations made in the **site layer** affect users at a particular location.
- User layer: The **user layer** is where all personalizations are made. Users do not have to explicitly select this layer. It is automatically selected when you use the Personalization menu.

These layers are applied in a hierarchy, and the highest layer in that hierarchy in the current context is considered the **tip**. Within the default customization layers, the global layer is the base layer, and the user layer is the tip. If customizations are done to the same object, but in different layers, at runtime, the tip layer customizations take precedence. For example, if you customize the label for a field in the site layer using Page Composer and customize the same label in the global layer using JDeveloper, the site layer customization will be displayed at runtime.

Because customizations are saved in these XML files, when you patch or upgrade your Oracle Fusion applications, the base artifacts can be updated without touching your changes. The base artifact is replaced, and when the application is run after the patch or upgrade, the XML files are simply layered on top of the new version. You do not need to redo your customizations.

Before you create customizations, you must select the layer to which you want your customizations to be applied. Most of the tools that you use to create your customizations provide a dialog where you can pick the layer for your customizations.

1.3 Understanding the Runtime Tools

Oracle Fusion Applications provides several runtime tools to enable you to customize and extend Oracle Fusion applications. With these tools, you can perform the following tasks:

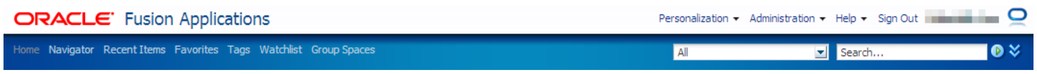
- Personalize and customize pages using Page Composer
- Customize pages using Application Composer
- Create and customize objects using Application Composer
- Create business process flows for custom objects
- Define security policies for custom objects
- Add custom attributes to a **business object**
- Customize reports and analytics

For a more detailed description of the workflow you must follow when customizing and extending Oracle Fusion applications, see Chapter 2, "Understanding the Customization Development Lifecycle."

1.3.1 Understanding Role-Based Access to Tools

The user interfaces in Oracle Fusion applications are controlled by role-based authentication, meaning that the information presented in the UI, and what the user can do in the UI, depends on the role assigned to the currently logged-in user. For example, if you are assigned a role with an administrative **privilege**, when you log in to Oracle Fusion Applications, you will see an Administration menu, as shown in Figure 1–5. This menu allows you to do things such as customize a page for all users, or manage customizations.

Figure 1–5 Oracle Fusion Applications Menu Bar

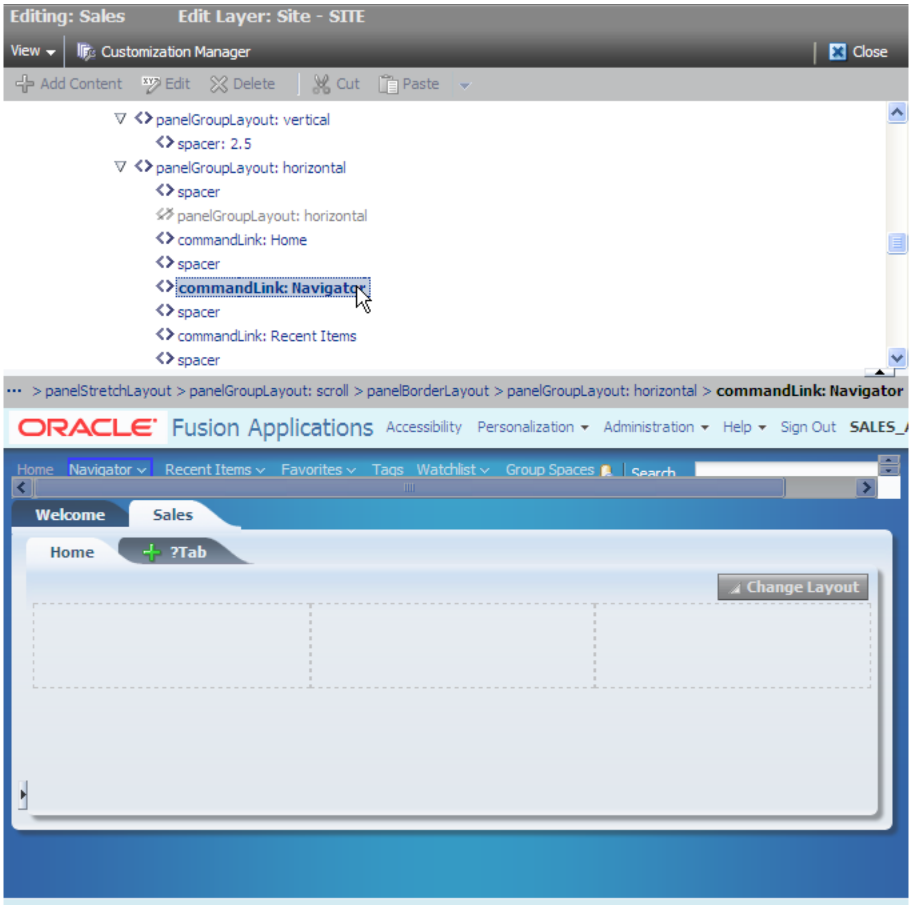


1.3.2 Personalizing and Customizing Pages Using Page Composer

Both personalization and customization use Page Composer to make changes to an application page. Using personalization, any user can drag and drop fields, rearrange regions, add approved external content, and save their favorite queries.

Using administration customization, you also use Page Composer to customize pages for other users. You can add fields, add validation, change defaults, rearrange regions, and add external content. Page Composer allows you to work in a WYSIWYG view, and, in some cases, Source view, as shown in Figure 1–6.

Figure 1–6 Page Composer



For more information about customizing pages, see Chapter 3, "Customizing Existing Pages."

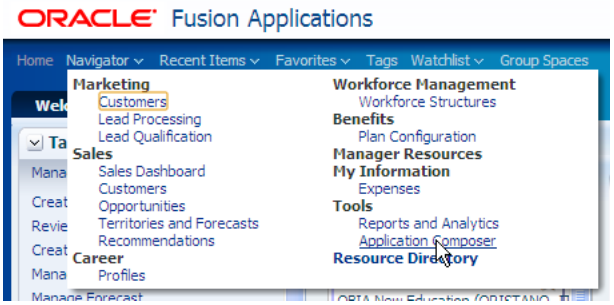
1.3.3 Customizing Pages Using Application Composer

If you want to extend or customize the Sales, Marketing, Customer Center, Trading Community Architecture (TCA), and Order Capture applications that are part of the Oracle Fusion CRM product family of Oracle Fusion Applications, you can use Application Composer to customize your pages, as described in the "Creating a Work Area: Explained" section in the *Oracle Fusion Applications CRM Extensibility Guide*.

> **Note:** Only certain pages are available for customization. For a complete list, see the product-specific documentation from Oracle Enterprise Repository for Oracle Fusion Applications.

You access Application Composer by clicking the **Application Composer** link from the **Navigator** menu of Oracle Fusion Applications, as shown in Figure 1–7.

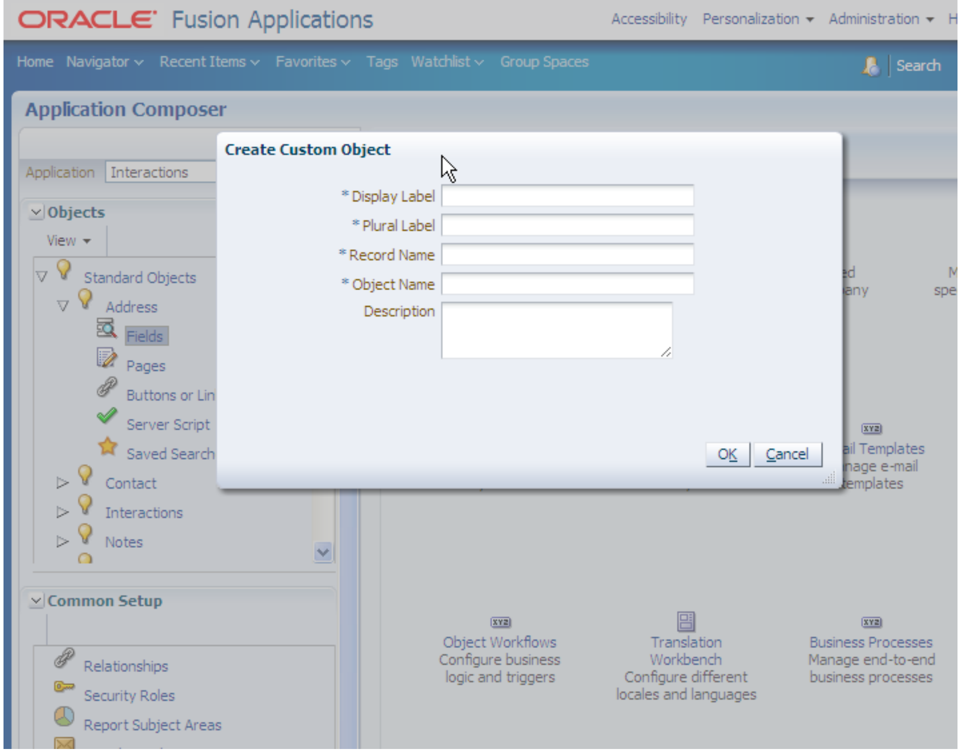
Figure 1–7 Navigator Menu



1.3.4 Creating and Customizing Objects

Application Composer allows business analysts to make more complex runtime customizations to Oracle Fusion CRM applications. In addition to customizing pages, business analysts can customize objects and all the artifacts that support them (such as fields, pages, buttons and links, security, server scripts, and saved searches), and can also extend Oracle Fusion applications by creating completely new objects and artifacts, as shown in Figure 1–8. For more information, see the "Defining Objects: Explained" section in the *Oracle Fusion Applications CRM Extensibility Guide*.

Figure 1–8 Application Composer



When new objects are created, you often also create associated Work Area pages for those objects. You can add those pages to the navigator menu so that they can be accessed in the same way as standard objects. For more information, see Chapter 5, "Customizing Menus."

1.3.5 Creating and Customizing Business Process Flows for Custom Objects

> **Note:** This feature is not available in Oracle Cloud implementations of Oracle Fusion Human Capital Management.

When you create a new object that is not a subclass of another object, you can also create a new object workflow to manage any business processes associated with it. For example, say you used Application Composer to create a marketing object and you want to create an associated approval flow. From within Application Composer, you can access Oracle Business Process Composer and create the process that defines that flow. For applications outside of Oracle Fusion CRM, you access Business Process Composer directly from the **Navigator** menu. For more information about using the Business Process Composer, see Chapter 6, "Customizing and Extending BPMN Processes."

1.3.6 Defining Security Policies for Custom Objects

When you create a new object in Application Composer, you can define security policies for it. A security policy defines the end user's level of access to the data records of the object. For more information about creating security policies for custom Oracle Fusion CRM objects, see the "Securing Custom Objects: Explained" section in the *Oracle Fusion Applications CRM Extensibility Guide*.

1.3.7 Adding Custom Attributes to Business Components

If you need to add an attribute to a business component in an application that is not one of the five Oracle Fusion CRM applications, you can often use flexfields. A **flexfield** enables you to define attributes on a business component and then apply business logic to them. For example, an airline manufacturer might require very specific attributes for their orders that are not provided by the out-of-the-box implementation of an order. Because a flexfield exists for the order business component, you can use it to create and configure the desired attribute. Flexfield configurations are stored in an MDS repository, and so are safe during patching and upgrading. You access flexfields from the Setup and Maintenance menu from the Administration menu. For more information about flexfields, see Chapter 4, "Using Flexfields for Custom Attributes."

1.3.8 Customizing Reports and Analytics

Oracle Fusion Applications comes with a complete set of reports. You can customize these reports (for example, change the layout) to fit your particular business needs. Additionally, if you customize or create a business object, you can create a new report for that object. For more information, see Chapter 7, "Customizing Reports and Analytics."

1.3.9 Performing Design Time Customizations

> **Note:** This functionality is not available in Oracle Cloud implementations.

Design time customizations and extensions include complex changes that require deployment into the runtime environment. For information about design time customizations, see the *Oracle Fusion Applications Extensibility Guide for Developers*.

1.3.10 What You Can Customize and Extend and with Which Tool

There are many scenarios for which you can customize Oracle Fusion applications. The following tables identify for each scenario the artifacts that you can customize or create in Oracle Fusion Applications, what tool you use, the type of user that can make the change, and whether the changes are stored in an MDS repository:

- View page customizations: Table 1–1
- Branding customizations: Table 1–2
- Object customization: Table 1–3
- Business process customizations: Table 1–4
- Report customizations: Table 1–5
- Analysis and dashboard customizations: Table 1–6
- Security customizations: Table 1–7

Note: Application Composer is available only if you want to make changes in the following Oracle Fusion CRM applications:

- Marketing
- Sales
- Customer Center
- Trading Community Architecture (TCA)
- Order Capture

Some features described in this section are not available in Oracle Cloud implementations

Note: While you can customize view pages in Page Composer and Application Composer, only certain pages are configured to allow it. If the customization that you want to make is not available in Page Composer, then you must use JDeveloper to make the customization.

Table 1–1 View Page Customization Scenarios in Oracle Fusion Applications

Customization/ Extension	Tool	Type of User	MDS?	Where to Find Information
Add, move, delete, show, or hide components on a page.	Page Composer	Business Analyst	Yes	Section 3.2, "Editing a Page in Page Composer"
Change a page layout.	Page Composer	Business Analyst	Yes	Section 3.2, "Editing a Page in Page Composer"
Create a site-level search for all users.	Page Composer	Business Analyst	Yes	Section 3.2, "Editing a Page in Page Composer"

Table 1–1 (Cont.) View Page Customization Scenarios in Oracle Fusion Applications

Customization/ Extension	Tool	Type of User	MDS?	Where to Find Information
Customize a page title.	Page Composer	Business Analyst	Yes	Section 3.2, "Editing a Page in Page Composer"
Customize a task list menu.	Page Composer	Business Analyst	Yes	Section 3.2, "Editing a Page in Page Composer"
Customize popup window content.	Page Composer	Business Analyst	Yes	Section 3.2, "Editing a Page in Page Composer"
Add fields, buttons, links, to a standard page (Oracle Fusion CRM).	Application Composer	Business Analyst	Yes	"Application Composer: Using the Application Composer" chapter in the *Oracle Fusion Applications CRM Extensibility Guide*
Customize attributes for a flexfield on a page.	Page Composer	Business Analyst	Yes	Section 3.2, "Editing a Page in Page Composer"
Customize properties for UI components on a standard page.	Page Composer	Business Analyst	Yes	Section 3.3, "Editing Component Properties in Page Composer"
Customize properties for UI components on a standard page (Oracle Fusion CRM).	Application Composer	Business Analyst	Yes	"Editing an Object: Explained" section in the *Oracle Fusion Applications CRM Extensibility Guide*
Customize the **UI Shell template**.	Page Composer	Business Analyst	Yes	Section 3.4, "Editing the UI Shell Template Used by All Pages"

Table 1–2 Branding Customization Scenarios in Oracle Fusion Applications

Customization/ Extension	Tool	Type of User	MDS?	Where to Find Information
Customize the UI Shell template.	Page Composer	Business Analyst	Yes	Section 3.4, "Editing the UI Shell Template Used by All Pages"
Customize report layouts.	Oracle BI Publisher	Business Analyst	No	Section 7.2.2, "Customizing Layouts"

Table 1–3 Object Customization Scenarios in Oracle Fusion Applications

Customization/ Extension	Tool	Type of User	MDS?	Where to Find Information
Customize objects (Oracle Fusion CRM).	Application Composer	Business Analyst	Yes	"Editing an Object: Explained" section in the *Oracle Fusion Applications CRM Extensibility Guide*
Add an attribute to a business object using flexfields (not Oracle Fusion CRM).	Setup and Maintenance work area	Business Analyst	No	Chapter 4, "Using Flexfields for Custom Attributes"
Create objects (Oracle Fusion CRM).	Application Composer	Business Analyst	Yes	"Editing an Object: Explained" section in the *Oracle Fusion Applications CRM Extensibility Guide*

Table 1–3 (Cont.) Object Customization Scenarios in Oracle Fusion Applications

Customization/ Extension	Tool	Type of User	MDS?	Where to Find Information
Add a business object page to the navigator menu	Setup and Maintenance work area	Business Analyst	No	Chapter 5, "Customizing Menus"
Add custom object work area pages to the navigator menu (Oracle Fusion CRM)	Application Composer	Business Analyst	No	Chapter 5, "Customizing Menus"
Add validation to an object (Oracle Fusion CRM).	Application Composer	Business Analyst	Yes	"Groovy Scripting: Explained" section in the *Oracle Fusion Applications CRM Extensibility Guide*"
Customize saved searches for a custom object (Oracle Fusion CRM).	Application Composer	Business Analyst	Yes	"Saved Searches for CRM Objects: Explained" section in the *Oracle Fusion Applications CRM Extensibility Guide*"
Customize object workflows for an object (Oracle Fusion CRM).	Application Composer	Business Analyst	Yes	"Object Workflows: Explained" section in the *Oracle Fusion Applications CRM Extensibility Guide*
Create object workflows for an object (Oracle Fusion CRM).	Application Composer	Business Analyst	Yes	"Object Workflows: Explained" section in the *Oracle Fusion Applications CRM Extensibility Guide*

Table 1–4 Business Process Customization Scenarios in Oracle Fusion Applications

Customization/ Extension	Tool	Type of User	MDS?	Where to Find Information
Create a BPMN process in a BPM project.	Business Process Composer	Business Analyst	Yes	Section 6.2, "Creating a BPM Project"
Create a BPMN approval process in a BPM project (Oracle Fusion CRM).	Application Composer	Business Analyst	Yes	Section 6.2, "Creating a BPM Project"
Customize custom BPM projects.	Business Process Composer	Business Analyst	Yes	Section 6.3, "Customizing BPMN Processes"
Customize custom BPM projects (Oracle Fusion CRM).	Application Composer	Business Analyst	Yes	Section 6.3, "Customizing BPMN Processes"

Table 1–5 Report Customization Scenarios in Oracle Fusion Applications

Customization/ Extension	Tool	Type of User	MDS?	Where to Find Information
Create report layout.	Oracle BI Publisher	Business Analyst	No	Section 7.2.2, "Customizing Layouts"
Customize report layouts.	Oracle BI Publisher	Business Analyst	No	Section 7.2.2, "Customizing Layouts"
Customize style templates.	Oracle BI Publisher	Business Analyst	No	Section 7.2.2, "Customizing Layouts"

Table 1–5 (Cont.) Report Customization Scenarios in Oracle Fusion Applications

Customization/ Extension	Tool	Type of User	MDS?	Where to Find Information
Create a report.	Oracle BI Publisher	Business Analyst	No	Section 7.2.4, "Creating Custom Reports"
Translate a report.	Oracle BI Publisher	Business Analyst	No	Section 7.2.5, "Adding Translations"
Create a report subject area (Oracle Fusion CRM)	Application Composer	Business Analyst	No	"Custom Subject Areas: Explained" section in the *Oracle Fusion Applications CRM Extensibility Guide*

Table 1–6 Analysis and Dashboard Customization Scenarios in Oracle Fusion Applications

Customization/ Extension	Tool	Type of User	MDS?	Where to Find Information
Customize analytics.	Reports and Analytics pane	Business Analyst	No	Section 7.3.2, "Customizing Analytics"
Customize and extend the Oracle BI repository (RPD file).	JDeveloper, Oracle BI Administration Tool	Developer	No	Section 7.3.3, "Customizing the Oracle BI Repository (RPD)"

Table 1–7 Security Customization Scenarios in Oracle Fusion Applications

Customization/ Extension	Tool	Type of User	MDS?	Where to Find Information
Grant access to custom objects (Oracle Fusion CRM).	Application Composer	Business Analyst	No	"Securing Custom Objects: Explained" section in the *Oracle Fusion Applications CRM Extensibility Guide*
Enable elevated privileges customization.	Application Composer	Business Analyst	No	"Securing Custom Objects: Explained" section in the *Oracle Fusion Applications CRM Extensibility Guide*

2

Understanding the Customization Development Lifecycle

This chapter discusses the typical workflow for customizing and extending Oracle Fusion applications. It describes how to use sandboxes to perform customizations in an environment that is separate from the full test environment, publish the changes to a full test environment, and move the changes to other environments.

This chapter includes the following sections:

- Section 2.1, "Understanding Typical Customization Workflows"
- Section 2.2, "Using the Sandbox Manager"
- Section 2.3, "Exporting and Moving Customizations"

> **Note:** For Oracle Cloud implementations, check the Oracle Cloud documentation to determine which Oracle Fusion Applications versions are supported by that Oracle Cloud implementation.

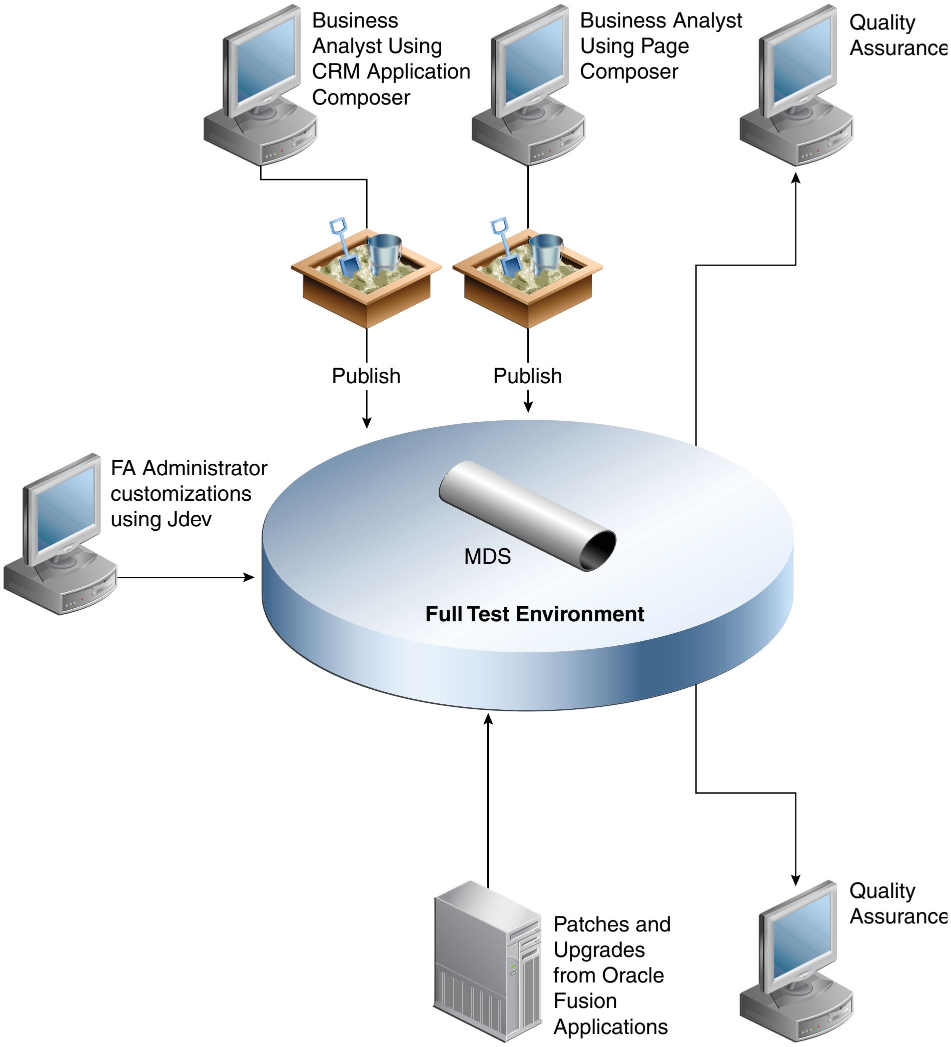
2.1 Understanding Typical Customization Workflows

All customizations and extensions to Oracle Fusion Applications should be done in a full test environment, as shown in Figure 2–1. Typically, this environment contains one or more Oracle Fusion applications that will then be moved to a production environment after all customizations and extensions are complete and tested.

As described in Section 2.1.1, "Runtime Customization Workflow," business analysts using Page Composer and Oracle Fusion CRM Application Composer (Application Composer) make their application customizations in a **sandbox**. Sandboxes store the customizations in isolated, protected Oracle Metadata Services (MDS) labels that are available only when you work in that particular sandbox. The changes can be done in a test-only sandbox (that is, the code in the sandbox is for testing only, and is never deployed), or they can be done in a sandbox that is then published to the full test environment.

After testing, you can then move the customizations to the mainline code as described in Section 2.3, "Exporting and Moving Customizations."

Figure 2–1 Customization Workflow in a Full Test Environment


Business Analyst Using CRM Application Composer
Business Analyst Using Page Composer
Quality Assurance
Publish
Publish
FA Administrator customizations using Jdev
MDS
Full Test Environment
Patches and Upgrades from Oracle Fusion Applications
Quality Assurance

Tip: When you **extend** Oracle Fusion applications, you might want users to be able to configure the extensions using Oracle Fusion Functional Setup Manager. For more information about creating task flows for setup activities for extensions, see the *Oracle Fusion Functional Setup Manager Developer's Guide*.

2.1.1 Runtime Customization Workflow

When you use Application Composer and Page Composer to make runtime customizations to Oracle Fusion applications, you use sandboxes to save your changes in an isolated environment. For example, before you begin making customizations, you create a sandbox named `MySandbox` and make your customizations in that sandbox. If others want to see the customizations, then they would use `MySandbox`.

Note: There are restrictions for when more than one user is working in a sandbox. For more information, see Section 2.2.1, "Sandboxes and Concurrent Usage."

You also use a sandbox when you define security policies for custom objects that you have created using Application Composer. A **security sandbox** stores the security information in new database tables that are available only when you choose to work in that sandbox.

After you complete your customizations, others can review and validate the sandbox. Then you can publish the sandbox to the full test environment where your customizations become part of that repository. For more information about sandboxes, see Section 2.2, "Using the Sandbox Manager."

Note: A **flexfield sandbox** is for testing only and cannot be published. Instead, you deploy a **flexfield** to the full test environment using the flexfield UI. To test a flexfield configuration before deploying it to the full test environment, deploy it to a flexfield sandbox, as described in Section 4.7, "Deploying Flexfield Configurations." The changes that you deploy to a sandbox are isolated from the full test environment and can be seen only by those who make the flexfield sandbox active in their session. After you are satisfied with the changes in the sandbox, you can deploy the changes to the full test environment.

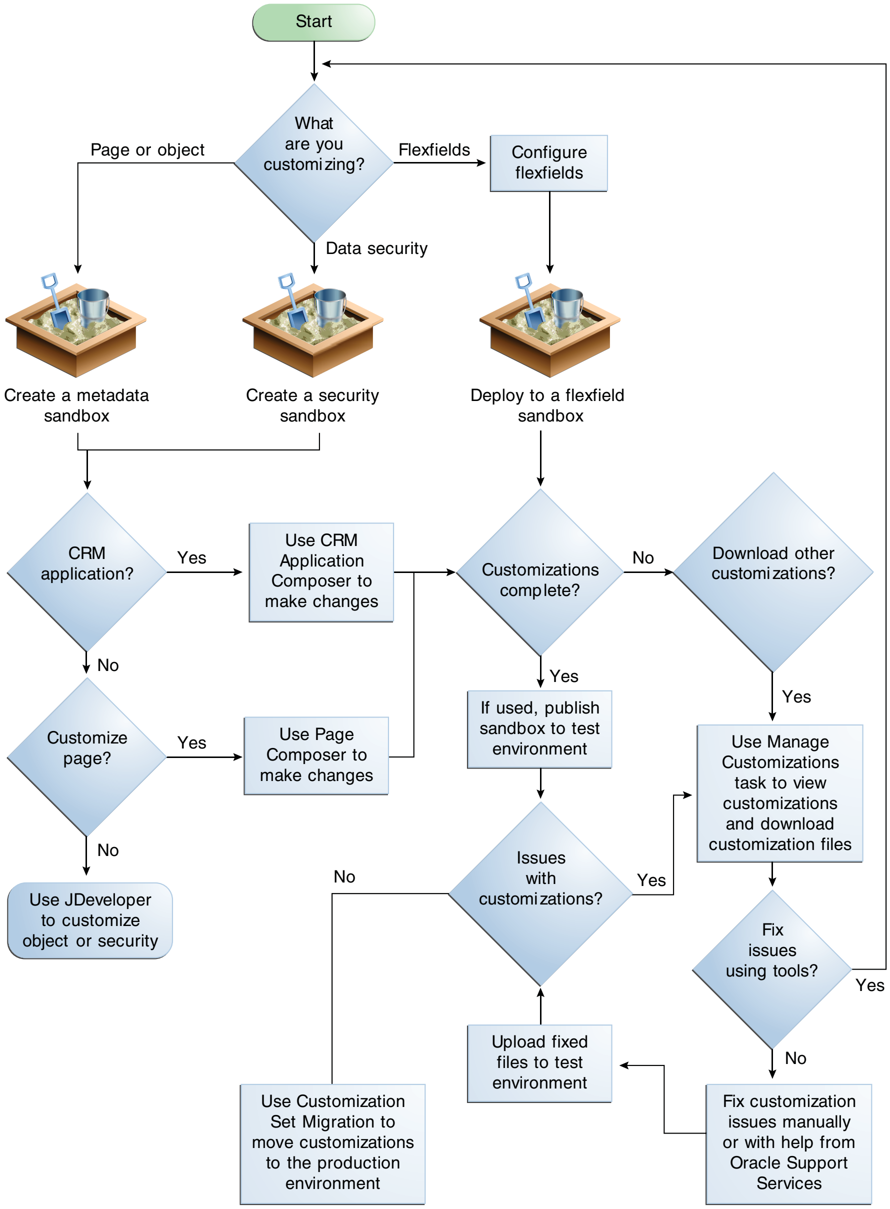
When you publish a sandbox, the published customizations are labeled. Labeling can act as a **save point**, meaning that if a future customization causes issues, you can use the Manage Customizations dialog to remove all customizations done after that point by **promoting** the last known good label **back to the tip** as described in Section 2.1.1.1.3, "Backing Out Customizations."

You can also use the Manage Customizations dialog to view others' customization metatdata files, and to download those files to diagnose any issues. For more information, see Section 2.1.1.1, "Viewing and Diagnosing Runtime Customizations."

Note: The navigator menu, Business Process Modeling Notation (BPMN) process flows, and report customizations do not use an MDS repository. See Chapter 5, "Customizing Menus," Chapter 6, "Customizing and Extending BPMN Processes," and Chapter 7, "Customizing Reports and Analytics" for more information.

Figure 2–2 illustrates the use of sandboxes when customizing pages, objects, and security using Page Composer and Application Composer and when configuring flexfields.

Figure 2–2 Typical Runtime Customization Workflow


Start
What are you customizing?
Page or object
Flexfields
Data security
Configure flexfields
Create a metadata sandbox
Create a security sandbox
Deploy to a flexfield sandbox
CRM application?
Yes
Use CRM Application Composer to make changes
No
Customize page?
Yes
Use Page Composer to make changes
No
Use JDeveloper to customize object or security
Customizations complete?
No
Download other customizations?
Yes
If used, publish sandbox to test environment
Yes
Use Manage Customizations task to view customizations and download customization files
Issues with customizations?
Yes
No
Fix issues using tools?
Yes
No
Upload fixed files to test environment
Fix customization issues manually or with help from Oracle Support Services
Use Customization Set Migration to move customizations to the production environment

2.1.1.1 Viewing and Diagnosing Runtime Customizations

Use the Manage Customizations dialog to view and diagnose runtime customizations that have been made to application pages.

By default, the Manage Customizations dialog displays the customizations that have been performed by the logged-in user. Administrators can optionally see the customizations made by other users.

> **Tip:** You can also use the Manage Customizations dialog to export customizations for diagnostic purposes. For more information, see Section 2.3.3, "Using the Manage Customizations Dialog to Download Customizations."

2.1.1.1.1 Before You Begin Using the Manage Customizations Dialog

You must do the following before you can use the Manage Customizations dialog:

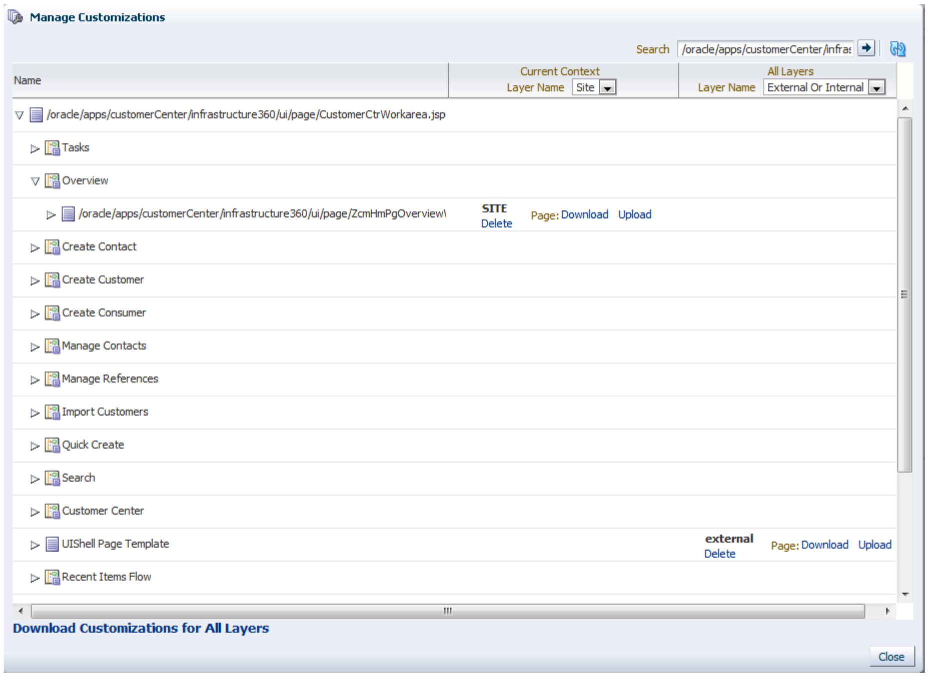
- You must have specific privileges to access the Manage Customizations dialog. Contact your security administrator for details.
- To access the Manage Customizations dialog, choose **Manage Customizations** from the **Administration** menu in the global area.

2.1.1.1.2 Viewing Customizations Using the Manage Customizations Dialog

Figure 2–3 shows the Manage Customizations dialog on a **desktop page** with all artifacts related to the `CustomerCtrWorkarea.jsp` file.

Note: When viewing the Manage Customizations dialog from a **simplified page**, you can see only the top-level nodes and you cannot expand them.

Figure 2–3 Manage Customizations Dialog



Manage Customizations

Search /oracle/apps/customerCenter/infras

Name	Current Context Layer Name Site		All Layers Layer Name External Or Internal	
▽ /oracle/apps/customerCenter/infrastructure360/ui/page/CustomerCtrWorkarea.jsp				
▷ Tasks				
▽ Overview				
▷ /oracle/apps/customerCenter/infrastructure360/ui/page/ZcmHmPgOverview\	SITE Delete	Page: Download Upload		
▷ Create Contact				
▷ Create Customer				
▷ Create Consumer				
▷ Manage Contacts				
▷ Manage References				
▷ Import Customers				
▷ Quick Create				
▷ Search				
▷ Customer Center				
▷ UIShell Page Template			external Delete	Page: Download Upload
▷ Recent Items Flow				

Download Customizations for All Layers

Close

As shown in Figure 2–3, a customization was made to a **page fragment file** in the **site layer**, and a customization was made to the UIShell Page Template in the external layer (the external layer is a layer that is specific to Oracle Fusion Customer Relationship Management applications). By default, the customizations in the **Current Context** column show the customizations that the currently logged-in user would see in a running version of the application. All customizations, including those the currently logged-in user might not see (perhaps because of security policies), are listed in the **All Layers** column.

For example, say you are a customization developer. Because you as a user have been assigned a particular **role**, the page that you see will display only those customizations appropriate for your role. Those customizations will be listed in the **Current Context** column (provided you chose the proper layer from the **Layer Name** dropdown menu). This helps you to determine what customizations are affecting what you are seeing.

There may, however, be customizations that you do not see. You can view those customizations from the **All Layers** column.

Sometimes, an administrator might need to view a **personalization** that was made by another end user. For example, a user might have made an error while personalizing a page and that page is no longer displayed for the user. Because the user cannot access the page, the user cannot correct the error. In this case, the administrator can access the page, request to see the user's changes, and delete those changes to restore the page to its original settings.

To view customizations:

1. Go to the page for which you wish to view customizations.
2. Select the task flow or page fragment that contains the customizations.
3. From the **Administration** menu in the global area of Oracle Fusion Applications, choose **Manage Customizations**.

   > **Tip:** After you are in the Manage Customizations dialog, you can change the page, page fragment, or task flow for which you are viewing customizations using the **Search** field.

4. From the **Layer Name** dropdown list for the **Current Context** column, select the customization layer for which you want to see the customizations as the user (or users) see it. For more information about customization layers, see Section 1.2, "Understanding Customization Layers."
5. If you have administrator privileges and you want to see the customizations for a user other than yourself, then select **Select User** from the **Layer Name** dropdown list for the **All Layers** column, and select the user. You can select multiple users by repeating this step.

2.1.1.1.3 Backing Out Customizations Metadata labels identify the state of the objects in an MDS repository at a given point in time. Labels can serve as **save points** to which you can **roll back** your customizations if the customizations create problems. You roll back to a label by making that label become the latest version. This action is often referred to as **promoting the label to the tip**.

> **Note:** You cannot promote design time customizations. If a design time customization is causing a problem, you must undeploy the changes, fix the issues, and then redeploy to a sandbox.

To create a label across the entire MDS repository, use the `createMetadataLabel` Oracle WebLogic Scripting Tool (WLST) command as described in the "Creating Metadata Labels" section of the *Oracle Fusion Middleware Administrator's Guide*. You can also create a label when you save a customization in Page Composer.

To roll back all customizations, use the `promoteMetadataLabel` WLST command as described in the "Promoting Metadata Labels" section of the *Oracle Fusion Middleware Administrator's Guide*.

> **Note:** WLST is not available in Oracle Cloud implementations.

To roll back the customizations for a specific page, use the Manage Customizations dialog accessed from Page Composer as described in the following steps. Note that when you use the Manage Customizations dialog, you are rolling back only the customizations for the page and its `pageDef` file. You are not rolling back the other customizations made at the label's save point.

> **Note:** Some labels are created automatically. For example, customizations that are created in a sandbox are automatically labeled when the sandbox is published. You can identify an automatically created label by its prefix. For a list of these prefixes, see the "Managing Oracle Fusion Applications-Specific Labels in the Oracle Metadata Repository" section in the *Oracle Fusion Applications Administrator's Guide*.

To promote a page's customizations to the tip:

1. Go to the page for which you wish to view customizations.
2. From the **Administration** menu in the global area of Oracle Fusion Applications, choose **Customize** ***page_name*** **Pages** to open the page in Page Composer.
3. In the tool bar, click **Manage Customizations**.
4. To promote a customization to the tip, click **Promote** for the corresponding artifact.
5. In the Promote Documents dialog, select the label that you want to promote to the tip and click **OK**.

2.1.2 Design Time Customization Workflow

> **Note:** The functionality described in this section is not available in Oracle Cloud implementations.

For information about the design time customization workflow, see the *Oracle Fusion Applications Extensibility Guide for Developers*.

2.2 Using the Sandbox Manager

The sandbox manager is a tool for managing the different types of customization changes that can be applied to an application. The different types of sandboxes are:

- Metadata

  The metadata sandbox supports making changes to the application's metadata stored in the MDS repository.

- Security

  The security-enabled sandbox supports making **data security** changes.

- Flexfield

  The flexfield sandbox is not created using the sandbox manager. Use the flexfield UI to make changes to the flexfields and then deploy them to the sandbox. The flexfield deployment process manages the creation of the sandbox. For more information about flexfields, see Section 4.7, "Deploying Flexfield Configurations."

To **customize** an Oracle Fusion application in runtime, you must first create a sandbox and then use Page Composer or Application Composer to make the customizations.

These changes will be contained within the sandbox so they do not affect the mainline code. You then test and validate the changes by publishing the sandbox to the full test environment. After the application has been tested, it can then be moved to the production environment. The customizations created in the sandbox will be migrated to the production environment and will be available to users of the system.

You make changes to an application at runtime in a sandbox so that the changes are isolated from the mainline code. The mainline code is a branch of data that serves as a single source. After you are satisfied with the changes in the sandbox and want to commit them, you can publish the metadata or security-enabled sandbox to the mainline code. If customizations existed in the mainline code before the sandbox was created, you will see the customization information in the sandbox. If you add customizations to the mainline code after a sandbox has been created, you will not see the newly added customizations in the sandbox. You will need to exit, publish, or destroy the sandbox and create a new sandbox to see the customizations that were newly added to the mainline code.

Flexfield sandboxes are for testing only and cannot be published. You make flexfield configurations that are then stored in a database, and then deploy those configurations to a sandbox to see the resulting deployment artifacts in a sandbox environment. Flexfields are deployed directly to the mainline code using the flexfield UI. For more information about flexfields, see Section 4.7, "Deploying Flexfield Configurations."

You can use runtime tools to customize the application. The sandbox manager works with Application Composer and Page Composer to customize objects and pages:

- For information about using Application Composer, see the "Application Composer: Introduction" chapter in the *Oracle Fusion Applications CRM Extensibility Guide*.
- For information about using Page Composer, see Chapter 3, "Customizing Existing Pages."

Oracle Business Process Composer and Oracle SOA Composer are also runtime customization tools, but they do not use the sandbox manager. They have their own mechanisms for handling customization changes:

- For information about using Oracle Business Process Composer, see Chapter 6, "Customizing and Extending BPMN Processes."
- For information about using Oracle SOA Composer, see the "Customizing and Extending SOA Components" chapter in the *Oracle Fusion Applications Extensibility Guide for Developers*.

For non-Cloud implementations, a **metadata sandbox** that you create using the sandbox manager is available in JDeveloper when you are creating and deploying customizations intended for a deployed Oracle Fusion application in Oracle WebLogic Server. The available sandboxes will appear in a selection list in JDeveloper during deployment. For more information, see the "Deploying Oracle ADF Customizations and Extensions" section in the *Oracle Fusion Applications Extensibility Guide for Developers*. Note that the security sandboxes created using the sandbox manager are not available in JDeveloper.

In Application Composer, you customize security policies using a sandbox specifically for editing them. When you enable the security sandbox, the operation will duplicate the schema for Oracle Fusion Data Security tables, which is a lengthy setup operation. Therefore, ensure that you are fully ready to customize the security policies *before* you enable the security sandbox.

The security policy customizations that you publish from a sandbox will be merged into the Oracle Fusion security policy repository as part of the native application, and will overwrite any previous customizations. Ensure that you are not editing the same object concurrently with another user to avoid inconsistencies that can result when multiple users edit the security policies associated with the same object in different sandboxes.

The metadata and security sandbox sessions can be saved, downloaded, and imported as files into other Oracle Fusion applications.

2.2.1 Sandboxes and Concurrent Usage

In the customization runtime workflow, you use sandboxes to isolate the changes from the mainline code for testing and validating. After you are satisfied with the changes, you can publish the changes back to the mainline code. You can also create sandboxes for testing purposes only, and not publish them.

There are two types of sandboxes:

- Sandboxes intended to be published.

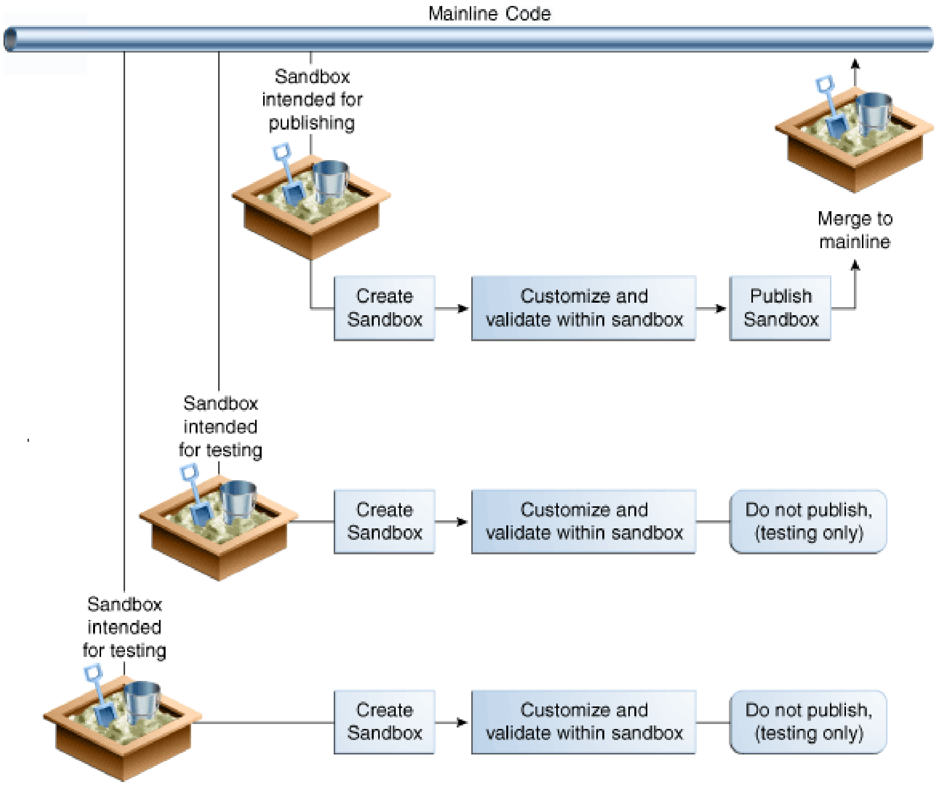
  These sandboxes will be merged back to the mainline code.

- Sandboxes intended for test-only purposes.

  These test-only sandboxes will not be published and therefore produce no concurrency conflicts between sandboxes. You can have many test-only sandboxes at the same time. But if you have multiple users working on the same test-only sandbox, then they must adhere to the guidelines described in Section 2.2.1.3, "Guidelines for One Sandbox, Multiple Users."

Figure 2–4 illustrates the two types of sandboxes and their relationship to the mainline code.

Figure 2–4 Two Types of Sandboxes


Mainline Code
Sandbox intended for publishing
Merge to mainline
Create Sandbox
Customize and validate within sandbox
Publish Sandbox
Sandbox intended for testing
Create Sandbox
Customize and validate within sandbox
Do not publish, (testing only)
Sandbox intended for testing
Create Sandbox
Customize and validate within sandbox
Do not publish, (testing only)

When multiple users can customize an application using sandboxes, there are two types of concurrency conflicts:

- Conflicts within a sandbox: Users overwriting changes created by other users, either directly by changing the same artifact, or indirectly by affecting files that are shared between the artifacts. For more information, see Section 2.2.1.3, "Guidelines for One Sandbox, Multiple Users."
- Conflicts between sandboxes (intended for publishing only): Multiple sandboxes with the same customized artifact publishing to the mainline code. For more information, see Section 2.2.1.4, "Guidelines for Multiple Sandboxes, Multiple Users."

An application artifact typically includes several metadata files. Therefore, creating or editing an artifact usually means making changes, whether directly or indirectly, to more than one file. Some of these metadata files may be shared between artifacts.

Note: Many customization scenarios, including customizing security policies, involve editing the same underlying artifacts. The only way to be certain of eliminating the possibility of any concurrency conflicts is to allow only one user at a time in an active sandbox. If you must have multiple users or multiple sandboxes, then follow the usage guidelines described in Section 2.2.1.3, "Guidelines for One Sandbox, Multiple Users," and Section 2.2.1.4, "Guidelines for Multiple Sandboxes, Multiple Users."

Note: Features or functionality described in this note are not available in Oracle Cloud implementations.

When a sandbox intended for publishing is active, do not use Oracle Enterprise Manager Fusion Applications Control or Oracle WebLogic Scripting Tool (WLST) commands to import any metadata documents into the mainline code. The imported metadata may be in conflict with the changes made in the sandbox. For more information, see the "Transferring Metadata Using Fusion Middleware Control" section of the *Oracle Fusion Middleware Administrator's Guide*.

If you need to import multiple customizations available in the metadata repository for an application, you can use the `exportMetadata` WLST command. For more information, see the "Application Metadata Management Commands" section of *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*. This command saves the customization files in a JAR file that you can import into your workspace. For more information about procedures, see the "Viewing ADF Library Runtime Customizations from Exported JARs" section of the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*.

2.2.1.1 Conflicts Within a Sandbox

Conflicts within a sandbox can arise when multiple users are customizing an application using the same sandboxes at the same time, because more than one user may be attempting to customize the same artifact, or performing a customization task that indirectly affects other shared files. An example of a **direct conflict** is when different users attempt to customize the same page, the same fragment, or the same metadata file within the same layer. An example of an **indirect conflict** is when two users, each creating their own object, cause a conflict in the metadata file that tracks which new objects have been created by both saving their changes around the same time.

Conflicts may also arise when users are editing a shared artifact, such as when a user performs an operation that adds or edits a translatable string. For example, a user edits a field's display label or help text, or a validation rule's error message, while another user performs an operation around the same time that similarly affects translatable strings. Another example of a shared artifact conflict is when two or more users are working in navigator menus that are shared across applications.

When you are using Page Composer, you must use a sandbox. Page Composer provides concurrency warning messages when there is a potential customization conflict with another user. For Page Composer, these warning messages help prevent conflicts. For more information, see the "A Few Notes About Concurrent Users" section of the *Oracle Fusion Middleware User's Guide for Oracle WebCenter Portal: Spaces*.

When you are using Application Composer, you must also use a sandbox. Application Composer provides error messages when there is a concurrency conflict. If multiple users in a single sandbox attempt to edit the same artifact, then an error message is displayed. For example, if two users are editing the label of the same attribute, then Application Composer will display an error message.

2.2.1.2 Conflicts Between Sandboxes

Conflicts between sandboxes can arise when there is more than one sandbox intended for publishing in use. If two sandboxes contain customization changes to the same artifact and both are being published, then the sandbox that is being published last is given an option (by the sandbox manager) to overwrite the changes for that artifact from the sandbox that was published first. If the user working in the second sandbox decides to force the publication of the second sandbox, then the changes published by the first sandbox are overwritten. For example, these types of conflicts can also occur with shared metadata files such as a **resource bundle** that stores translatable strings.

> **Note:** You should force-publish a sandbox only when there is no other option available. Typically, if you encounter a conflict during publishing, you should delete the sandbox, create a new sandbox, re-enter your changes and try publishing again.

If there is a concurrent change made in the mainline code after the sandbox was created and the user attempts to publish that sandbox, then such conflicts are detected at publication time and error messages occur.

> **Tip:** If you encounter a message showing a conflict on `oracle/apps/menu/fnd/applcore/dataSecurity/dataSecurityService/mds/DSMO.xml` when you publish your sandbox, this means the security changes that you made in your sandbox conflict with other security changes in the mainline code. Abort the sandbox and recreate your changes in a new sandbox.

2.2.1.3 Guidelines for One Sandbox, Multiple Users

Regardless of whether the sandbox is intended for publishing or for test-only, if multiple users are allowed to work in a single sandbox simultaneously, follow these guidelines:

- Multiple concurrent users in the same sandbox must operate only on different and unrelated objects.

  For example, if `user1` updates `object1`, then `user2` can update `object2` but should not update `object1`. Be aware that if both modifications involve changes to translatable strings, then saving changes to separate objects around the same time may still cause a conflict in the resource bundle that stores the translatable strings.

- If multiple users update the same artifact concurrently (either the same object or the same underlying frequently modified file), then they will get a concurrent update error. In this case, the second user's changes will not be saved (the **Save** button will be disabled) and one of the users will have to cancel and try again.

  Application Composer retains any uncommitted changes in the UI. For example, if the user is editing a Groovy expression when the error is encountered, then the expression is retained in the editor so the user can copy and paste the expression to try customizing again. (No metadata corruption or partial transactional commit operations will happen.)

- Users in the same sandbox will see the changes created by one another. The latest version of each object gets loaded on-demand the first time it is viewed in Application Composer. Users can refresh their view to the latest definition from MDS and see the changes listed in the tree structure at the left-hand side of the Application Composer UI. However, if there are ADF Business Components customizations, then users might need to sign out and sign back in to see those changes reflected in the UI.

2.2.1.4 Guidelines for Multiple Sandboxes, Multiple Users

If multiple users are permitted to work in multiple sandboxes, the guidelines in Section 2.2.1.3, "Guidelines for One Sandbox, Multiple Users," and follow these guidelines:

- There can be any number of test-only sandboxes operating concurrently. That is, multiple users can use multiple sandboxes concurrently for testing if these sandboxes will never be published. Sandboxes that are used for testing only, and that are not published, cause no conflicts with each other. Be aware, however, that all modifications will be lost when the sandboxes are destroyed.

  Note: Even though sandboxes that are for test-only cause no conflicts with each other because they are not published, multiple users who work in the same test-only sandbox still must follow the guidelines in Section 2.2.1.3, "Guidelines for One Sandbox, Multiple Users."

- For sandboxes that are not for test-only and will be published, multiple concurrent sandboxes can be used only if they operate on mutually exclusive artifacts. For example, you can have one sandbox that contains a page that is being customized to add a task flow, and another sandbox that contains a different page from a different application.

  Note: For Application Composer, use only one sandbox at a time that is intended for publishing. You cannot have multiple concurrent sandboxes that are intended for publishing. You can still have other sandboxes that are test-only.

- If an artifact is updated in both the mainline code and in the sandbox (or in two different sandboxes), when the sandbox is published, such conflicts are detected and an error is raised. At this point, cancel publishing of the sandbox to avoid overwriting previous changes.

> **Note:** For a sandbox that contains ADF Business Components customizations, sign out and sign back in after switching in or out of this sandbox to avoid any inconsistencies between the runtime caches and the ADF Business Components definitions.
>
> Sign out and sign back in when you are performing any of the following sandbox-related actions:
>
> - Before entering a sandbox
> - After exiting a sandbox
> - After publishing a sandbox
> - After destroying a sandbox

2.2.2 Setting Up Sandboxes

When you create a sandbox, the currently available metadata is gathered into a sandbox session. You designate a sandbox to be the active sandbox by choosing either the newly created sandbox or selecting from a list of existing sandboxes in a table. The active sandbox is the context for all changes. After you activate the sandbox, you can make customization changes to the application artifacts and these changes will be stored in the sandbox. The sandbox uses a database to store the actual change information. After you are satisfied with the changes, you can publish the sandbox, or deploy the flexfield, and the changes will be merged into the mainline code and the sandbox will be archived.

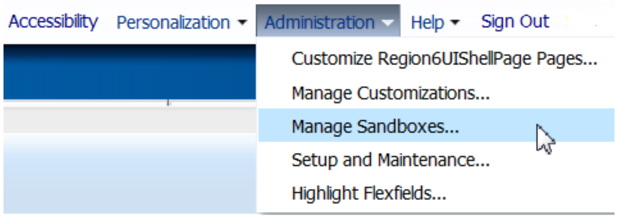
To set up a sandbox:

1. Access the sandbox manager.

   For nonflexfield sandboxes, you can access the sandbox manager by selecting the **Manage Sandboxes** menu item from the **Administration** menu in the global area of Oracle Fusion Applications.

   Figure 2–5 shows the Manage Sandboxes menu selection from the Administration dropdown list.

Figure 2–5 Accessing the Sandbox Manager

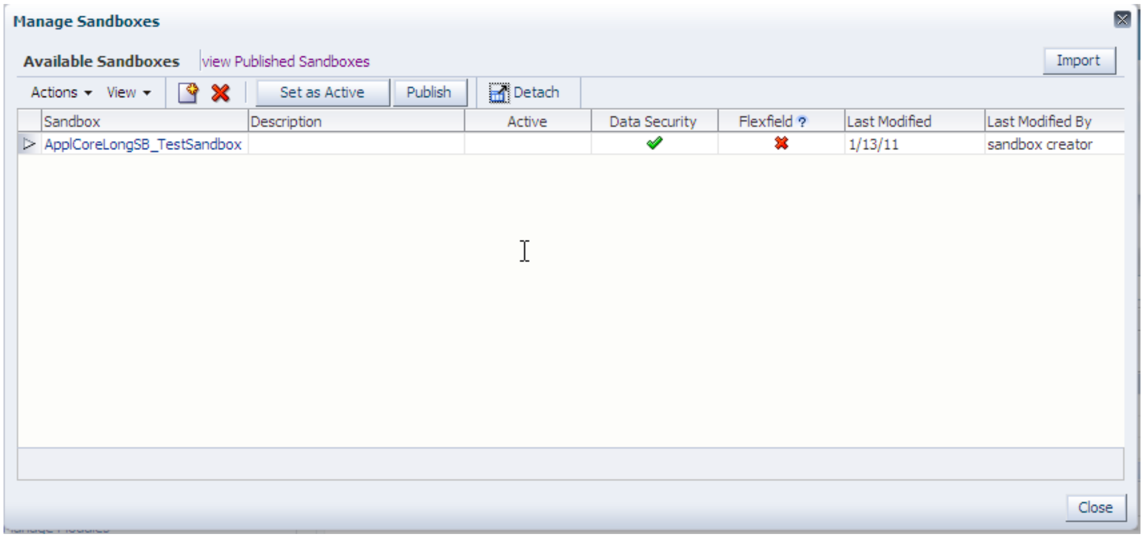


For flexfields, you must be in the Manage Descriptive Flexfields task or the Manage Extensible Flexfields task and then select the **Deploy Flexfield to Sandbox** menu item to deploy the customizations to a flexfield sandbox. Flexfield changes are stored in flexfield metadata in the database and are available to users only when the flexfield is deployed, as described in Section 4.7, "Deploying

Flexfield Configurations." Thus for flexfields, the remaining steps in this procedure do not apply.

2. In the Manage Sandboxes dialog, you can create a new sandbox, select from a list of sandboxes in the Available Sandboxes page, import a sandbox as a file, or perform other sandbox functions. Figure 2–6 shows a Manage Sandbox dialog with one available sandbox.

Figure 2–6 Manage Sandboxes Dialog



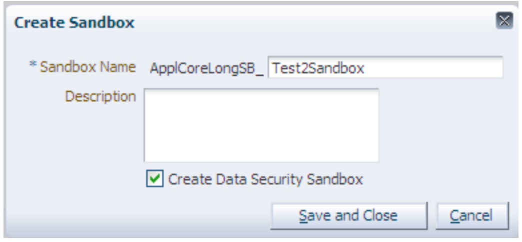
	Sandbox	Description	Active	Data Security	Flexfield ?	Last Modified	Last Modified By
▷	ApplCoreLongSB_TestSandbox			✓	✗	1/13/11	sandbox creator

3. Create a new sandbox by clicking the **New** icon or selecting **Actions** then **New**

4. In the Create Sandbox dialog, enter a name for the sandbox. If you want a security-enabled sandbox, then select **Create Data Security Sandbox** and click **Save and Close**. Figure 2–7 shows the dialog with security enabled.

Note: Because setting up the security sandbox requires duplicating the schema for Oracle Fusion Data Security tables, this will always be a lengthy operation in Application Composer. Allow sufficient time for the process to be completed and do not terminate it early. You may want to defer customizing security and enabling the security sandbox until you are sure that customizations are required.

Figure 2–7 Create Sandbox Dialog



* Sandbox Name ApplCoreLongSB_ Test2Sandbox

Description

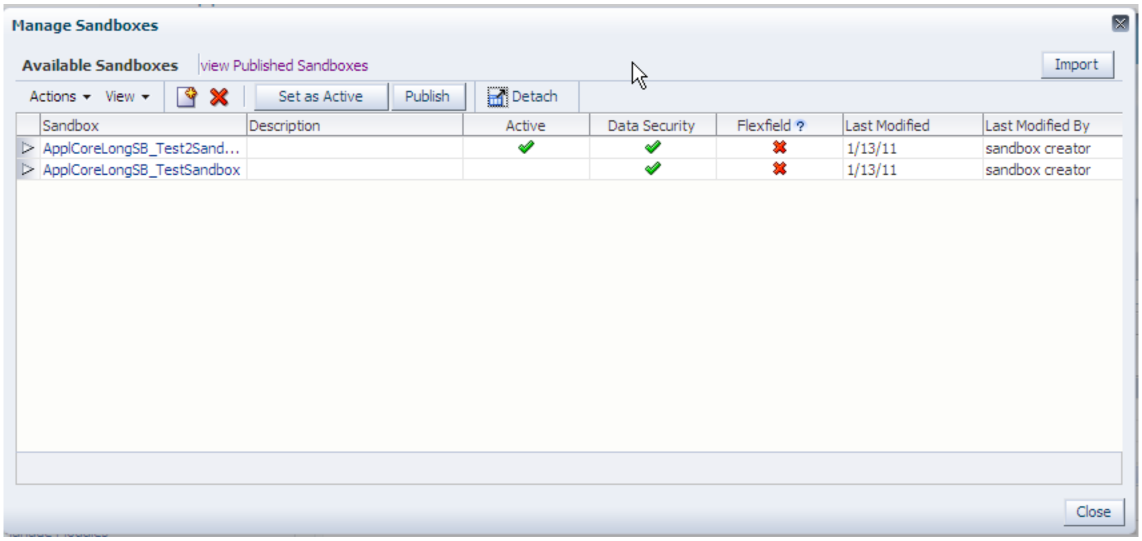
☑ Create Data Security Sandbox

When the sandbox has been created successfully, a confirmation dialog appears. Click **OK** to dismiss the dialog.

5. Identify the active sandbox. You can make the newly created sandbox or an existing sandbox the active sandbox.

In the Manage Sandboxes dialog, in the Available Sandboxes page, select the sandbox you want to make active and click **Set as Active** as shown in Figure 2–8.

Figure 2–8 Manage Sandboxes Dialog



Sandbox	Description	Active	Data Security	Flexfield	Last Modified	Last Modified By
ApplCoreLongSB_Test2Sand...		✓	✓	✗	1/13/11	sandbox creator
ApplCoreLongSB_TestSandbox			✓	✗	1/13/11	sandbox creator

Only one sandbox can be active at one time. The customization changes are captured in the active sandbox.

6. Export a sandbox.

   A sandbox can be exported as a file for transporting, sharing, and other uses where packaging it as a file is required. Consequently, a sandbox exported as a file can be imported.

   In the Sandbox Details dialog, click **Download All**. Enter the location and file name for the exported sandbox.

7. Import a sandbox.

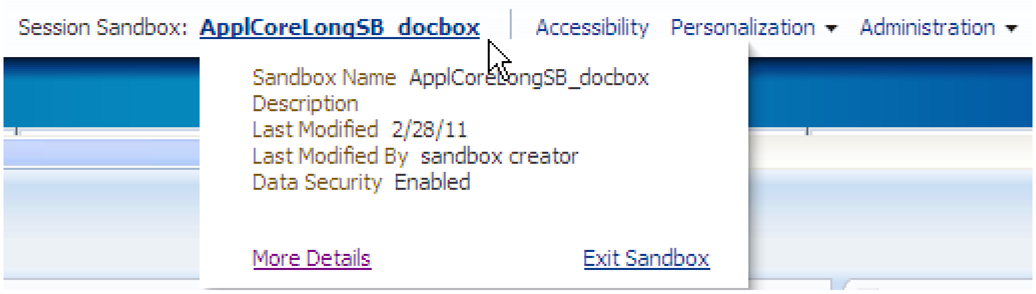
   In the Manage Sandboxes dialog, click **Import**.

   In the Import dialog, click **Browse** to navigate to the file or enter the fully qualified file name.

8. Exit the sandbox.

   Mouse over the sandbox name next to the **Session Sandbox** label in the global area of the Oracle Fusion application to open a popup dialog, as shown in Figure 2–9. Click **Exit Sandbox**.

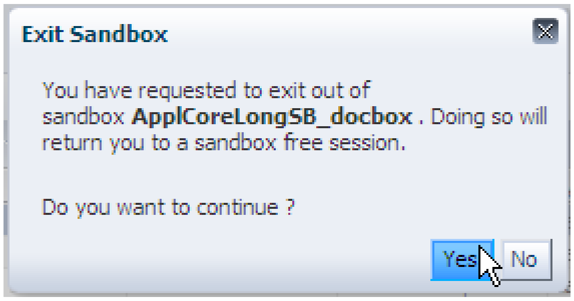
Figure 2–9 Sandbox Popup Dialog



Sandbox Name ApplCoreLongSB_docbox
Description
Last Modified 2/28/11
Last Modified By sandbox creator
Data Security Enabled

In the Exit Sandbox dialog, click **Yes**, as shown in Figure 2–10.

Figure 2–10 Exit Sandbox Dialog



Exit Sandbox

You have requested to exit out of sandbox **ApplCoreLongSB_docbox** . Doing so will return you to a sandbox free session.

Do you want to continue ?

2.2.3 Publishing Sandboxes

If there are changes to the mainline code from another source and you publish your sandbox, then the mainline code is not overwritten. If there are conflicts, then you will be warned and given a chance to fix the conflicts.

To publish a sandbox:

1. Make the customization changes to the application by going to the various customization environments. For example, you can create new objects and customize the objects in Application Composer.

   You can make metadata changes in the sandbox. Metadata changes are usually associated with MDS. These include making changes to a page, to Oracle ADF customization, and to objects. For more information about making changes to the page, see Chapter 3, "Customizing Existing Pages." For more information about making changes to objects, see the "Defining Objects: Explained" section in the *Oracle Fusion Applications CRM Extensibility Guide*.

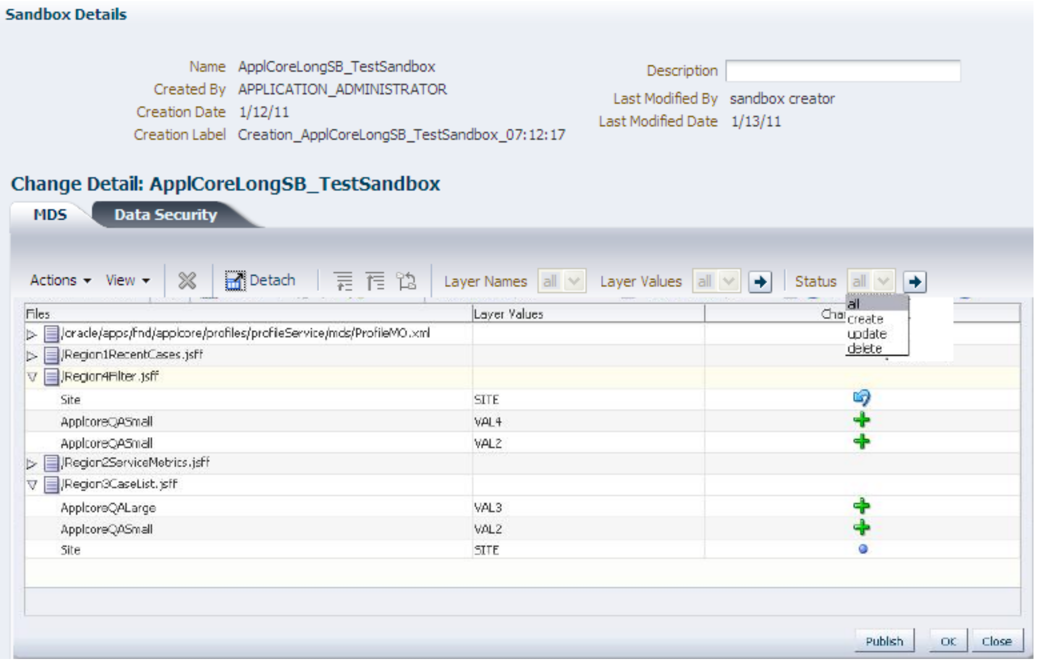
   You make security changes by applying your changes to a security-enabled sandbox. Test your security changes and policies in the sandbox before you commit the changes by publishing the sandbox. For more information about customizing security policies, see the "Securing Custom Objects: Explained" section in the *Oracle Fusion Applications CRM Extensibility Guide*.

   For more information about security in custom business components in non-Cloud implementations, see the "Customizing Security for Oracle ADF Application Artifacts" chapter in the *Oracle Fusion Applications Extensibility Guide for Developers*.

2. Test or validate the changes at runtime using test or production systems and any combination of validation setups.

   In the Manage Sandboxes dialog, click the sandbox link to open the Sandbox Details dialog. You can view the layers and objects where you have made customization changes, as shown in Figure 2–11.

Figure 2–11 Sandbox Details Dialog



Sandbox Details

Name ApplCoreLongSB_TestSandbox
Created By APPLICATION_ADMINISTRATOR
Creation Date 1/12/11
Creation Label Creation_ApplCoreLongSB_TestSandbox_07:12:17

Description
Last Modified By sandbox creator
Last Modified Date 1/13/11

Change Detail: ApplCoreLongSB_TestSandbox

Files	Layer Values	Cha
▷ /oracle/apps/fnd/applcore/profiles/profileService/mds/ProfileMO.xml		
▷ /Region1RecentCases.jsff		
▽ /Region4Filter.jsff		
Site	SITE	↻
ApplcoreQASmall	VAL4	+
ApplcoreQASmall	VAL2	+
▷ /Region2ServiceMetrics.jsff		
▽ /Region3CaseList.jsff		
ApplcoreQALarge	VAL3	+
ApplcoreQASmall	VAL2	+
Site	SITE	●

3. Publish the sandbox to the mainline code.

   After you are satisfied with the changes, select the sandbox and click **Publish** in the Manage Sandbox dialog or in the Sandbox Detail dialog to commit the changes to the mainline code.

2.3 Exporting and Moving Customizations

There are several tools available for exporting and moving customizations. These tools enable you to perform the following tasks:

- Move customizations and extensions to another Oracle Fusion Applications environment, such as the production environment.
- Diagnose issues seen in the test environment.
- Send files to Oracle Support Services for further diagnosing.
- Import a customization into another environment. For example, a customization developer using JDeveloper might need to see customizations done by someone else.

Table 2–1 lists the tools to use to export and move customizations and extensions.

> **Note:** Oracle WebLogic Scripting Tool (WLST) and Oracle Enterprise Manager Fusion Applications Control (Fusion Applications Control) are not available in Oracle Cloud implementations. For Oracle Cloud implementations, to complete the tasks in Table 2–1 that require use of WLST commands or Fusion Applications Control, log a service request using My Oracle Support at `https://support.oracle.com`.

Table 2–1 Tools to Use for Exporting and Moving Customizations

Task	Tools to Use
Move all customizations described in Section 2.3.1, "Using Customization Set Migration to Move Customizations" to another Oracle Fusion Applications environment.	Customization Set Migration.
Move only MDS customizations made to pages and the user interface to another Oracle Fusion Applications environment.	WLST commands or Fusion Applications Control.
Move only descriptive flexfield configurations to another Oracle Fusion Applications environment.	Oracle Fusion Functional Setup Manager (Functional Setup Manager). Moves configurations for a specified module. To move configurations for all modules, use Customization Set Migration.
Move only extensible flexfield configurations to another Oracle Fusion Applications environment.	Functional Setup Manager. Moves configurations for a specified module. To move configurations for all modules, use Customization Set Migration.
Move only value set configurations to another Oracle Fusion Applications environment.	Functional Setup Manager. Moves configurations for a specified module. To move configurations for all modules, use Customization Set Migration.
Move only lookups to another Oracle Fusion Applications environment	Functional Setup Manager. Move application standard lookups, application common lookups, or both.
Move only data security policies to another Oracle Fusion Applications environment.	Functional Setup Manager. Note that this does not move Oracle Fusion Human Capital Management roles.
Export customizations to a file to help diagnose an issue.	WLST commands or Manage Customizations dialog.
Export customizations for import into an application workspace in JDeveloper.	WLST commands or Manage Customizations dialog.

2.3.1 Using Customization Set Migration to Move Customizations

The Customization Set Migration dialog enables you to move customizations and extensions from one Oracle Fusion Applications environment to another. You **create** a set of all customizations and extensions that have been made to the **source** environment, download the customization set, upload it into another (**target**) environment, and then **apply** the customizations that are in the set to that environment. Should you need to back out the customizations that you have applied, you can **restore** the environment to the state that it was in before you applied the customizations.

Note: The Customization Set Migration dialog is to be used to move customizations from one environment to another. It is not a tool for backing up and restoring customizations in the same environment.

After you have downloaded a customization set from the source environment, you must **compact** the set to remove the temporary files that were created during the create process. After you apply a customization set on the target environment, you must compact it to remove the temporary files that were created when you uploaded it into the target environment.

The customization set includes customizations across all Oracle Fusion Applications product families, such as Oracle Fusion Financials, Oracle Fusion Customer Relationship Management (Oracle Fusion CRM), and Oracle Fusion Human Capital Management (Oracle Fusion HCM).

The customization set includes only the customizations and extensions that you make using the following tools and features. Personalizations are not included in the set. (For the definitions of customization, extension, and personalization, see Section 1.1, "Understanding Customizing and Extending Oracle Fusion Applications.")

- Application Composer, with the exception of the following customizations:
  - Object artifacts that were generated from the Import and Export page in Application Composer to make extensions available for importing and exporting as described in the "Importing and Exporting Custom Objects: Explained" section in the *Oracle Fusion Applications CRM Extensibility Guide*.
  - Email templates, as described in the "E-Mail Templates: Explained" section in the *Oracle Fusion Applications CRM Extensibility Guide*
  - Submitted custom subject areas that are available from Oracle BI Composer and Oracle Business Intelligence Answers, as described in the "Publishing Custom Subject Areas: Explained" section in the *Oracle Fusion Applications CRM Extensibility Guide*.
  - User names and passwords for secured SOAP web service connections.
  - The enabling of the attachment feature for custom objects.

  Note: The steps in this section include steps for manually moving the customizations and extensions that are described in the preceding bulleted list.

- Page Composer
- Tasks and dialogs for configuring descriptive and extensible flexfields and value sets
- Manage Menu Customizations task
- Manage Oracle Social Network Objects task
- Manage Standard Lookups task
- Data security policies created or customized using Oracle Authorization Policy Manager
- For non-Cloud implementations, customization metadata that is created from JDeveloper using the Oracle Fusion Applications Administrator Customization **role** and then packaged and deployed to the source Oracle Fusion Applications environment as described in the "Using Oracle JDeveloper for Customizations" chapter in the *Oracle Fusion Applications Extensibility Guide for Developers*.

Notes: The Customization Set does not include code extensions, such as managed beans, that you implement in JDeveloper using the Oracle Fusion Applications Developer role. These code extensions are deployed into the application as a library JAR file and must be redeployed manually into each target environment.

Tip: To prevent including in-progress customizations in the customization set, always make Application Composer, Page Composer, and JDeveloper customizations in a sandbox. The customization set does not include customizations that are in sandboxes.

WARNING: During the time that an apply or restore is processing Oracle Fusion CRM customizations and extensions, the following issues can occur for approximately 5 minutes of the processing time:

- **End users can sign in, but they might receive page-not-found (404) errors when they access dashboard and work area pages.**
- **Oracle Enterprise Scheduler jobs that are running or scheduled to run during that time might fail. For information about viewing scheduled processes, see Before you begin: in this section.**

While you can use the Customization Set Migration dialog to move customizations and extensions from any source environment to any target environment, you should always perform your customizations and extensions in a full test environment and use the Customization Set Migration dialog to move these changes to a production environment, as illustrated in Figure 2–1 and Figure 2–2. Because customization set migration does not provide a merge capability, never customize or extend a production environment unless you are performing steps in this section that explicitly instruct you to manually move certain customizations and extensions. When you import a customization set, the rows in the database that are not preconfigured are updated if a matching record exists. Otherwise a record is inserted.

Note: The customization set does not include all deletions. For example, the set does not include the removal of a customization document using the Manage Customizations dialog. After you import a customization set into the target environment, you must examine the environment for any deletions that you must make manually.

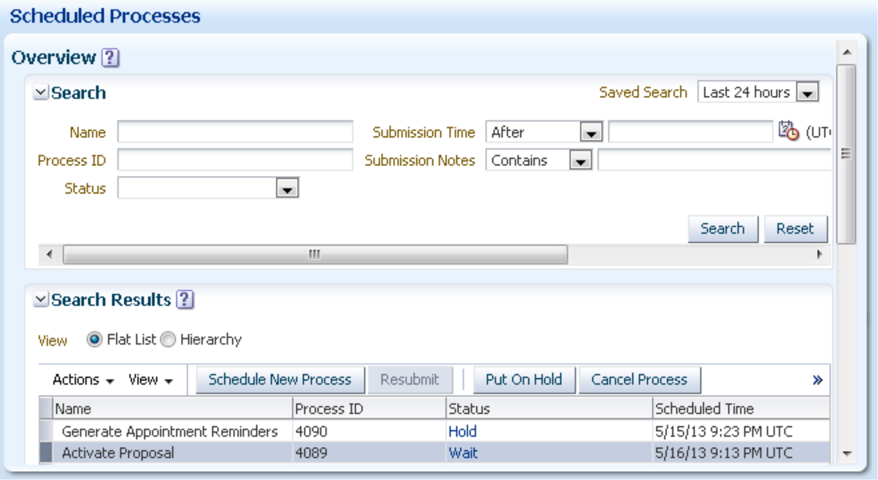
Before you begin:

You will need to do the following before you can begin migrating a customization set:

- Ensure that the source and target Oracle Fusion Applications environments are of the same release and that the same standard and one-off patches have been applied to both environments. For Oracle Cloud implementations, log a service request using My Oracle Support at `https://support.oracle.com` and request a **production-to-test movement**, a reprovisioning of the test environment, or an upgrade of the production environment, whichever is appropriate.

- If you are moving Oracle Fusion CRM customizations, ensure that there are no Oracle Enterprise Scheduler jobs that are running or scheduled to run during the time that customizations are being applied to the target environment. The apply process takes approximately 25 minutes. To view the Oracle Enterprise Scheduler jobs choose **Tools** from the **Navigator** menu, and then choose **Scheduled Processes**. If there is a job that is running or is scheduled to run during the apply process, either cancel the job or put it on hold, as shown in Figure 2–12.

Figure 2–12 Scheduled Processes Dialog



Saved Search: Last 24 hours

Submission Time: After

Submission Notes: Contains

Name	Process ID	Status	Scheduled Time
Generate Appointment Reminders	4090	Hold	5/15/13 9:23 PM UTC
Activate Proposal	4089	Wait	5/16/13 9:13 PM UTC

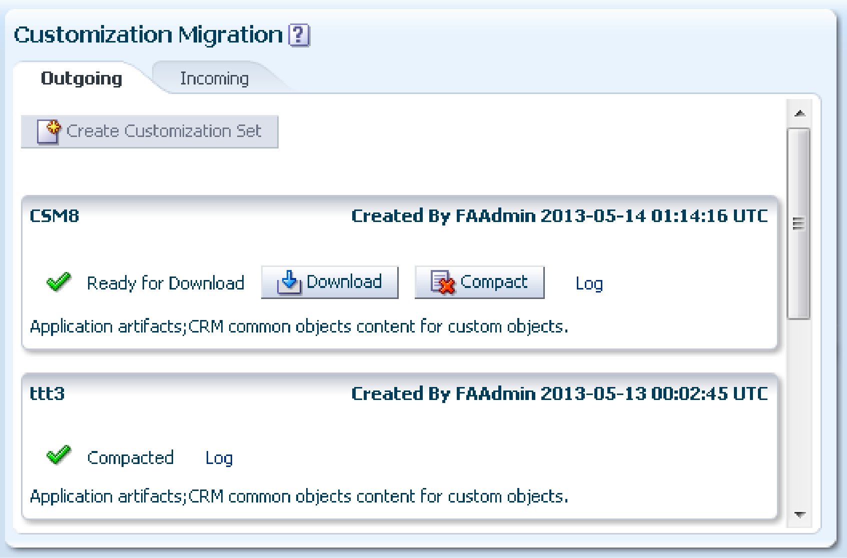
- To prevent the movement of incomplete customizations, perform the following tasks:
  - Ensure that all Application Composer, Page Composer, and JDeveloper customizations are made in sandboxes and are not published until they are complete, as described in Section 2.2, "Using the Sandbox Manager."
  - Ensure that all other customizations and extensions, such as customizations made using the Manage Menu Customizations task, the Manage Standard Lookups task, and Oracle Authorization Policy Manager, are complete.
  - Do not allow customizations in the source environment during the export process.
  - Do not allow users to publish sandboxes during the export process.
- Ensure that users do not make customizations in the target environment during the apply process because the results of doing so are unpredictable. As stated previously, with the exception of customizations that must be moved manually, never make customizations in production environments. However, if you must make emergency customizations to the production environment, you must also make the same customizations to the test environment to ensure that they are included in the next customization set migration.
- Ensure that you have been granted access to the FND_CUSTOMIZATION_SET_MANAGEMENT_DUTY role, which enables you to access the Customization Set Migration dialog. Contact your security administrator for details.
- For the following reasons, ensure that you apply and restore customizations only when few people are signed in:
  - The users might encounter the problems listed in the warning at the beginning of this section.

- – Anyone who is signed in when the migration occurs must sign out and sign back in to see the changes that are made by the process.

To migrate customizations using the Customization Set Migration dialog:

1. From the source environment, choose **Customization Migration** from the Tools section in the **Navigator** menu.
2. Click the **Outgoing** tab.
3. If the **Create Customization Set** button is disabled as shown in Figure 2–13, click the **Compact** button for the previous customization set. This removes the temporary files that are on the server from the previous customization set creation. You will not be able to create a customization set until the previous set has been compacted

Figure 2–13 Customization Set Migration Outgoing Tab



CSM8 Created By FAAdmin 2013-05-14 01:14:16 UTC

✓ Ready for Download

Application artifacts;CRM common objects content for custom objects.

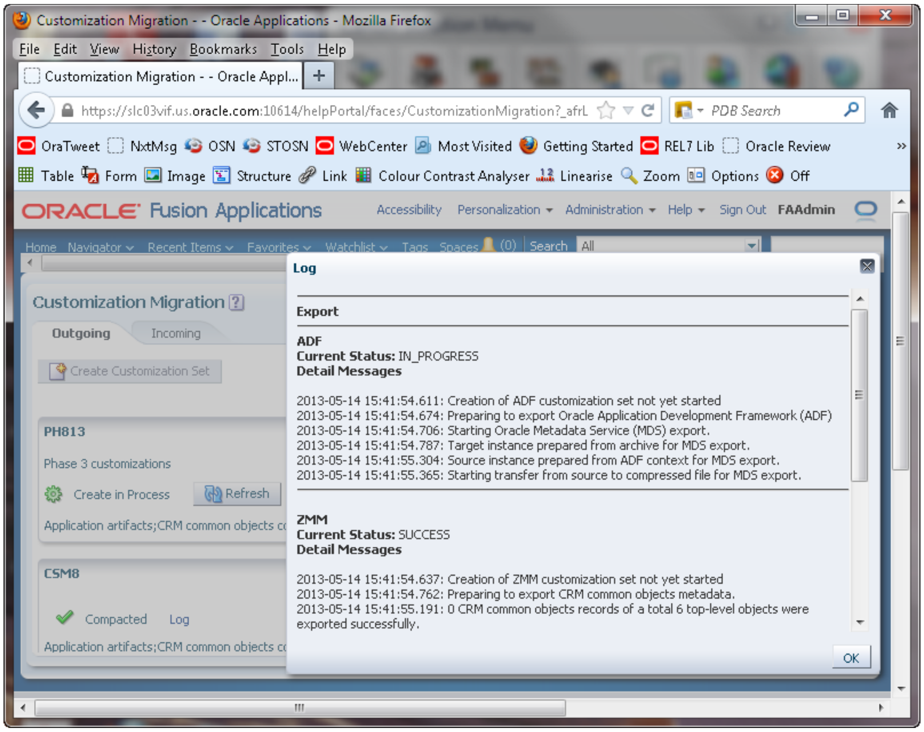
ttt3 Created By FAAdmin 2013-05-13 00:02:45 UTC

✓ Compacted

Application artifacts;CRM common objects content for custom objects.

4. From the **Outgoing** tab of the Customization Set Migration dialog, click **Create Customization Set**.
5. Provide a name for the customization set.
6. Optionally, type a description of the set.
7. Click **Save and Close**.
8. (Optional) Click **Log** to view the process status details as shown in Figure 2–14.

Figure 2–14 Customization Set Migration Create Process Log


Customization Migration - - Oracle Applications - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Customization Migration - - Oracle Appl...
https://slc03vif.us.oracle.com:10614/helpPortal/faces/CustomizationMigration?_afrL
PDB Search
OraTweet NxtMsg OSN STOSN WebCenter Most Visited Getting Started REL7 Lib Oracle Review
Table Form Image Structure Link Colour Contrast Analyser Linearise Zoom Options Off
ORACLE Fusion Applications
Accessibility Personalization Administration Help Sign Out FAAdmin
Home Navigator Recent Items Favorites Watchlist Tags Spaces (0) Search All
Customization Migration
Outgoing Incoming
Create Customization Set
PH813
Phase 3 customizations
Create in Process Refresh
Application artifacts;CRM common objects c
CSM8
Compacted Log
Application artifacts;CRM common objects c
OK

Log

Export

ADF
Current Status: IN_PROGRESS
Detail Messages

```
2013-05-14 15:41:54.611: Creation of ADF customization set not yet started
2013-05-14 15:41:54.674: Preparing to export Oracle Application Development Framework (ADF)
2013-05-14 15:41:54.706: Starting Oracle Metadata Service (MDS) export.
2013-05-14 15:41:54.787: Target instance prepared from archive for MDS export.
2013-05-14 15:41:55.304: Source instance prepared from ADF context for MDS export.
2013-05-14 15:41:55.365: Starting transfer from source to compressed file for MDS export.
```

ZMM
Current Status: SUCCESS
Detail Messages

```
2013-05-14 15:41:54.637: Creation of ZMM customization set not yet started
2013-05-14 15:41:54.762: Preparing to export CRM common objects metadata.
2013-05-14 15:41:55.191: 0 CRM common objects records of a total 6 top-level objects were
exported successfully.
```

9. Periodically, click **Refresh** to view the current status of the set creation. When the status changes to Ready for Download, proceed to the next step.

   This process takes approximately 15 minutes. If, when you click **Refresh**, the export has not progressed for over one hour, a dialog box appears with the following message. If there is a reason for the stalled process, such as a slow server, click **Continue**. Otherwise, click **Restart** to restart the export process.

   ```
   The process might have stalled due to unexpected issues, possibly with the
   server. Do you want to restart the process or continue with this process?
   ```

   Note: The process runs asynchronously. You can exit the dialog and return to it at a later time.

   ---

10. When the customization set is ready for download, click **Download**, specify the name and location for the file that will be created, and click **Save**.
11. After the download file is successfully created on your local file system, click **Compact** to remove the temporary files that were created on the server.
12. Open the Customization Set Migration dialog in the target environment.
13. From the **Incoming** tab, click **Browse**, specify the name and location of the customization set file, and click **Open**.

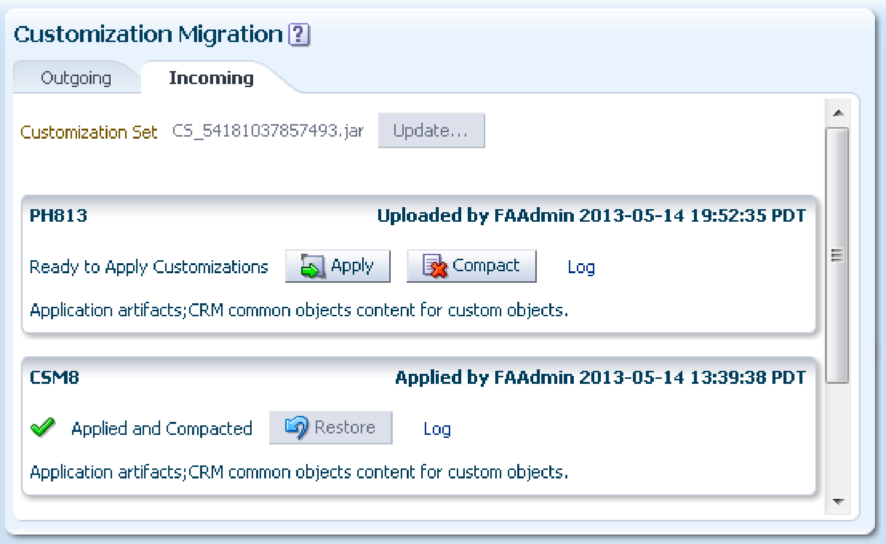
    Note: If the **Browse** button is not enabled, you must first apply the previously uploaded customization set, or, if you do not intend to apply the set, you must first compact it to enable the **Browse** button.

WARNING: **Always upload the most recent customization set from the source environment. Never upload a customization set into the same environment from which it was created.**

---

14. When the status for the customization set is Ready to Apply Customizations as shown in Figure 2–15, click **Apply**.

Figure 2–15 Customization Set Migration Incoming Tab



Customization Set CS_54181037857493.jar

PH813 Uploaded by FAAdmin 2013-05-14 19:52:35 PDT

Ready to Apply Customizations

Application artifacts;CRM common objects content for custom objects.

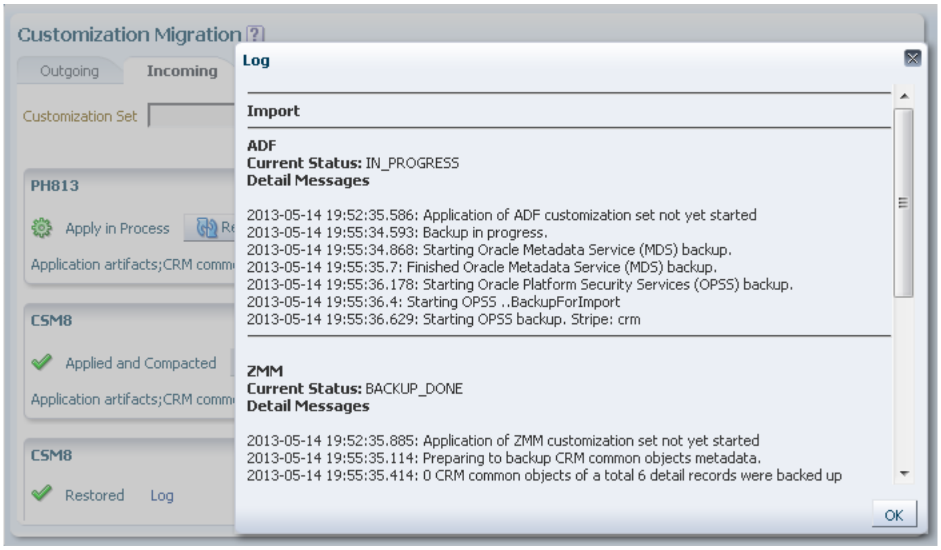
CSM8 Applied by FAAdmin 2013-05-14 13:39:38 PDT

Applied and Compacted

Application artifacts;CRM common objects content for custom objects.

15. (Optional) Click **Log** to view the process status details as shown in Figure 2–16.

Figure 2–16 Customization Set Migration Apply Process Log



```
Import

ADF
Current Status: IN_PROGRESS
Detail Messages

2013-05-14 19:52:35.586: Application of ADF customization set not yet started
2013-05-14 19:55:34.593: Backup in progress.
2013-05-14 19:55:34.868: Starting Oracle Metadata Service (MDS) backup.
2013-05-14 19:55:35.7: Finished Oracle Metadata Service (MDS) backup.
2013-05-14 19:55:36.178: Starting Oracle Platform Security Services (OPSS) backup.
2013-05-14 19:55:36.4: Starting OPSS ..BackupForImport
2013-05-14 19:55:36.629: Starting OPSS backup. Stripe: crm

ZMM
Current Status: BACKUP_DONE
Detail Messages

2013-05-14 19:52:35.885: Application of ZMM customization set not yet started
2013-05-14 19:55:35.114: Preparing to backup CRM common objects metadata.
2013-05-14 19:55:35.414: 0 CRM common objects of a total 6 detail records were backed up
```

16. Periodically, click **Refresh** to view the current status of the apply action. When the apply process has completed, proceed to the next step.

    This process takes approximately 25 minutes. If, when you click **Refresh**, the process has not progressed for over one hour, a dialog box appears with the following message. If there is a reason for the stalled process, such as a slow server, click **Continue**. Otherwise, click **Restart** to restart the apply process.

```
The process might have stalled due to unexpected issues, possibly with the
server. Do you want to restart the process or continue with this process?
```

> **Tip:** The process runs asynchronously. You can exit the dialog and return to it at a later time.

> **Note:** For Oracle Cloud implementations, if problems occur during an apply action, log a service request using My Oracle Support at `https://support.oracle.com`. For non-Cloud implementations, if problems occur, see Section A.2.3, "Troubleshooting Customization Set Migration Issues" for information about diagnosing issues.

17. Access the target environment and examine the environment for any deletions that you must make manually.

18. Deploy all flexfields that have a status of Patched as described in Section 4.7, "Deploying Flexfield Configurations."

    > **Tip:** Click the header in the **Status** column to sort the flexfields by status.

    In non-Cloud environments, you can also use the `deployPatchedFlex` WLST command as described in the "Deploying Flexfields Using the Command Line: Explained" section in the *Oracle Fusion Applications Common Implementation Guide*.

19. Complete the following substeps to send the newly created and newly updated Oracle Social Network definitions to the Oracle Social Network server:

    a. Access the Manage Oracle Social Network Objects task by choosing **Setup and Maintenance** from the **Administration** menu in the global area of Oracle Fusion Applications and searching for the task.

    b. For each object that was created or updated by the apply process, if its **Enabled** value is other than `No`, trigger the sending of its definition to the Oracle Social Network server by changing its **Enabled** value, resetting it back to the original **Enabled** value, and then saving the change. For example, if the **Enabled** value is `Manual`, click **Disable Object**, click **Enable Object**, select **Manual**, click **OK**, and then click **Save**.

20. If you applied Oracle Fusion CRM application customizations, you must submit all custom subject areas for publishing as described in the "Publishing Custom Subject Areas: Explained" section in the *Oracle Fusion Applications CRM Extensibility Guide*. Note that you must submit all custom subject areas, including those that were previously published after applying earlier customization migration sets.

21. If you applied Oracle Fusion CRM application customizations, perform the following tasks from Application Composer to complete the movement of Oracle Fusion CRM customizations and extensions:

    - Generate the artifacts that are required to register the migrated extensions by clicking **Import and Export** in the Common Setup pane and then clicking **Generate**. This makes the migrated extensions available for importing and exporting. For more information, see the "Importing and Exporting Custom

Objects: Explained" section in the *Oracle Fusion Applications CRM Extensibility Guide*.

- Create the same email templates that were created in the source environment. For information about creating email templates, see the "E-Mail Templates: Explained" section in the *Oracle Fusion Applications CRM Extensibility Guide*.
- From the **Common Setup** region, click **Web Services** and complete the following substeps for every secured SOAP web service connection that uses either of the following authentication schemes to manually migrate the user names and passwords:
  - Invoke with separate user credentials over SSL
  - Invoke with separate user credentials and message protection

  a. Make a note of the name and WSDL URL for the web service.

  b. Delete the secured web service.

  c. Click the **Create** icon to re-create the secured web service that you just deleted.

  d. In the Create SOAP Web Service Connection dialog, type the name and WSDL URL for the service that you just deleted. These values must match the values that you noted in Step a.

  e. Click **Read WSDL**.

  f. Ensure that the appropriate authentication scheme is selected.

  g. Click the **New** icon for the **Credential Key** field.

  h. In the Create Key dialog, type the credential key, user name, and password, and then click **OK**.

  i. Click **Save and Close**.

22. If attachments were enabled on runtime pages using Application Composer, complete the following steps to enable attachments for those pages in the target environment:

    a. From the source environment, choose **Setup and Maintenance** from the **Navigator** menu.

    b. In the **Tasks** region, click **Manage Implementation Projects**.

    c. On the Manage Implementation Projects page, choose **Create** from the **Actions** menu.

    d. On the Enter Basic Information page, specify the project details.

    You can accept the defaults that are supplied. If you change the **Code** value, you must specify a unique code. Use the default value for the **Start Date** and leave the **Finish Date** blank. The **Assigned To** user must have access to Customization Set Migration.

    Tip: Use the **Name** field to specify a meaningful project name. When you change the name, the code and description fields change automatically.

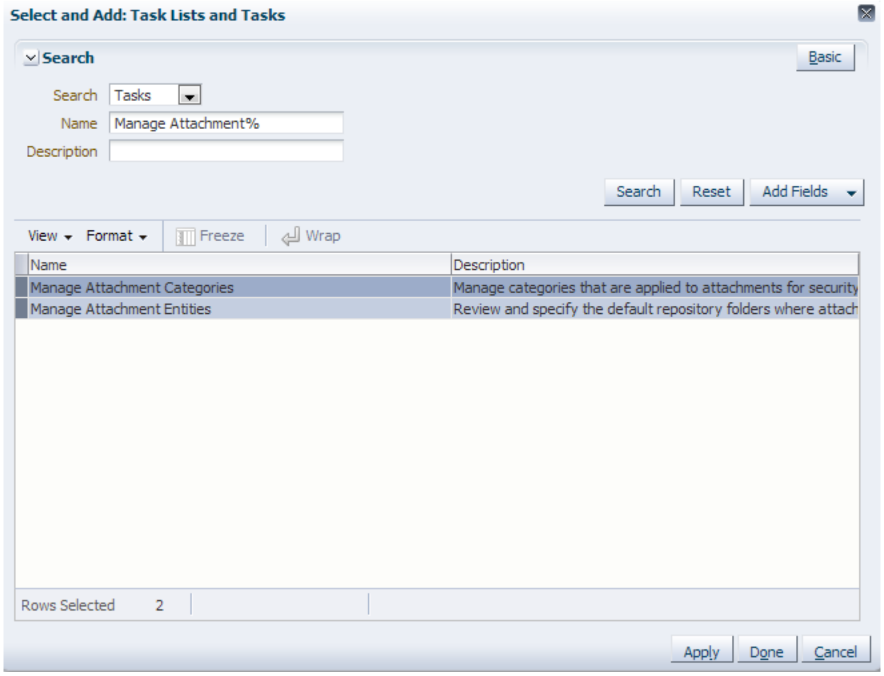
    e. Click **Save and Open Project**.

    f. On the Implementation Project page, choose **Select and Add** from the **Actions** menu.

g. Select **Tasks** from the **Search** dropdown list.

h. Search for and select the **Manage Attachment Categories** task and the **Manage Attachment Entities** task, as shown in Figure 2–17, and then click **Apply**.

Figure 2–17 Selection of Manage Attachment Tasks



Search: Tasks

Name: Manage Attachment%

Description:

Name	Description
Manage Attachment Categories	Manage categories that are applied to attachments for security
Manage Attachment Entities	Review and specify the default repository folders where attach

Rows Selected 2

i. Click **Done** and click **Done** again.

j. In the **Tasks** region on the Setup and Maintenance page, click **Manage Configuration Packages**.

k. On the Manage Configuration Packages page, choose **Create** from the **Actions** menu.

l. From the **Name** dropdown list in the **Source Implementation Project** region on the Create Configuration Package page, select the implementation project that you just created.

m. Click **Next** and click **Next** again.

n. On the Create Configuration Package: Schedule and Notifications page, click **Submit**.

o. When the Warning dialog appears, click **Yes** to continue the export process.

p. Periodically, click the **Refresh** icon to display the current status.

q. After the process has completed, download the configuration package.

r. On the target environment, create an implementation project that is similar to the one that you created on the source environment.

s. Create a configuration project similar to the one that you created on the source environment. Do not submit the project. Click **Save and Close**.

t. On the Manage Configuration Packages page, select the configuration package that you just created, and click **Upload**.

u. Select the configuration package that you just downloaded from the source environment and click **Open**.

v. Click **Details** and then click **Submit**.

w. Select the configuration package and click **Import Setup Data**.

x. Click **Submit**.

y. Periodically, click the **Refresh** icon to display the current status.

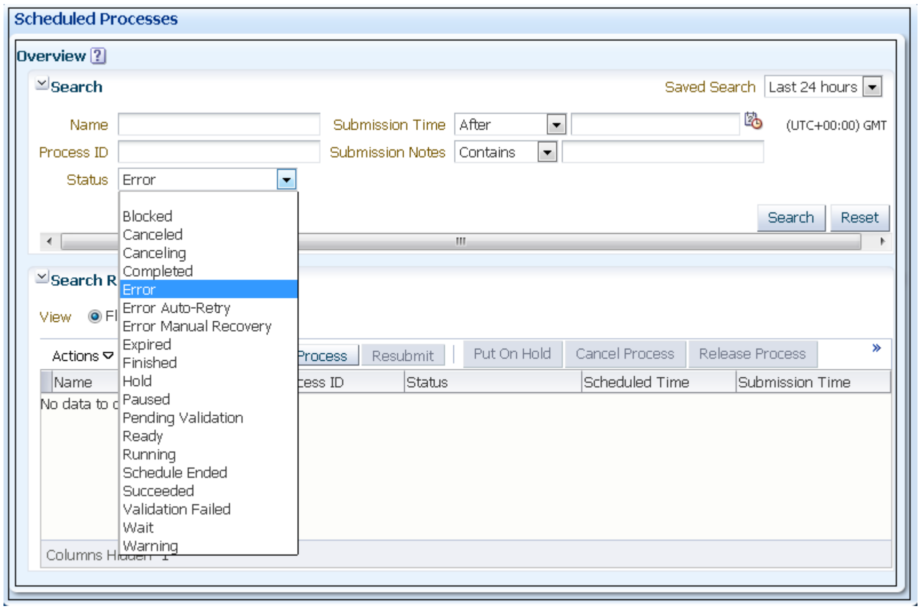
z. After the job completes, click **Done**.

23. (Optionally) From the **Navigator** menu, choose **Tools** and then choose **Scheduled Processes** to identify and reschedule any Oracle Fusion CRM processes that failed during the apply process. Look for processes with the following statuses:

    - Blocked
    - Error
    - Error Auto-Retry
    - Error Manual Recovery
    - Paused
    - Validation Failed

    As shown in Figure 2–18, you can choose a value from the **Status** field in the **Search** region to filter the search results by status. Alternatively, you can click the **Status** column header in the **Search Results** table to sort the search results by status.

Figure 2–18 Searching Scheduled Processes by Status



Scheduled Processes

Overview

Search

Saved Search: Last 24 hours

Name:

Submission Time: After (UTC+00:00) GMT

Process ID:

Submission Notes: Contains

Status: Error

No data to d

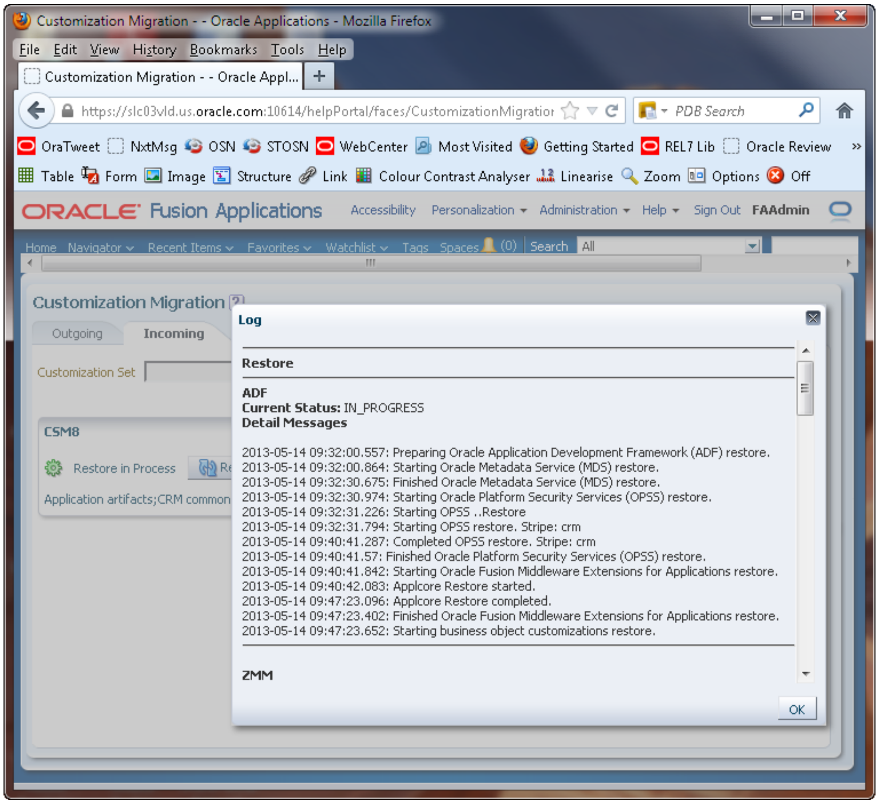
Columns Hidden 1

24. Perform functional testing to verify the changes. If testing exposes problems with the customizations, such as importing more than you intended, or the changes were not what you expected, access the customization set in the **Incoming** tab of the Customization Set Migration dialog, and click **Restore** to revert back to the state before the customization set was applied. Skip the next step.

WARNING: All user personalizations that are performed after a customization set is applied are lost when you perform a restore on that customization set.

Figure 2–19 shows the Log dialog box for a restore that is in progress.

Figure 2–19 Customization Set Migration Restore Process Log


Customization Migration - - Oracle Applications - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Customization Migration - - Oracle Appl...
https://slc03vld.us.oracle.com:10614/helpPortal/faces/CustomizationMigratior
PDB Search
OraTweet NxtMsg OSN STOSN WebCenter Most Visited Getting Started REL7 Lib Oracle Review
Table Form Image Structure Link Colour Contrast Analyser Linearise Zoom Options Off
ORACLE Fusion Applications
Accessibility Personalization Administration Help Sign Out FAAdmin
Home Navigator Recent Items Favorites Watchlist Tags Spaces (0) Search All
Customization Migration
Outgoing Incoming
Customization Set
CSM8
Restore in Process
Application artifacts;CRM common
OK

Log

Restore

ADF
Current Status: IN_PROGRESS
Detail Messages

```
2013-05-14 09:32:00.557: Preparing Oracle Application Development Framework (ADF) restore.
2013-05-14 09:32:00.864: Starting Oracle Metadata Service (MDS) restore.
2013-05-14 09:32:30.675: Finished Oracle Metadata Service (MDS) restore.
2013-05-14 09:32:30.974: Starting Oracle Platform Security Services (OPSS) restore.
2013-05-14 09:32:31.226: Starting OPSS ..Restore
2013-05-14 09:32:31.794: Starting OPSS restore. Stripe: crm
2013-05-14 09:40:41.287: Completed OPSS restore. Stripe: crm
2013-05-14 09:40:41.57: Finished Oracle Platform Security Services (OPSS) restore.
2013-05-14 09:40:41.842: Starting Oracle Fusion Middleware Extensions for Applications restore.
2013-05-14 09:40:42.083: Applcore Restore started.
2013-05-14 09:47:23.096: Applcore Restore completed.
2013-05-14 09:47:23.402: Finished Oracle Fusion Middleware Extensions for Applications restore.
2013-05-14 09:47:23.652: Starting business object customizations restore.
```

ZMM

25. Broadcast information to the users that they must sign out and sign back in to see the changes.

2.3.2 Using Oracle Fusion Functional Setup Manager to Move Customizations

You can use the Functional Setup Manager to move common reference objects, such as application data security policies, application extensible flexfields, application descriptive flexfields, and application value sets from one Oracle Fusion Applications environment to another. For more information, see "Importing and Exporting Setup Data" in the *Oracle Fusion Applications Common Implementation Guide*.

2.3.3 Using the Manage Customizations Dialog to Download Customizations

The Manage Customizations dialog enables you to download customization files for a given page. You can use this downloaded file for diagnosing customization issues.

To download customizations using the Manage Customizations dialog:

1. Go to the page for which you wish to download or upload customizations.
2. Select the page, task flow, or page fragment that contains the customizations.

3. From the **Administration** menu in the global area of Oracle Fusion Applications, choose **Manage Customizations**.

   Tip: After you are in the Manage Customizations dialog, you can change the page, task flow, or page fragment for which you are viewing customizations using the **Search** field.

4. To download a file, click the **Download** link for the corresponding customization. The file will be downloaded to your local machine.

   Note: To download all customizations of the page for all layers, click the **Download Customizations for All Layers** link, located at the bottom of the window. This will download the `AllCustomization.zip` file, which contains all the customization XML files for the page.

2.3.4 Using WLST Commands to Move Customizations

Note: WLST is not available in Oracle Cloud implementations.

You can use WLST commands to upload and download customization files for one page, multiple pages, the navigator menu, and Oracle Application Development Framework (ADF) Business Components.

Note: You can also use Fusion Applications Control to move customizations as described in Section 2.3.5, "Using Fusion Applications Control to Move Customizations."

Example 2–1 shows how to export all customizations for a specific application.

Example 2–1 WLST Command to Export All Customizations for an Application

```
exportMetadata (application='application name',
server='server name', docs='/**',
excludeBaseDocs='true',
toLocation='temp location')
```

Example 2–2 shows how to export the site layer customizations of the navigator menu.

Example 2–2 WLST Command to Export Site Layer Customizations of the Navigator Menu

```
exportMetadata (application='application name',
server='server name', docs='oracle/apps/menu/mdssys/Site/SITE/root_menu.xml.xml',
toLocation='temp location')
```

> **Tip:** If you are not sure of the document name, append `'/*'` to the path in the `docs` argument to include all customization documents in the directory. Append `'/**'` to the path in the `docs` argument to also include the customization documents in the subdirectories. For example, use `'/oracle/apps/hcm/dashboard/hrSpecialist/publicUi/**'` to import or export all documents under the `publicUi` directory and its subdirectories.

For more information about uploading and downloading customization files, see the "Importing Customizations into Your Application Workspace" section in the *Oracle Fusion Applications Extensibility Guide for Developers*. For information about uploading and downloading the customizations of resources, see the "Translating Resource Bundles from an MDS Repository" section in the *Oracle Fusion Applications Extensibility Guide for Developers*

For more information about the `exportMetadata` command, see the "Managing the Metadata Repository" chapter in the *Oracle Fusion Middleware Administrator's Guide*.

2.3.5 Using Fusion Applications Control to Move Customizations

> **Note:** Fusion Applications Control is not available in Oracle Cloud implementations.

Use Fusion Applications Control to move MDS customizations made to a page or other UI from one environment to another. For more information, see the "Transferring Metadata Using Fusion Middleware Control" section of the *Oracle Fusion Middleware Administrator's Guide*. The referenced procedure describes using Fusion Middleware Control, but also applies to Fusion Applications Control.

Part II

Runtime Customizations and Extensions

Part II contains the following chapters:

3 Customizing Existing Pages

This chapter describes how administrative users can customize pages in Oracle Fusion applications at runtime using Page Composer.

This chapter includes the following sections:

Notes:

- Features and functionality involving Oracle JDeveloper, as described in this chapter, are not available in Oracle Cloud implementations.
- Images in this chapter might not accurately represent the user interface in Oracle Cloud implementations.

3.1 About Customizing Existing Pages

The pages in Oracle Fusion applications provide content and functionality to users that enable them to complete their tasks (for example, learning about a product or service, keeping up with sales data, or submitting an order) as easily and efficiently as possible. Because different users have different needs, Oracle Fusion applications enable you to customize pages to fit those needs using Page Composer. End users can use Page Composer for **personalization** of pages. For example, end users can reorganize content in dashboards to place the content they use most frequently at the top of the page. Administrative users can use Page Composer to **customize** pages in any layer of the application or to customize the **UI Shell template** used by Oracle Fusion Customer Relationship Management (Oracle Fusion CRM) applications—the template used for the base UI for all Oracle Fusion CRM application pages. For example, administrative users can add a logo or contact information to the header and footer of the application.

Most of what you do in Page Composer is perform actions on a UI object, or UI **component**, on a page. Components are grouped together on the page into boxes to simplify management of related components. For example, you might group components relating to a particular project or to a particular subject. The grouped components are referred to as **Child Components**. Page Composer enables you to edit an individual child component or manage all the grouped child components at one time by editing the box containing the child components. For example, you might want all the related child components to have the same style (background color, font, and such) so that users can more easily see their relationship.

Figure 3–1 shows an application page as the user sees it. The outlined components (in red) are grouped together. Figure 3–2 shows the same application page in Page Composer, where you can see the dashed outline of the box containing the grouped components.

Figure 3–1 Components in an Application Page

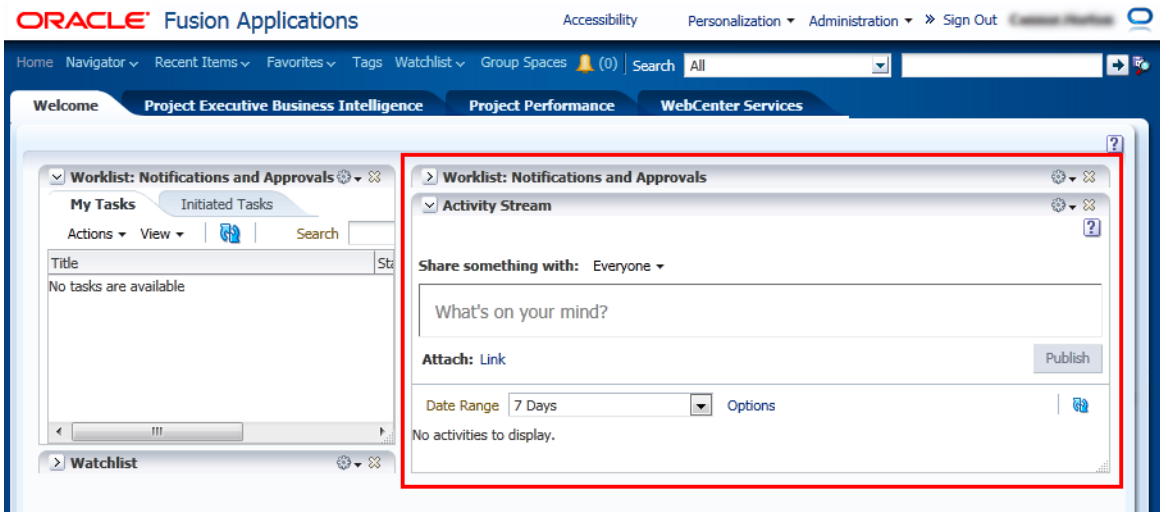
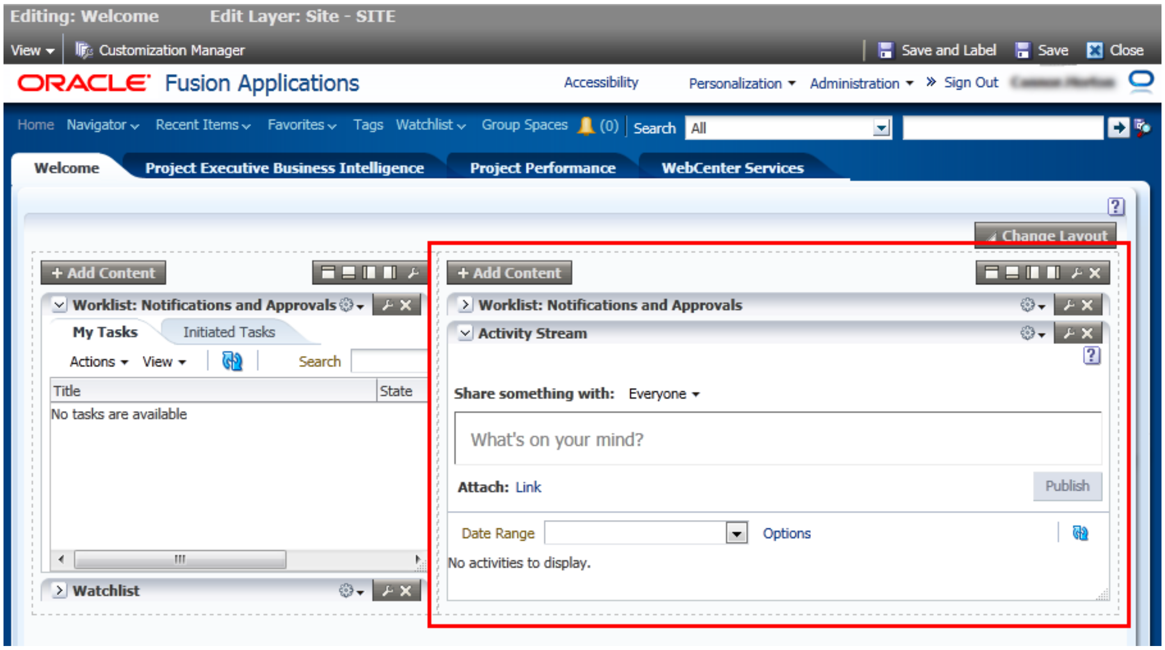


Figure 3–2 Components in Page Composer



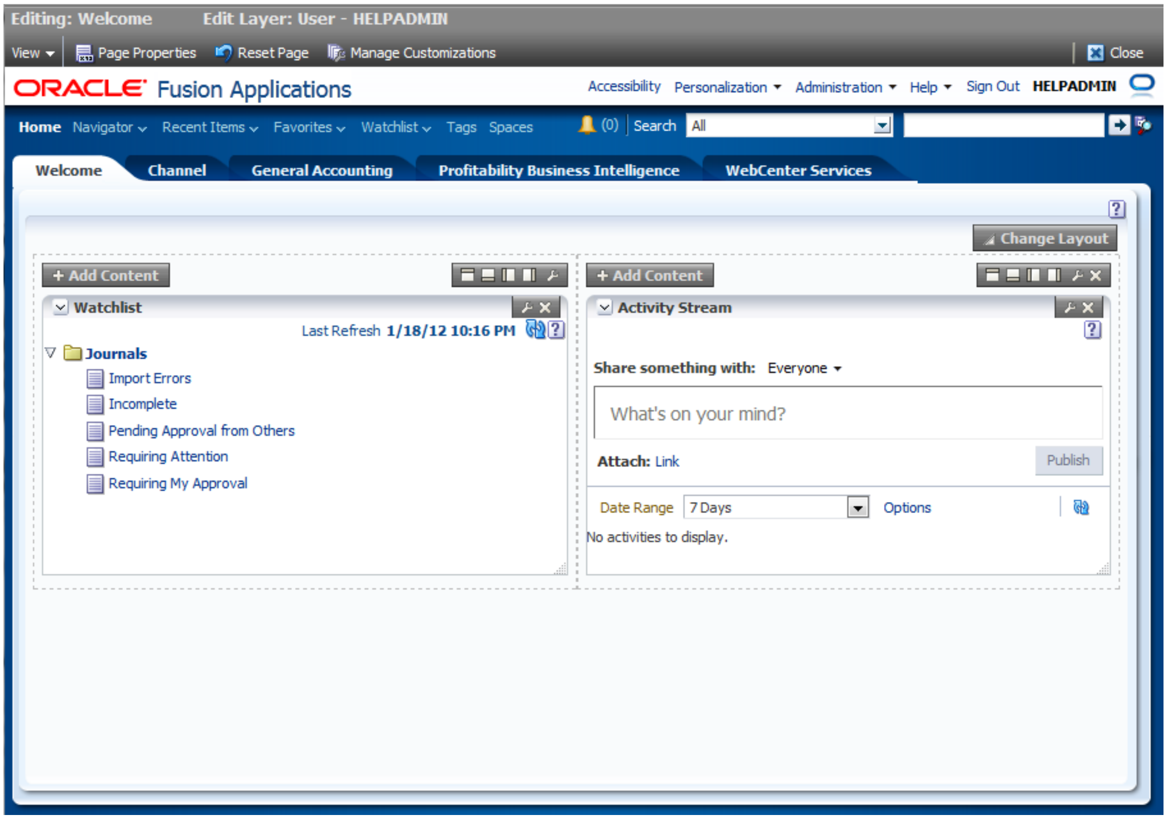
3.1.1 Page Composer User Interface Overview

> **Note:** The user interface and capabilities of Page Composer differ depending on the product family and how Page Composer is accessed. See the product-specific documentation for more information.

Page Composer provides two views for working with page content:

- Design view (Figure 3–3) provides a WYSIWYG rendering of the page and its content, where controls can be selected directly on each component. Design view is available in Oracle Fusion CRM and non-CRM applications, for both personalization and customization.

Figure 3–3 Page Composer's Design view



Last Refresh 1/18/12 10:16 PM

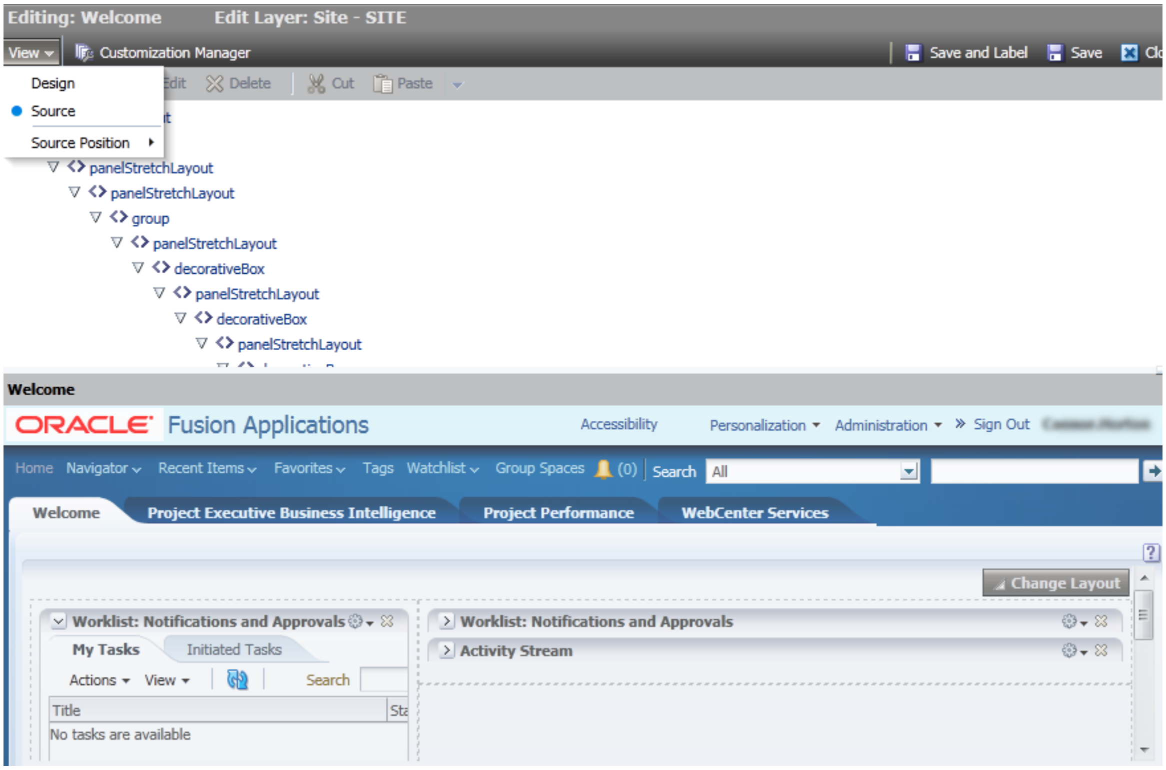
Share something with: Everyone

Date Range 7 Days

No activities to display.

- Source view (Figure 3–4) provides a combined WYSIWYG and hierarchical rendering of page components for advanced users (such as developers or users with knowledge of ADF artifacts, JavaScript, and Expression Language (EL)). Source view is available for non-CRM applications and when customizing the UI Shell template in Oracle Fusion CRM applications. Source view is available only for customization (not personalization).

Figure 3–4 Page Composer's Source view



For more information, see the "Introducing Design View and Source View" section in the *Oracle Fusion Middleware User's Guide for Oracle WebCenter Portal: Spaces*.

3.1.2 Effects of Editing Objects That Display on Multiple Pages

Sometimes objects are included on multiple pages. The effect of changing such an object depends on how the object is included on the page and whether you are working with a Oracle Fusion CRM or non-CRM application.

- If you use Page Composer to make changes to an object that is displayed on multiple pages (and is not part of a shared task flow), then only the object on the page you edit is affected.
- If the object is part of a shared task flow, then the change will affect the object on all pages that include the shared task flow, with one exception. If the shared task flow includes embedded logic that uses data from the page, then the embedded logic might override the changes you make in Page Composer.
- For Oracle Fusion CRM applications, you can change objects globally (on all pages that display the object) by editing them through Oracle Fusion CRM Application Composer. For information, see the "Application Composer: Using the Application Composer" chapter in the *Oracle Fusion Applications CRM Extensibility Guide*.

3.1.3 What You Can Do with Pages at Runtime

Depending on your **role**, the type of page, and the type of application, you can perform different actions in Page Composer.

3.1.3.1 Nondashboard Pages in Oracle Fusion CRM Applications

Sales Administrators, Marketing Operations Managers, and Channel Managers can customize nondashboard pages in the following ways:

- By adding task flows, portlets, documents, layout components, and other objects to a page

  Refer to Task: Add Components to a Page.

- By creating site-level saved searches that can be used by all users

  Refer to Task: Create a Site-Level Saved Search for All Users.

- By changing a page layout

  Refer to Task: Change the Layout of a Page.

- By moving objects on a page

  Refer to Task: Move Components on a Page.

- By changing the display of the object using the tools in the object header (for example, expand or collapse an object; hide, show, or reorder columns in a table)

3.1.3.2 Dashboard Pages in Oracle Fusion CRM Applications

Users can personalize their dashboard pages and administrators can customize site, internal or external, and job-role dashboard pages with the tasks mentioned in Section 3.1.3.1, "Nondashboard Pages in Oracle Fusion CRM Applications" and in the following additional ways:

- By showing or hiding objects on a page

  Refer to Task: Show and Hide Components on a Page.

- By removing objects from a page

  Refer to Task: Delete Components from a Page.

3.1.3.3 Pages in Non-CRM Applications

Administrators can customize pages in non-CRM applications with the tasks mentioned in Section 3.1.3.1, "Nondashboard Pages in Oracle Fusion CRM Applications," Section 3.1.3.2, "Dashboard Pages in Oracle Fusion CRM Applications," and in the following additional ways:

- By customizing a page title

  Refer to Task: Customize a Page Title.

- By customizing a task list menu

  Refer to Task: Customize a Task List Menu.

- By customizing attributes for a **flexfield** on a page

  Refer to Task: Customize Attributes for a Flexfield on a Page.

> **Notes:** Many pages in Oracle Fusion applications include a **flexfield**. Flexfields allow you to add custom attributes to a page. You add flexfields to a page using the Manage Descriptive Flexfields task and the Manage Extensible Flexfields task, then customize the attributes through Page Composer. For more information about adding flexfields to a page, see Chapter 4, "Using Flexfields for Custom Attributes."

- By customizing popup window content

  Refer to Task: Customize Popup Window Content.

- By providing values for the properties associated with pages and the objects the pages contain

  Refer to Section 3.3, "Editing Component Properties in Page Composer."

- By editing the component header and other display options

  Refer to Task: Edit Component Header and Other Display Options.

- By editing the look and feel of the component

  Refer to Task: Edit Component and Content Style.

- By editing the component's unique parameters

  Refer to Task: Edit Component Parameters.

- By rolling back changes to a page or task flow

  Refer to Task: Reset a Page or Task Flow to a Previously Saved Version.

- By wiring pages, task flows, and portlets to each other

  Refer to Task: Allow Certain Component Property Values to Persist Across Sessions.

3.1.3.4 UI Shell Template

Application administrators can customize the UI Shell template used by all pages in the application. See Section 3.4, "Editing the UI Shell Template Used by All Pages."

3.1.4 What You Cannot Do with Pages at Runtime

> **Note:** Features and functionality involving JDeveloper, as described in this section, are not available in Oracle Cloud implementations.

Business users cannot perform the following tasks in a runtime environment with Page Composer:

- Change an object globally (change an object on every page on which it is displayed)

  For Oracle Fusion CRM applications, you can change objects globally by editing them through Application Composer. For information, see the "Application Composer: Using the Application Composer" chapter in the *Oracle Fusion Applications CRM Extensibility Guide*.

- Add a custom attribute to a page using the flexfield feature

If a flexfield exists on a page, you must use the appropriate manage flexfield task to add the custom attributes to the page before working with them in Page Composer. For more information, see Chapter 4, "Using Flexfields for Custom Attributes."

- Make a page personalizable

  To permit end users to edit a page, a developer must use JDeveloper to enable personalization. For more information, see the "Allowing Pages to Be Personalized by End Users in Page Composer" section in the *Oracle Fusion Applications Extensibility Guide for Developers*.

- Edit user interface text that is part of a **skin**

  To edit user interface text that is part of a skin, a developer must use JDeveloper to change the **resource bundle** used by the skin. For more information, see the "How to Apply Skins to Text" section in the *Oracle Fusion Middleware Web User Interface Developer's Guide for Oracle Application Development Framework*.

- Edit text that is part of the embedded help on the page

  To edit text that is part of the embedded help on a page, a developer must use JDeveloper. For more information, see the "Customizing or Adding Static Instructions, In-Field Notes, and Terminology Definitions" section in the *Oracle Fusion Applications Extensibility Guide for Developers*.

- Change ADF task flows

  To edit ADF task flows, a developer must use JDeveloper. For more information, see the "Customizing and Extending Oracle ADF Application Artifacts" chapter in the *Oracle Fusion Applications Extensibility Guide for Developers*.

- Change ADF Business Components objects

  To edit the ADF Business Components objects (for example, to add validation to an ADF Business Components object), a developer must use JDeveloper. For more information, see the "Customizing and Extending Oracle ADF Application Artifacts" chapter in the *Oracle Fusion Applications Extensibility Guide for Developers*.

- Change Oracle Fusion CRM administrative pages (such as Set Up and Look Up Management)

  To edit the Oracle Fusion CRM administrative pages, a developer must use JDeveloper.

- Change mobile web pages (pages built using Trinidad components for mobile clients)

  To edit mobile web pages, a developer must use JDeveloper. For more information, see the "Editing Pages" section in the *Oracle Fusion Applications Extensibility Guide for Developers*.

3.1.5 Before You Begin Customizing Existing Pages

As an administrative user, before you implement customizations in applications, you should be familiar with the Oracle Fusion application architecture that enables customization, as described in Chapter 1, "Customizing and Extending Oracle Fusion Applications." You should also understand the typical workflows for working with runtime customizations, as described in Chapter 2, "Understanding the Customization Development Lifecycle."

You must do the following before you can begin customizing existing pages:

- Confirm that the page is customizable.

  For Oracle Fusion CRM applications:

  – Any user can personalize dashboard pages.

  – Sales Administrator, Marketing Operations Manager, and Channel Manager users can customize transactional pages (such as Edit).

  For HCM applications, all pages are customizable.

  For non-CRM applications, see product-specific documentation in Oracle Fusion Applications Help for information about which pages are customizable.

- Set up a **sandbox**.

  Page Composer uses sandboxes to manage your customizations. For more information, see Section 2.2, "Using the Sandbox Manager."

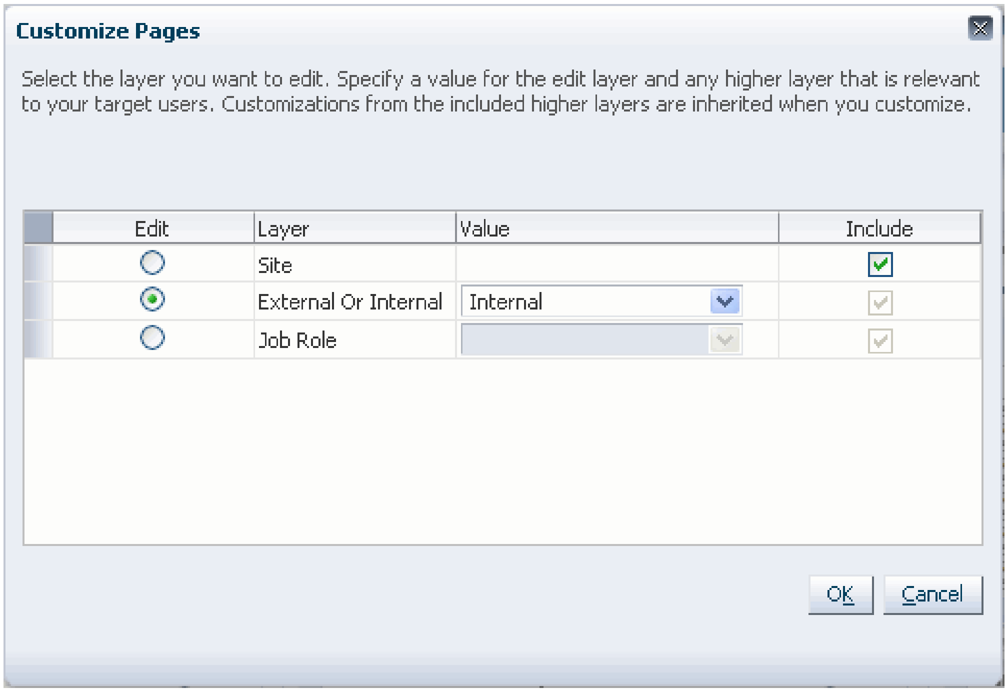
- Access Page Composer.

  – To access Page Composer, go to the page you want to edit, then, from the **Administration** menu in the global area of Oracle Fusion applications, choose **Customize** ***page_name*** **Pages**.

  – To customize existing pages, you must have specific privileges. Contact your security administrator for details.

- If you have more than one layer available for customization, then when you open Page Composer, the UI displays the Layer Picker dialog (Figure 3–5), which you use to specify the layer that you want to edit and its customization context. In the **Edit** column, select the layer you want to edit. The layers that are selected in the **Include** column will inherit any changes you make to the layer you edit.

Figure 3–5 Layer Picker Dialog



Customize Pages

Select the layer you want to edit. Specify a value for the edit layer and any higher layer that is relevant to your target users. Customizations from the included higher layers are inherited when you customize.

Edit	Layer	Value	Include
○	Site		☑
●	External Or Internal	Internal	☑
○	Job Role		☑

For more information about customization layers, including selecting a layer and customization context, see Section 1.2, "Understanding Customization Layers."

3.2 Editing a Page in Page Composer

You can use Page Composer to perform basic editorial tasks, such as editing page components, changing the layout of the page, and the like. The tasks available to you depend on your role, the type of page, and the type of application. For details, see Section 3.1.3, "What You Can Do with Pages at Runtime."

> **Note:** If the object you want to edit is included on multiple pages, then see Section 3.1.2, "Effects of Editing Objects That Display on Multiple Pages," before editing the object.

Task: Add Components to a Page

To add components to a page, in Design view, click **Add Content** at the top of the container component to which you want to add the new component.

For non-CRM pages, you can also add components in Source view (where UI widgets are available). To add components in Source view, you select the container component, then, in the Source view toolbar, click **Add Content**.

In either mode (Design view or Source view), clicking **Add Content** opens the Resource Catalog, enabling you to select from a wide range of task flows, portlets, layout components, and other types of resources. For more information about adding components to a page, see the "Adding a Component to a Page" section in the *Oracle Fusion Middleware User's Guide for Oracle WebCenter Portal: Spaces*.

> **Note:** If you want to display custom attributes, then you may be able to use a flexfield that has been defined for the page. Flexfields are available for many pages in Oracle Fusion applications, except for the Oracle Fusion CRM applications. For more information, see Chapter 4, "Using Flexfields for Custom Attributes."

Task: Create a Site-Level Saved Search for All Users

You can use Page Composer to create a saved search that can be used by all users. To create a site-level (or global) saved search, perform the following steps:

1. In Design view, locate the Search component, and perform a search.
2. In the Search component, click **Save**.
3. In the Create Saved Search dialog, type a name for your saved search and select any search options you want:
   - To make the selected search criteria appear each time a user views the Search component, select **Set as Default**.
   - To make the results of the selected search appear each time a user views the Search component, select **Run Automatically**.

   When you are done, click **OK**.
4. In Page Composer, click **Save**.
5. Log out and log back in to make sure the current list of site-level saved searches gets refreshed.

To delete, rename, or change the search options for a site-level saved search:

1. In Design view, locate the Search component. In the Saved Search drop-down list, select **Personalize**.

2. In the Personalize Saved Searches dialog, select the saved search you want to edit, and make the desired changes:
   - To make the selected search criteria appear each time a user views the Search component, select **Set as Default**.
   - To make the results of the selected search appear each time a user views the Search component, select **Run Automatically**.
   - To display the selected search in the Saved Search drop-down list, select **Show in Search List**.
3. In Page Composer, click **Save**.
4. Log out and log back in to make sure the current list of site-level saved searches gets refreshed.

Task: Change the Layout of a Page

Page layout defines the number, placement, and orientation of content regions on a page. A page's initial layout style is selected when the page is created. Some style selections can be switched even after you have added content to the page, others do not support layout changes after creation.

To change the layout of a page, in Design view, click **Change Layout**. For more information, see the "Changing the Page Layout" section in the *Oracle Fusion Middleware User's Guide for Oracle WebCenter Portal: Spaces*.

Task: Move Components on a Page

To move components on a page, in Design view, drag and drop the component.

For non-CRM pages, you can also cut and paste or drag and drop components in Source view, or you can access the Component Properties for the container component and rearrange the components on the **Child Components** tab. For more information, see the "Rearranging Child Components" section in the *Oracle Fusion Middleware User's Guide for Oracle WebCenter Portal: Spaces*.

Task: Show and Hide Components on a Page

Note: You can perform this task only on dashboard pages in Oracle Fusion CRM applications, on nondashboard pages in Oracle Fusion CRM applications if you have the proper job role, or in non-CRM applications.

You can control whether to show or hide a component on a page. For example, you have a list of checkboxes, and if a user selects checkbox B, you want a button to be displayed. You could set the Show Component property on the button to be an EL expression that says `#{if checkboxB.selected = true}`, meaning that if the checkbox B is selected, then display the button.

There are two ways to hide a component—hide the individual component or hide the box containing the component.

To hide an individual component, click the Edit icon in the component header, click the **Display Options** tab, deselect the **Show Component** box, and click **OK**. For more information, see the "Working with Component Display Options" section in the *Oracle Fusion Middleware User's Guide for Oracle WebCenter Portal: Spaces*. You can instead, click the Edit icon on the containing box's toolbar, then, on the **Child Components** tab, deselect the box next to the component you want to hide, and click **OK**. For more information, see the "Hiding and Showing Child Components" section in the *Oracle Fusion Middleware User's Guide for Oracle WebCenter Portal: Spaces*.

To hide the box and all its child components, click the Edit icon in the box's header, click the **Display Options** tab, deselect the **Show Component** box, and click **OK**. For more information, see the "Working with Component Display Options" section in the *Oracle Fusion Middleware User's Guide for Oracle WebCenter Portal: Spaces*.

Task: Delete Components from a Page

> **Note:** You can perform this task only on dashboard pages in Oracle Fusion CRM applications, on nondashboard pages in Oracle Fusion CRM applications if you have the proper job role, or in non-CRM applications.

> **WARNING: Delete a component only if you are positive that no other components or processes are dependent on the component you delete. If you are uncertain about any dependencies, then hide the component instead of deleting it.**

If you are sure no other components or processes are dependent on a component, you can delete the component from a page by clicking the Delete icon in the component header.

> **Note:** Some components might not be able to be deleted, such as mandatory or indexed fields or components that are installed as part of the Oracle Fusion applications.

If you delete a box, all of the child components—any components contained in the box—are also deleted.

For more information about deleting components, see the "Deleting a Component from a Page" section in the *Oracle Fusion Middleware User's Guide for Oracle WebCenter Portal: Spaces*.

Task: Customize a Page Title

> **Note:** You can perform this task only in non-CRM applications.

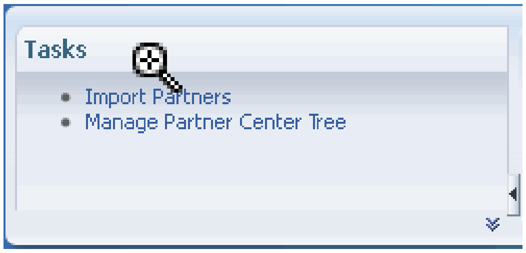
The page title is specified in three places:

- The Page Title property on the Task List Properties tab in Page Composer, which affects the browser title
- The Label property on the Task List Task Properties tab in Page Composer, which affects the task lists menu entry, page heading, and tab title
- The Label property for the navigator menu item in the Manage Menu Customizations task, which affects the navigator menu entry

You must change all three entries to change the page title. To change the first two entries in Page Composer, complete the following steps. To change the navigator menu entry, see Chapter 5, "Customizing Menus."

1. If Source view is not already displayed, switch to Source view. From the **View** menu, select **Source**.
2. In the Design region (at the bottom of the page), click the task list as shown in Figure 3–6.

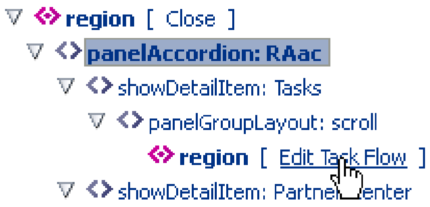
Figure 3–6 Selecting a Task List in the Design Region



3. When asked to confirm that you want to edit the task flow, click **Edit**.

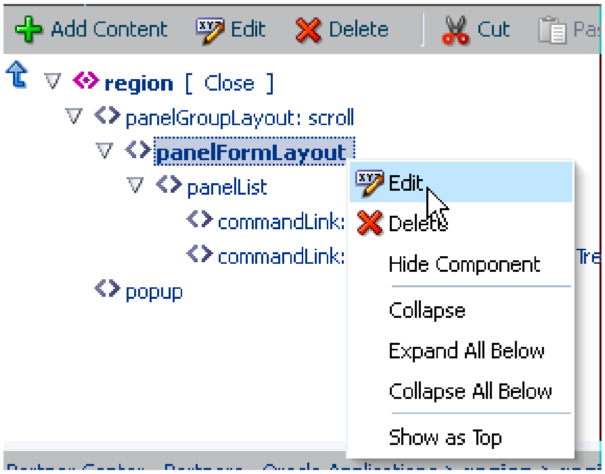
   This selects the task list entry in the Source region.
4. In the Source region, click **Edit Task Flow** next to the task list as shown in Figure 3–7.

Figure 3–7 Opening the Task Flow Editor



5. Again, when asked to confirm that you want to edit the task flow, click **Edit**.
6. In the Source region, right-click the `panelFormLayout` node, and select **Edit** as shown in Figure 3–8.

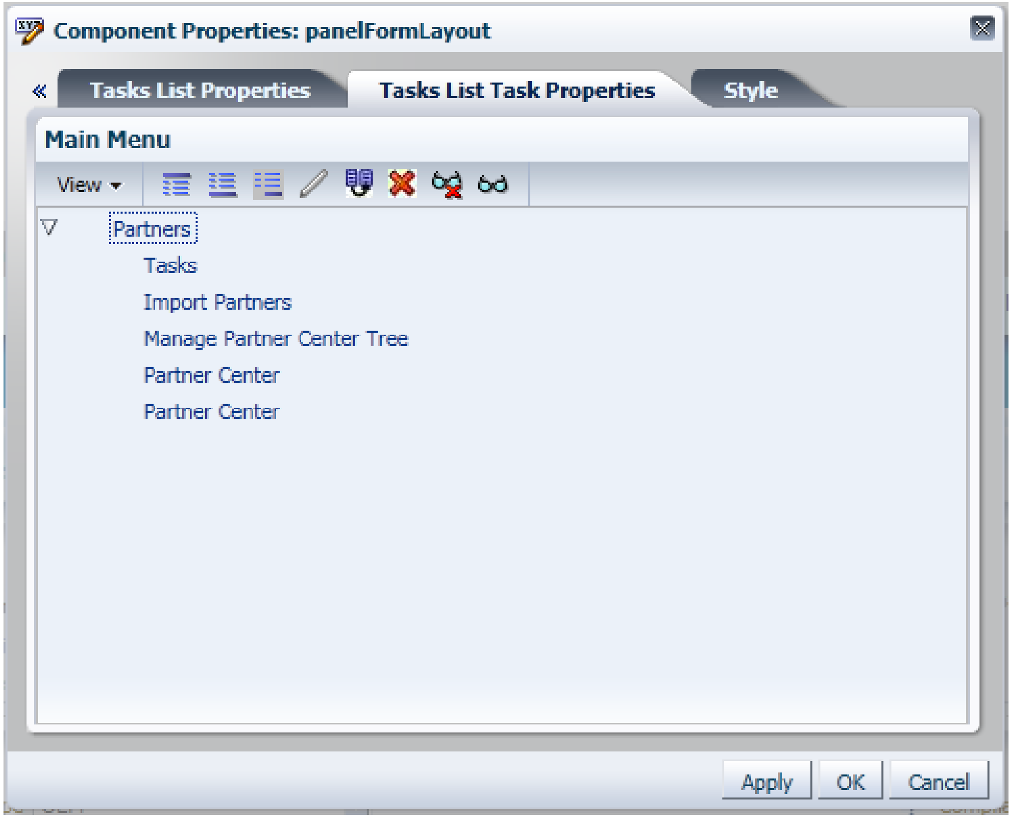
Figure 3–8 Editing the panelFormLayout Node



The Component Properties dialog for `panelFormLayout` is displayed.

7. On the Tasks List Properties tab, in the **Page Title** box, type the title for the page, then click **OK** to save your changes and close the Component Properties dialog.
8. Click the **Tasks List Task Properties** tab.
9. Expand the tree to display the items in the task list as shown in Figure 3–9.

Figure 3–9 Component Properties — Tasks List Task Properties Task List Items



10. In the tree hierarchy, click the first child item ("Tasks" in Figure 3–9), then click the Edit icon in the toolbar.
11. In the **Label** box, type the title for the page, then click **OK** to save your changes and close the Component Properties dialog.

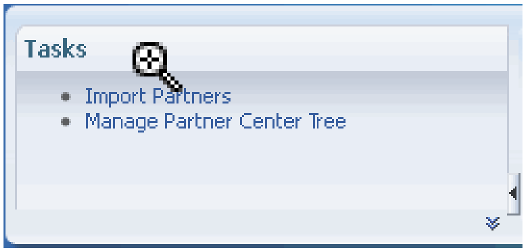
Task: Customize a Task List Menu

Note: You can perform this task only in non-CRM applications.

Task lists enable you to provide links to task flows in your application or web pages outside your application. For example, you can add links to frequently used task flows, so that users can quickly perform the most common tasks.

1. If Source view is not already displayed, switch to Source view. From the **View** menu, select **Source**.
2. In the Design region (at the bottom of the page), click the task list as shown in Figure 3–10.

Figure 3–10 Selecting a Task List in the Design Region

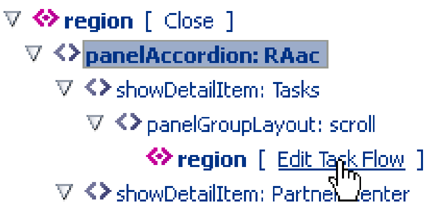


3. When asked to confirm that you want to edit the task flow, click **Edit**.

   This selects the task list entry in the Source region.

4. In the Source region, click **Edit Task Flow** next to the task list as shown in Figure 3–11.

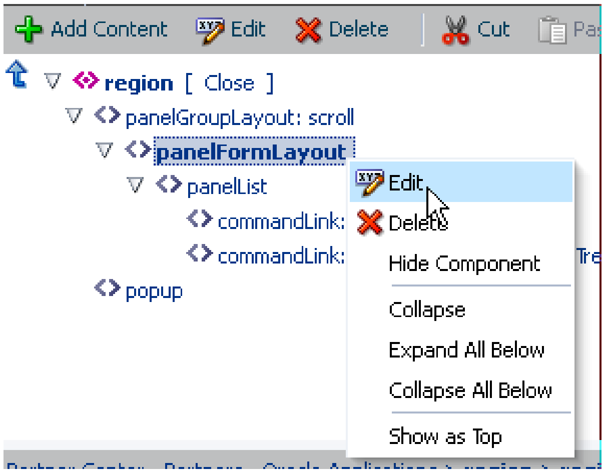
Figure 3–11 Opening the Task Flow Editor



5. Again, when asked to confirm that you want to edit the task flow, click **Edit**.

6. In the Source region, right-click the `panelFormLayout` node, and select **Edit** as shown in Figure 3–12.

Figure 3–12 Editing the panelFormLayout Node

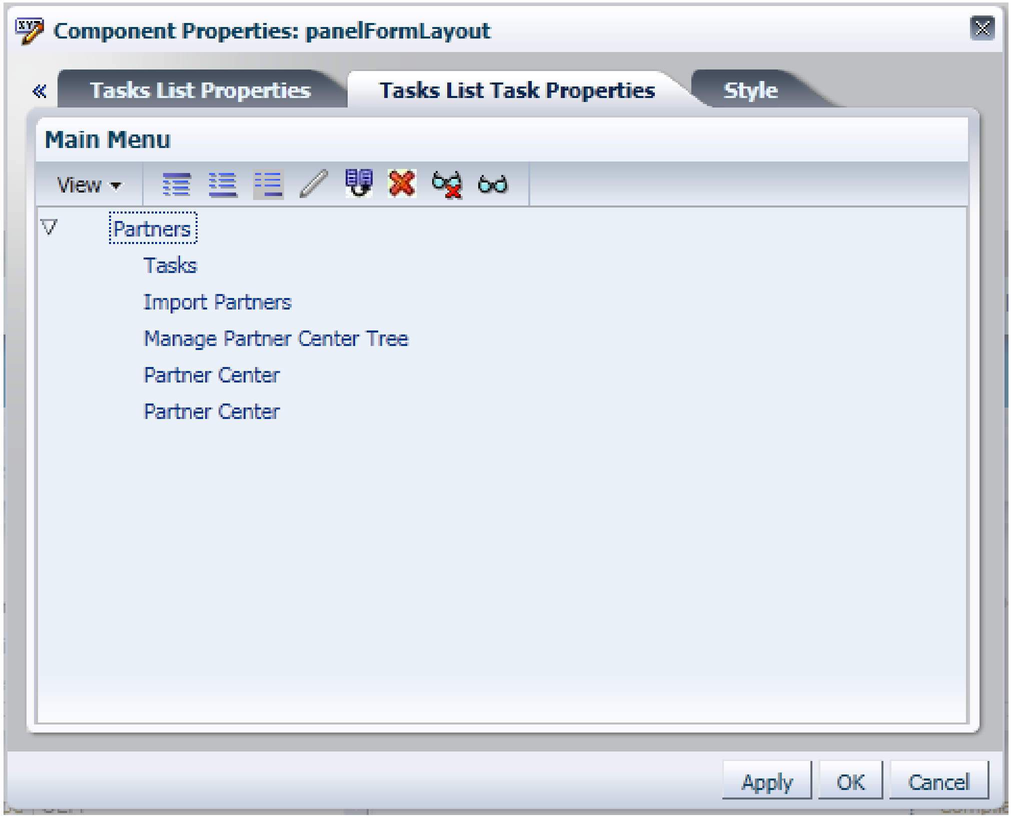


   The Component Properties dialog for `panelFormLayout` is displayed.

7. Click the **Tasks List Task Properties** tab.

8. Expand the tree to display the items in the task list, as shown in Figure 3–13.

Figure 3–13 Component Properties — Tasks List Task Properties Task List Items

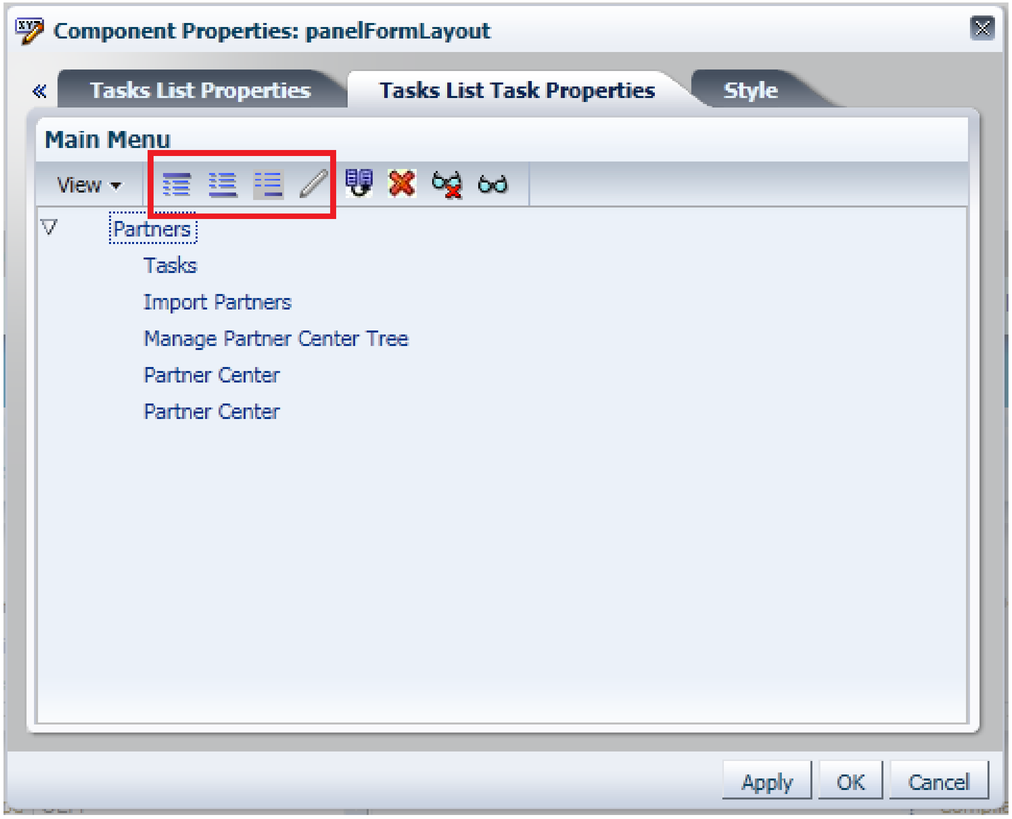


Component Properties: panelFormLayout

▽ Partners
- Tasks
- Import Partners
- Manage Partner Center Tree
- Partner Center
- Partner Center

9. In the tree hierarchy, select an item and customize the task list by inserting a new item above, inserting a new item below, inserting a child item, or editing the current item by clicking the appropriate icon in the toolbar, as shown in Figure 3–14.

Figure 3–14 Component Properties — Tasks List Task Properties Toolbar



10. Enter or edit the following values:

> **Note:** For Oracle Cloud implementations: to obtain a list a valid values, log a service request using My Oracle Support at `https://support.oracle.com`.

- **Web Application**: Use the dropdown list to select the target web application.

  This list contains web applications that are defined in the deployments tables.

  > **Caution:** If you enter a value for **Web Application** then you must enter a value for **Focus View Id**.

- **Focus View Id**: Enter the `focusViewId` value of the target page, for example, */ServiceRequest*.

  > **Caution:** If you enter values for **Web Application** and **Focus View Id**, do not enter a value for **Destination** and vice versa.

- **Action**: Enter the action that is taken when this item is selected by the user.

  Pages with actions are defined in the `adfc-config.xml` file, and these actions can take the user to a particular page. If you specify an action here, then the

Web Application and **Focus View Id** values are ignored. This **Action** attribute is used in an ADF Controller navigation.

- **Label**: Enter the label name for this new item. This is the name that appears on the task list menu. This label name also appears on the **Task** tab when opened if the page's `isDynamicTabNavigation="true"`.
- **Rendered**: Select this to display the item in the task list. Deselect to hide the item.

Note: When deselected, the item is displayed in italics on the customization dialog. This allows you to identify items that are currently hidden from users.

- **Destination**: Enter the full URL for this item, such as *http://www.example.com*.

  The Destination attribute is used for navigation outside of the Oracle Fusion Middleware UI Shell pages. It opens in a new window.

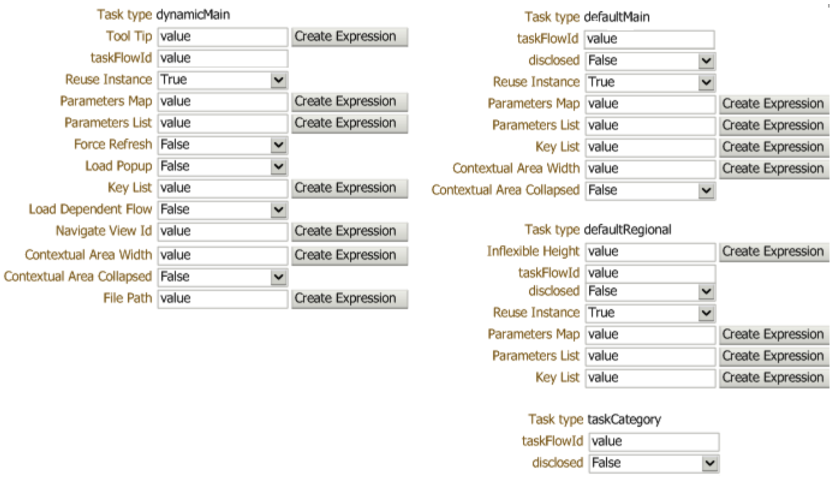
Note: Destination takes precedence over any specified Web Application value.

- **Task Type**: Choose the required task type for newly created items. Use the dropdown menu to select `dynamicMain`, `defaultMain`, `defaultRegional`, or `taskCategory`.

Caution: The task type can be specified by the administrator for the newly inserted item nodes only. It cannot be updated for an existing item node.

Figure 3–15 lists the properties that are applicable based on the task type of the currently edited item node.

Figure 3–15 Tasks List Task Properties — Task Types



Task type	dynamicMain	
Tool Tip	value	Create Expression
taskFlowId	value	
Reuse Instance	True	
Parameters Map	value	Create Expression
Parameters List	value	Create Expression
Force Refresh	False	
Load Popup	False	
Key List	value	Create Expression
Load Dependent Flow	False	
Navigate View Id	value	Create Expression
Contextual Area Width	value	Create Expression
Contextual Area Collapsed	False	
File Path	value	Create Expression

Task type	defaultMain	
taskFlowId	value	
disclosed	False	
Reuse Instance	True	
Parameters Map	value	Create Expression
Parameters List	value	Create Expression
Key List	value	Create Expression
Contextual Area Width	value	Create Expression
Contextual Area Collapsed	False	

Task type	defaultRegional	
Inflexible Height	value	Create Expression
taskFlowId	value	
disclosed	False	
Reuse Instance	True	
Parameters Map	value	Create Expression
Parameters List	value	Create Expression
Key List	value	Create Expression

Task type	taskCategory
taskFlowId	value
disclosed	False

11. Click **Apply** to save your changes, then click **OK** to save your changes and close the Component Properties dialog.

Task: Customize Attributes for a Flexfield on a Page

Note: You can perform this task only in non-CRM applications.

After you deploy an **extensible flexfield** or **descriptive flexfield**, you can use Page Composer to further control the custom attribute properties on a page-by-page basis. For example, you can hide some custom attributes or reorder how they appear on the page.

Note: For information about flexfields, see Chapter 4, "Using Flexfields for Custom Attributes."

To customize flexfield values, edit the page in Source view. In Source view, navigate through the tree and expand the flexfield component (for example `descriptiveFlexfield`) to see the custom attributes. Click the flexfield component to display the Component Properties dialog box, where you can edit the values for the attributes.

Task: Customize Popup Window Content

Note: You can perform this task only in non-CRM applications.

You can use Page Composer to customize the content in popup windows as follows:

1. If Source view is not already displayed, switch to Source view. From the **View** menu, select **Source**.
2. Select the button that brings up the popup window.
3. In the Source region, right-click the `popup` node, and select **Edit**.
4. In the Component Properties dialog, edit the popup window properties as desired.

   For information about popup window properties, see the "Using Popup Dialogs, Menus, and Windows" section in the *Oracle Fusion Middleware Web User Interface Developer's Guide for Oracle Application Development Framework*.
5. Click **Apply** to save your changes, then click **OK** to save your changes and close the Component Properties dialog.

3.3 Editing Component Properties in Page Composer

Note: You can perform these tasks only in non-CRM applications.

Components, such as task flows, portlets, documents, and layout components, carry with them a set of configurable properties that control the appearance and behavior of a particular component instance.

To edit component properties:

1. In Design view, click the Edit icon in the component header.
2. Click the appropriate tab in the Component Properties dialog.

3. Edit the properties.

4. Click **Apply** to save your changes, then click **OK** to save your changes and close the Component Properties dialog.

> **Note:** If the object you want to edit is included on multiple pages, then see Section 3.1.2, "Effects of Editing Objects That Display on Multiple Pages," before editing the object.

For more information, see the "Setting Properties on a Component" section in the *Oracle Fusion Middleware User's Guide for Oracle WebCenter Portal: Spaces*.

Task: Edit Component Header and Other Display Options

Typically, the **Display Options** tab presents settings that affect the display elements surrounding component content (that is, its chrome). Chrome includes the header, the **Actions** menu, **Expand** and **Collapse** icons, and the like. For example, use the display options on a task flow to hide or show a header, change the text in the header, enable or disable menus, show a tooltip for the component, and other options. Use the display options on an Image layout component to specify the image source URL and its optional link target. For more information, see the "Working with Component Display Options" section in the *Oracle Fusion Middleware User's Guide for Oracle WebCenter Portal: Spaces*.

Task: Edit Component and Content Style

Style and Content Style properties provide an opportunity to fine-tune your application look-and-feel at the component level. You can specify color, style, and margin settings on the selected component instance. For more information, see the "Working with Style and Content Style Properties" section in the *Oracle Fusion Middleware User's Guide for Oracle WebCenter Portal: Spaces*.

Task: Edit Component Parameters

Component parameters are settings, unique to the component type, that can control such things as the source of the component's content. Component parameters vary from component to component. For example, on some components, they provide the opportunity to specify the source of task flow content; on other components, they present read-only, application-generated identifiers that are used in maintaining a task flow instance's association with its customizations. For more information, see the "Working with Component Parameters" section in the *Oracle Fusion Middleware User's Guide for Oracle WebCenter Portal: Spaces*.

Task: Customize a Component Based on Role

Role level customization allows you to display a component only if the user has the specified role. For example, you might want to show a "Reports" button only if the user has a manager role.

If you are customizing a CRM application, you can select CRM's `Role` layer in the Layer Picker dialog, which appears when you open Page Composer.

If you are customizing a non-CRM based application, there is no `Role` layer to customize. Instead you could do the following:

- Use JDeveloper to customize the application, as described in the "Customizing at the Role Level" section in the *Oracle Fusion Applications Extensibility Guide for Developers*.

- Use an EL expression in the component property to conditionally render the component based on a specified security level associated with a role:

  1. Define a privilege to use to determine whether to render the component. For example, create a privilege called `MANAGER_REPORTS_BUTTON_PRIV`.
  2. Grant the privilege to the role the user requires to see the conditionally rendered component.

     For more information on defining privileges and granting them to roles, see the "Enabling ADF Security in a Fusion Web Application" chapter in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*.
  3. Make sure the users you want to see the conditionally rendered component are assigned the required role.
  4. Edit the component properties for the component you want to conditionally render (for example, the "Reports" button). On the **Display Options** tab, set the **Show Component** property to be an EL expression that verifies whether the user is granted the appropriate privilege. For example, use the following expression:

     ```
     #{securityContext.userGrantedPermission['MANAGER_REPORTS_BUTTON_PRIV']}
     ```

Task: Reset a Page or Task Flow to a Previously Saved Version

Page Composer provides controls for resetting a page or task flow to a previously saved version or to its original out-of-the-box state.

The **Reset Page** button is available on the page in both Design view and Source view. For more information, see the "Reset Page" section in the *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter Portal*.

The **Reset Task Flow** button on the Source view toolbar is rendered only when editing a task flow. For more information, see the "Reset Task Flow" section in the *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter Portal*.

Task: Allow Certain Component Property Values to Persist Across Sessions

Certain attribute values of ADF Faces components can be persisted for end users. For example, on the column component, an end user can change the width of a column, and that width will still be in effect when the user logs back in to the application. For information about which component properties can persist, see the "Introduction to Allowing User Customizations" section in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*.

To enable a property to be persistent, list that property as a value for the persist parameter, using the procedures in the "Working with Style and Content Style Properties" section in the *Oracle Fusion Middleware User's Guide for Oracle WebCenter Portal: Spaces*.

For more information about user personalization of components, see the "Configuring End-User Personalization for Components" section in the *Oracle Fusion Applications Extensibility Guide for Developers*.

3.4 Editing the UI Shell Template Used by All Pages

Notes:

- To customize the Oracle Fusion Applications skin (for example, to change the logo or change colors), follow the steps described in the "Customizing the Oracle Fusion Applications Skin" chapter in the *Oracle Fusion Applications Extensibility Guide for Developers*.
- In non-Cloud implementations, you can also use JDeveloper to edit the UI Shell template (for example, to make customizations that cannot be done through Page Composer). For more information, see the "Editing the UI Shell Template" section in the *Oracle Fusion Applications Extensibility Guide for Developers* or the "Introduction to Implementing the UI Shell" section in *Oracle Fusion Applications Developer's Guide*.

The UI Shell template provides the global or header area (shown in Figure 3–16) and the footer panel (shown in Figure 3–17, highlighted in red) for all pages in your application.

Figure 3–16 UI Shell Template — Global Area

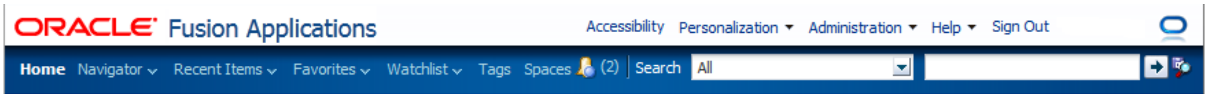
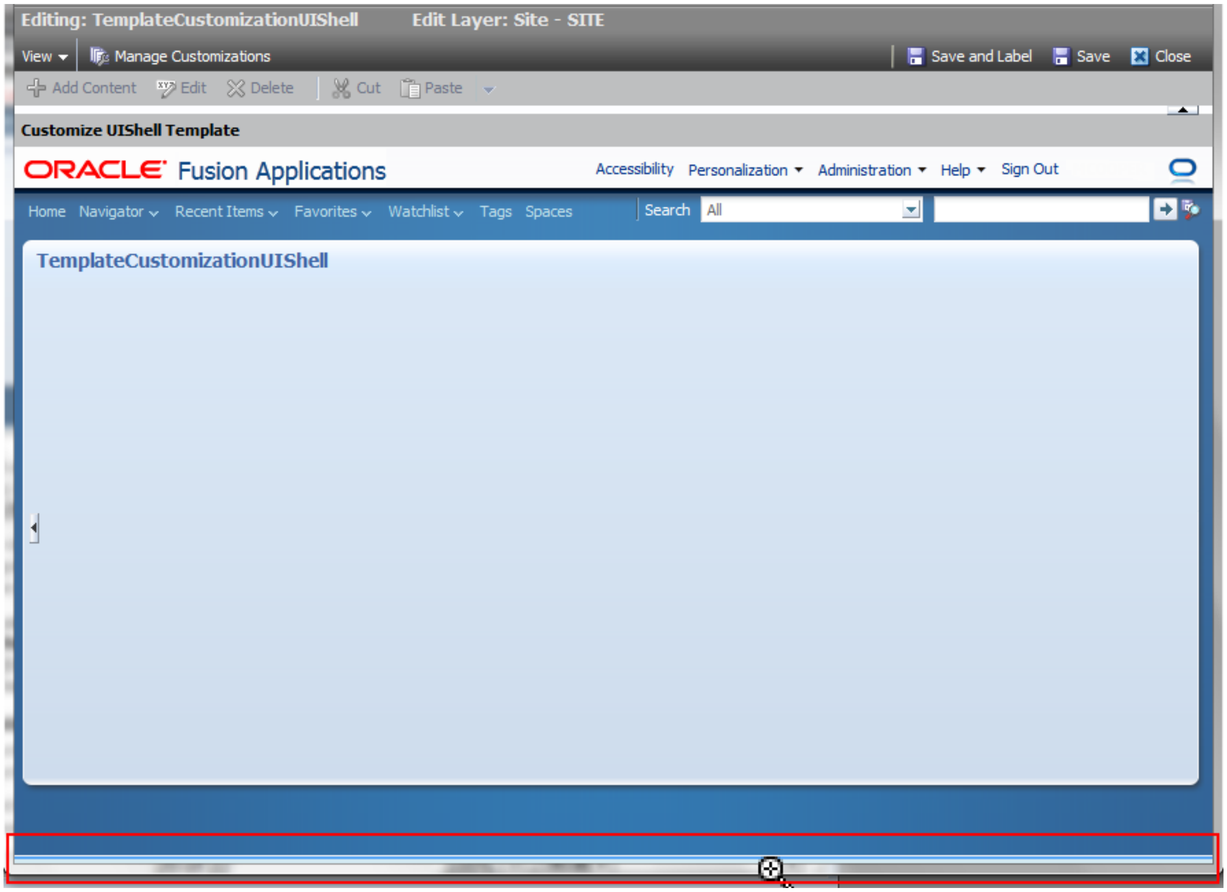


Figure 3–17 UI Shell Template — Footer



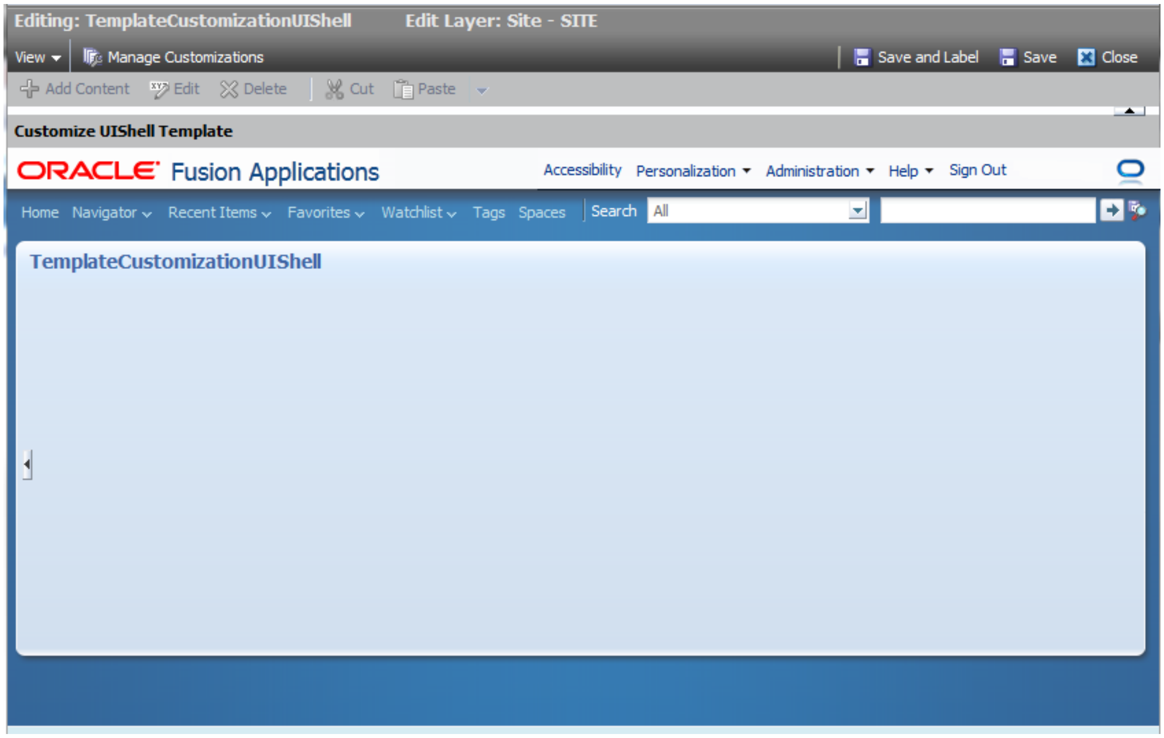
Note: If the footer does not contain any content it appears as a thin bar at the bottom of the browser window. The area highlighted in red in Figure 3–17 shows the footer ready to be selected.

You can use Page Composer to edit the UI Shell template. For example, you might want to edit the links available in the global area (such as removing the Tags link from the header area) or add company contact information to the footer.

To open the UI Shell template in Page Composer (shown in Figure 3–18), use one of the following methods:

- In the **Administration** menu, click **Customize Global Page template**.
- Open the UI Shell Template task flow.

Figure 3–18 UI Shell Template in Page Composer



Task: Add a Component to the UI Shell Template

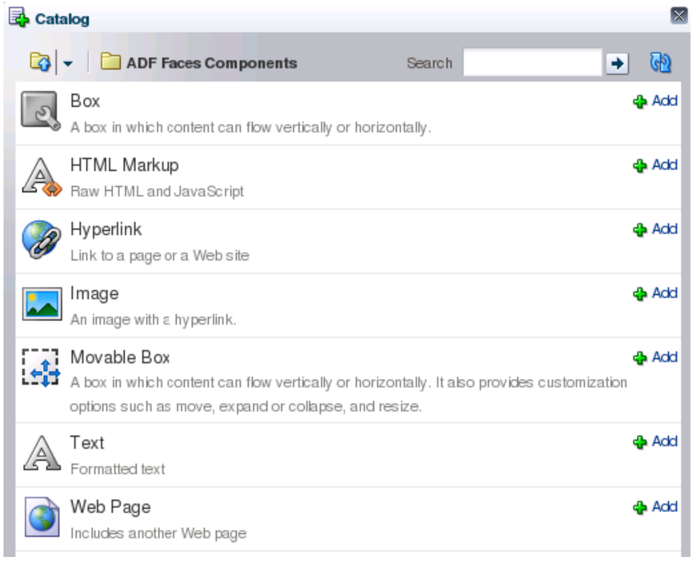
1. If you have not already opened the UI Shell template in Page Composer, open it now as previously described.

2. Select the portion of the global area to which you want to add a component, or select the footer, and click **Add Content**.

> **Note:** Editable areas display a blue outline when you place your cursor over them.

3. In the component catalog, select **ADF Faces Components** to display the list of available components, as shown in Figure 3–19.

Figure 3–19 ADF Faces Components Catalog



Box — Add
A box in which content can flow vertically or horizontally.

HTML Markup — Add
Raw HTML and JavaScript

Hyperlink — Add
Link to a page or a Web site

Image — Add
An image with a hyperlink.

Movable Box — Add
A box in which content can flow vertically or horizontally. It also provides customization options such as move, expand or collapse, and resize.

Text — Add
Formatted text

Web Page — Add
Includes another Web page

4. Next to the component that you want to add, click the associated **Add** icon.

   The component is appears in the global area or footer. If necessary, edit the component. For example, if you added the Text component, you can enter the text that you want displayed.

5. If you are done making changes to the UI Shell template, click **Close**. When prompted, click **Save** to save your changes.

Task: Edit a Component in the UI Shell Template

1. If you have not already opened the UI Shell template in Page Composer, open it now as previously described.
2. Select the component that you want to edit.

> **Note:** Editable components display a blue outline when you place your cursor over them.

3. Click **Edit**.
4. Edit the component, then click **OK** to save your changes.
5. If you are done making changes to the UI Shell template, click **Close**. When prompted, click **Save** to save your changes.

Task: Delete a Component in the UI Shell Template

1. If you have not already opened the UI Shell template in Page Composer, open it now as previously described.
2. Select the component that you want to delete.

> **Note:** Editable components display a blue outline when you place your cursor over them.

3. Click **Delete**. When prompted, click **Delete** to delete the component.

4. If you are done making changes to the UI Shell template, click **Close**. When prompted, click **Save** to save your changes.

3.5 Editing Pages in Oracle JDeveloper After Using Page Composer

Note: This section does not apply to Oracle Cloud implementations.

Using Page Composer, you (the administrative user) can implement a variety of customizations on an application's pages. Pages that were created or customized in JDeveloper are further customizable in Page Composer, and page customizations that were implemented in Page Composer can be viewed in JDeveloper.

To view Page Composer customizations in JDeveloper, and potentially further customize the pages, you must export the customizations from the runtime environment and import them into the JDeveloper customization application workspace. For more information, see the "Viewing ADF Library Runtime Customizations from Exported JARs" section in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*.

It is important to note that you cannot customize a given artifact at the same layer in both JDeveloper and Page Composer. You can, however, customize a given artifact in both tools provided the customizations are made at different layers. At run time, the tip layer customizations take precedence. For example, if you customize the label for a field in the **site layer** using Page Composer and customize the same label in the **global layer** using JDeveloper, the site layer customization will be displayed at run time.

4

Using Flexfields for Custom Attributes

This chapter describes how to use descriptive and extensible flexfields to add additional attributes to your Oracle Fusion applications. You create these custom attributes using runtime tasks in the Setup and Maintenance work area.

This chapter includes the following sections:

4.1 About Using Flexfields

If you need to add company-specific attributes to a **business object**, such as size and color attributes for the product business object, then you can often use flexfields to add the desired custom attributes. A **flexfield** is a set of placeholder fields that is associated with a business object. With flexfields, a field is called a **segment**.

Oracle Fusion Applications provides three types of flexfields — **descriptive flexfield**, **extensible flexfield**, and **key flexfield**. This chapter discusses how you can use descriptive and extensible flexfields to add additional attributes to Oracle Fusion applications. This chapter refers to these attributes as **custom attributes**.

> **Note:** This chapter does not discuss key flexfields, which you use to define keys such as part numbers, as explained in the product-specific documentation in Oracle Fusion Applications Help. In this chapter, the term flexfield applies to descriptive and extensible flexfields only.

Descriptive and extensible flexfields enable you to configure your applications to capture additional pieces of information (attributes) without having to perform custom development. The attributes that you add by configuring flexfields are available throughout the Oracle Fusion Middleware technology stack, allowing custom attributes to be used in user interface pages, incorporated into the service-oriented architecture (SOA) infrastructure, and, in the case of descriptive flexfields, integrated with Oracle Business Intelligence.

> **Note:** For Sales, Marketing, Customer Center, Trading Community Architecture, and Order Capture applications, use Oracle Fusion CRM Application Composer (Application Composer) to add custom attributes instead of using descriptive and extensible flexfields. For more information, see the "Application Composer: Introduction" chapter in the *Oracle Fusion Applications CRM Extensibility Guide*

4.1.1 About the Different Types of Flexfields

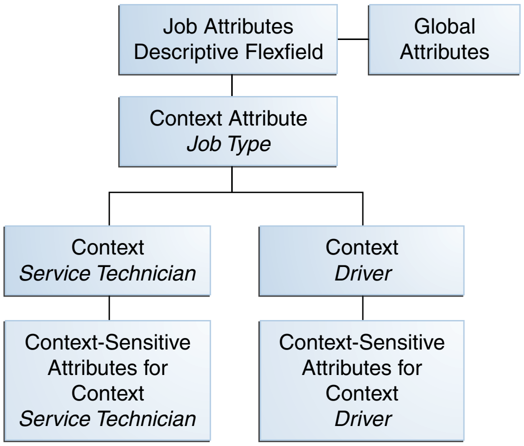
Many business objects in Oracle Fusion applications have an associated descriptive or extensible flexfield with which you can create custom attributes for the business object. A descriptive flexfield is the more basic of the two and is more commonly used. As explained later in this section, extensible flexfields offer more advanced features, such as hierarchical configurations.

The flexfield type determines how you configure the custom attributes.

- Descriptive flexfield configuration: A descriptive flexfield enables you to add three types of custom attributes to a page — global, context, and context-sensitive. Global attributes are always available as fields in the UI. Context-sensitive attributes appear on a UI page only under a certain condition or circumstance (the context).

  An example of where you would use context-sensitive attributes is for a job business object, as illustrated in Figure 4–1. Where some attributes are common to all jobs, such as whether the job is off site, some job attributes depend upon the job type. For example, you might want to store the service type for a service technician, and for a driver, you might want to store the required commercial driver's license (CDL) class.

Figure 4–1 Example of Descriptive Flexfield Global and Context-Sensitive Attributes


Job Attributes Descriptive Flexfield
Global Attributes
Context Attribute *Job Type*
Context *Service Technician*
Context *Driver*
Context-Sensitive Attributes for Context *Service Technician*
Context-Sensitive Attributes for Context *Driver*

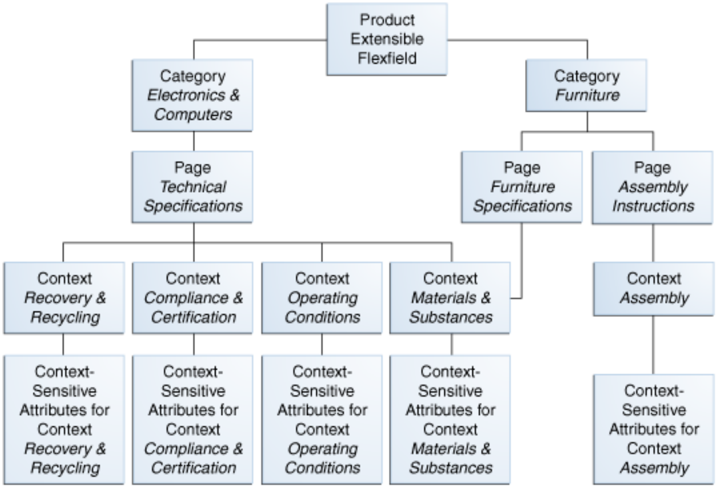
- Extensible flexfield configuration: An extensible flexfield enables you to add attributes to a page by configuring categories, logical pages, and contexts.

  Extensible flexfield contexts are somewhat different from descriptive flexfield contexts in that extensible flexfields enable you to configure multiple contexts, and you can group the contexts into categories. All extensible flexfields have at least one category (often referred to as the root category), and some extensible flexfields enable you to configure a hierarchy of categories, where a given category can inherit contexts from its parent categories. For more information about categories, see Task: Identify the Extensible Flexfield's Category Hierarchy Structure in Section 4.3.2, "Planning Extensible Flexfields."

You use logical pages to arrange how the contexts appear in the user interface. For example, the extensible flexfield in Figure 4–2 has been configured to include a Technical Specifications logical page in the user interface for the Electronics and Computers category. The Technical Specifications logical page contains the attributes for four contexts — Recovery and Recycling, Compliance and Certification, Operating Conditions, and Materials and Substances. Figure 4–3 shows the user interface for the Technical Specifications logical page.

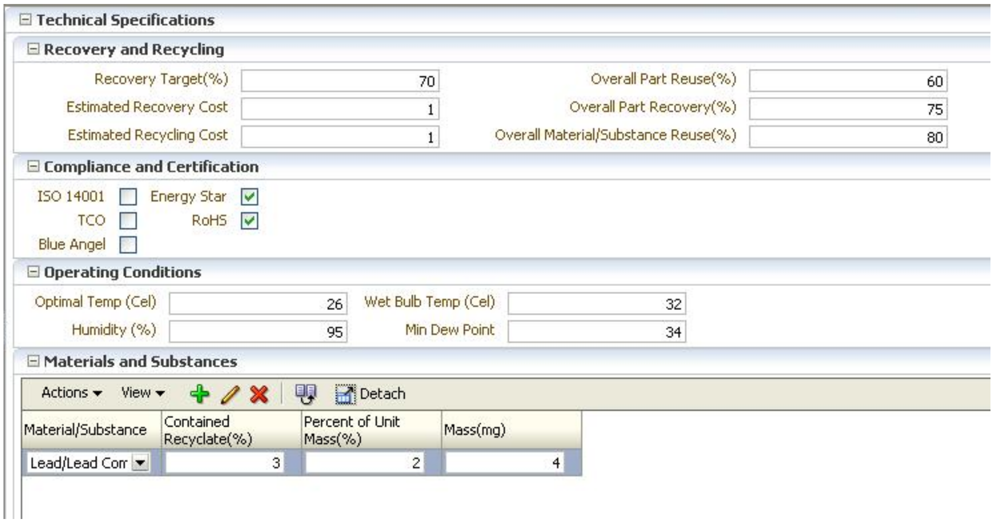
As shown in Figure 4–2, the user interface for the Furniture category has been configured to include a Furniture Specifications logical page and an Assembly Instructions logical page. Note that the two categories (Electronics & Computers and Furniture) share the Materials & Substances context.

Figure 4–2 Example of Extensible Flexfield Categories, Pages, and Contexts


Product Extensible Flexfield
Category *Electronics & Computers*
Category *Furniture*
Page *Technical Specifications*
Page *Furniture Specifications*
Page *Assembly Instructions*
Context *Recovery & Recycling*
Context *Compliance & Certification*
Context *Operating Conditions*
Context *Materials & Substances*
Context *Assembly*
Context-Sensitive Attributes for Context *Recovery & Recycling*
Context-Sensitive Attributes for Context *Compliance & Certification*
Context-Sensitive Attributes for Context *Operating Conditions*
Context-Sensitive Attributes for Context *Materials & Substances*
Context-Sensitive Attributes for Context *Assembly*

Another extensible flexfield feature is the ability to configure a context to store multiple rows per entity. For example, you can use a context to store all the materials and substances required to make a single product. The Materials and Substances context in Figure 4–3 has been configured for multiple rows, and is thus displayed as a table.

Figure 4–3 Technical Specifications Page with Associated Contexts



⊟ Technical Specifications

⊟ Recovery and Recycling

Recovery Target(%)	70	Overall Part Reuse(%)	60
Estimated Recovery Cost	1	Overall Part Recovery(%)	75
Estimated Recycling Cost	1	Overall Material/Substance Reuse(%)	80

⊟ Compliance and Certification

ISO 14001	□	Energy Star	☑
TCO	□	RoHS	☑
Blue Angel	□		

⊟ Operating Conditions

Optimal Temp (Cel)	26	Wet Bulb Temp (Cel)	32
Humidity (%)	95	Min Dew Point	34

⊟ Materials and Substances

Material/Substance	Contained Recyclate(%)	Percent of Unit Mass(%)	Mass(mg)
Lead/Lead Corr	3	2	4

Extensible flexfields also enable you to specify view and edit **privileges**. For example, you can set the privileges so that everyone can view the attributes but only managers can make changes.

4.1.2 What You Can Do with Flexfields at Runtime

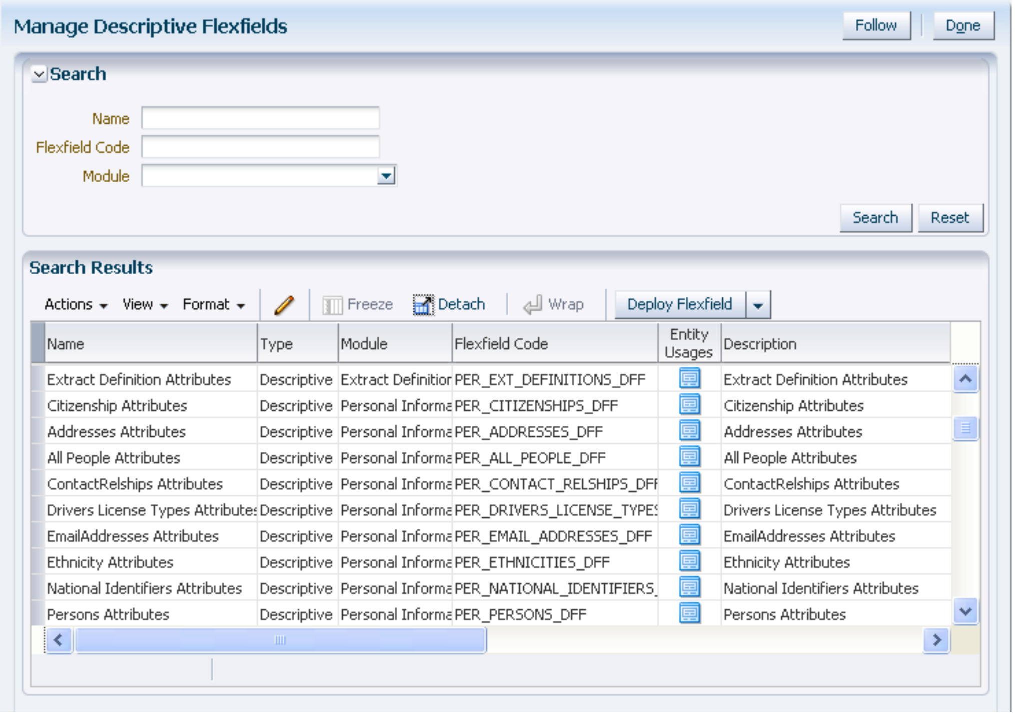
Using the runtime tools, you can perform the following flexfield tasks:

- Find the flexfields on a page.
- Manage flexfields.
- Specify flexfield validation.
- Deploy flexfields.
- Integrate flexfields into the technology stack.
- Customize the attributes on a per-page basis using Page Composer.
- Create translations of the value sets and the custom attributes for different locales.

You can use the Manage Descriptive Flexfields task, shown in Figure 4–4, and the Manage Extensible Flexfields task, shown in Figure 4–5, to create your custom attributes. You can also configure descriptive flexfields by using the three Quick Create dialogs — Create Context Value, Quick Create: Global Segment, and Quick Create: Context-Sensitive Segment. The Quick Create: Context-Sensitive Segment dialog is shown in Figure 4–6. For more information, see Section 4.5.1, "Configuring Descriptive Flexfields" and Section 4.5.2, "Configuring Extensible Flexfields."

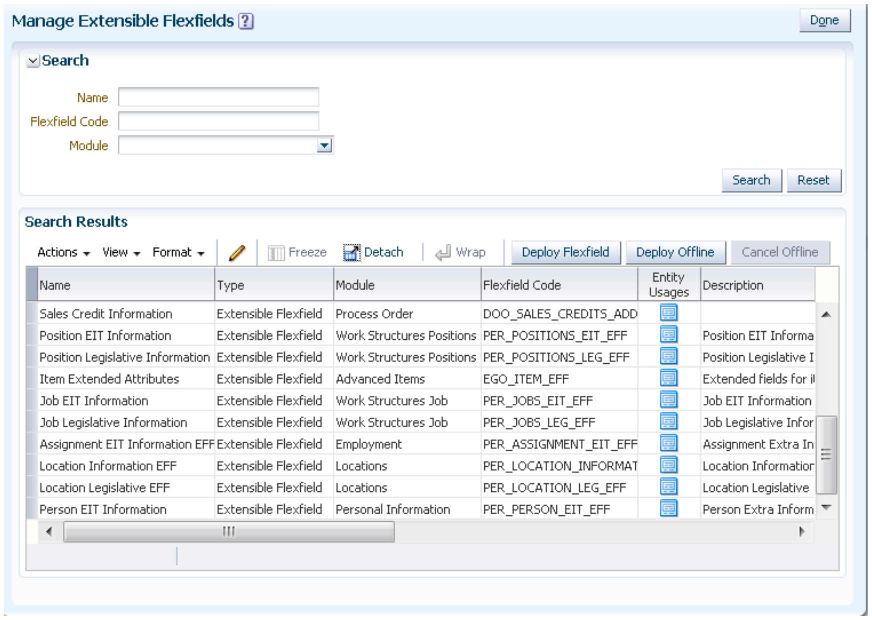
> **Note:** Some setup activities enable you to complete product-specific flexfield configuration. For example, you can use the Manage Item Classes task in the Product and Catalog Management application to manage a hierarchy of custom attributes for catalog items. For more information, see the product-specific documentation in Oracle Fusion Applications Help.

Figure 4–4 Manage Descriptive Flexfields Task



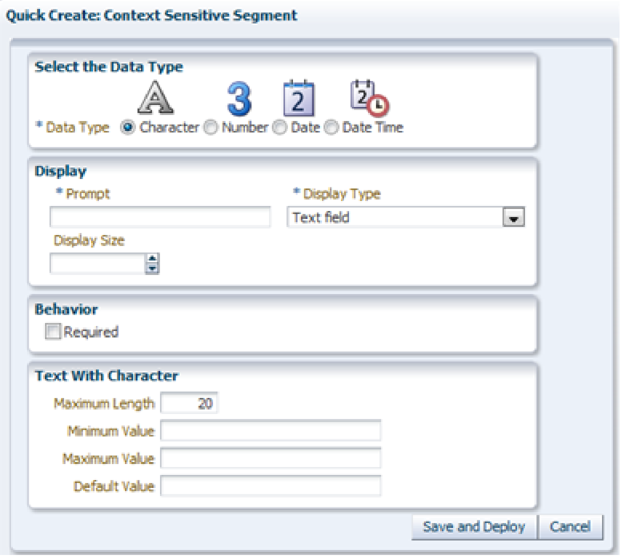
Name	Type	Module	Flexfield Code	Entity Usages	Description
Extract Definition Attributes	Descriptive	Extract Definitior	PER_EXT_DEFINITIONS_DFF	▤	Extract Definition Attributes
Citizenship Attributes	Descriptive	Personal Informa	PER_CITIZENSHIPS_DFF	▤	Citizenship Attributes
Addresses Attributes	Descriptive	Personal Informa	PER_ADDRESSES_DFF	▤	Addresses Attributes
All People Attributes	Descriptive	Personal Informa	PER_ALL_PEOPLE_DFF	▤	All People Attributes
ContactRelships Attributes	Descriptive	Personal Informa	PER_CONTACT_RELSHIPS_DFF	▤	ContactRelships Attributes
Drivers License Types Attribute:	Descriptive	Personal Informa	PER_DRIVERS_LICENSE_TYPES	▤	Drivers License Types Attributes
EmailAddresses Attributes	Descriptive	Personal Informa	PER_EMAIL_ADDRESSES_DFF	▤	EmailAddresses Attributes
Ethnicity Attributes	Descriptive	Personal Informa	PER_ETHNICITIES_DFF	▤	Ethnicity Attributes
National Identifiers Attributes	Descriptive	Personal Informa	PER_NATIONAL_IDENTIFIERS_	▤	National Identifiers Attributes
Persons Attributes	Descriptive	Personal Informa	PER_PERSONS_DFF	▤	Persons Attributes

Figure 4–5 Manage Extensible Flexfields Task



Name	Type	Module	Flexfield Code	Entity Usages	Description
Sales Credit Information	Extensible Flexfield	Process Order	DOO_SALES_CREDITS_ADD	▤	
Position EIT Information	Extensible Flexfield	Work Structures Positions	PER_POSITIONS_EIT_EFF	▤	Position EIT Informa
Position Legislative Information	Extensible Flexfield	Work Structures Positions	PER_POSITIONS_LEG_EFF	▤	Position Legislative I
Item Extended Attributes	Extensible Flexfield	Advanced Items	EGO_ITEM_EFF	▤	Extended fields for i
Job EIT Information	Extensible Flexfield	Work Structures Job	PER_JOBS_EIT_EFF	▤	Job EIT Information
Job Legislative Information	Extensible Flexfield	Work Structures Job	PER_JOBS_LEG_EFF	▤	Job Legislative Infor
Assignment EIT Information EFF	Extensible Flexfield	Employment	PER_ASSIGNMENT_EIT_EFF	▤	Assignment Extra In
Location Information EFF	Extensible Flexfield	Locations	PER_LOCATION_INFORMAT	▤	Location Informatior
Location Legislative EFF	Extensible Flexfield	Locations	PER_LOCATION_LEG_EFF	▤	Location Legislative
Person EIT Information	Extensible Flexfield	Personal Information	PER_PERSON_EIT_EFF	▤	Person Extra Inform

Figure 4–6 Descriptive Flexfield Quick Create: Context-Sensitive Segment Dialog



Quick Create: Context Sensitive Segment

Select the Data Type

* Data Type ● Character ○ Number ○ Date ○ Date Time

Display

* Prompt

* Display Type: Text field

Display Size

Behavior

□ Required

Text With Character

Maximum Length: 20

Minimum Value

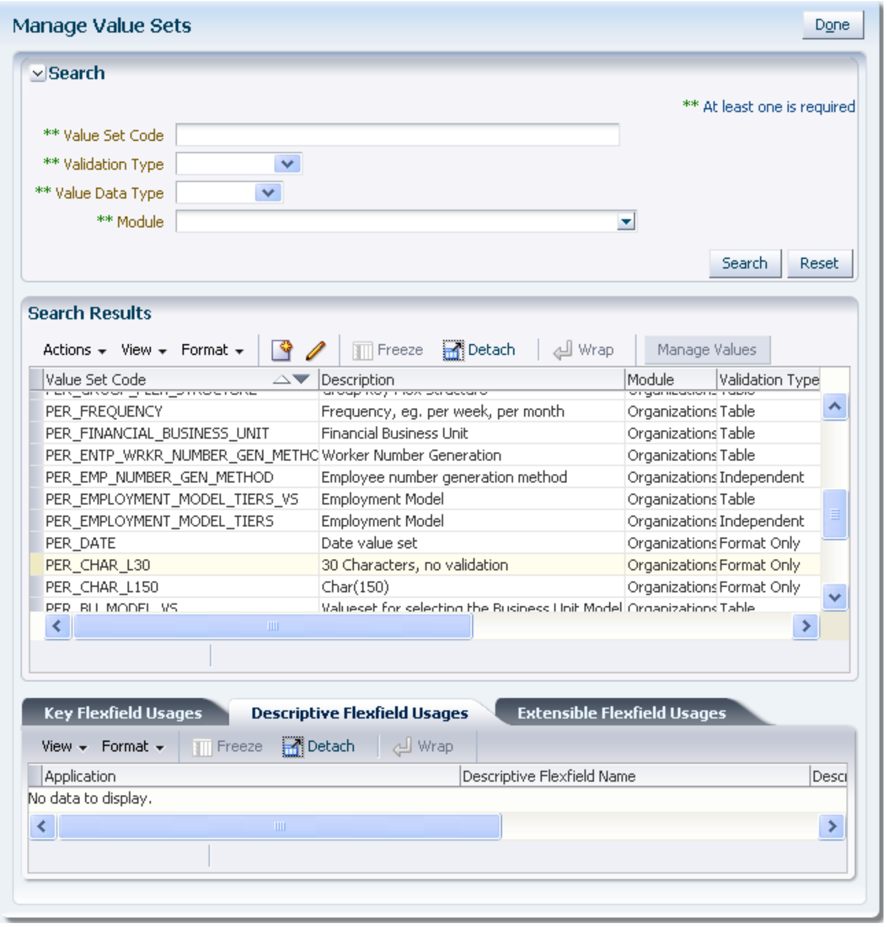
Maximum Value

Default Value

As shown in Figure 4–7, use the Manage Value Sets task to specify validation rules for your custom attributes, such as minimum and maximum values or a list of valid values.

Note: If you use the Quick Create dialogs to configure the attributes, the value sets are created automatically when you add the segments. You cannot use an existing value set.

Figure 4–7 Manage Value Sets Task



Manage Value Sets

Search

** At least one is required

** Value Set Code

** Validation Type

** Value Data Type

** Module

Search Results

Value Set Code	Description	Module	Validation Type
PER_FREQUENCY	Frequency, eg. per week, per month	Organizations	Table
PER_FINANCIAL_BUSINESS_UNIT	Financial Business Unit	Organizations	Table
PER_ENTP_WRKR_NUMBER_GEN_METHC	Worker Number Generation	Organizations	Table
PER_EMP_NUMBER_GEN_METHOD	Employee number generation method	Organizations	Independent
PER_EMPLOYMENT_MODEL_TIERS_VS	Employment Model	Organizations	Table
PER_EMPLOYMENT_MODEL_TIERS	Employment Model	Organizations	Independent
PER_DATE	Date value set	Organizations	Format Only
PER_CHAR_L30	30 Characters, no validation	Organizations	Format Only
PER_CHAR_L150	Char(150)	Organizations	Format Only
PER_BU_MODEL_VS	Valueset for selecting the Business Unit Model	Organizations	Table

Key Flexfield Usages | Descriptive Flexfield Usages | Extensible Flexfield Usages

Application	Descriptive Flexfield Name	Desci
No data to display.		

You must deploy a flexfield as described in Section 4.7, "Deploying Flexfield Configurations" before you can access the custom attributes from the user interface and the technology stack. After you deploy the flexfield, you can use Page Composer to configure the attributes on a per-page basis, and you can incorporate the custom attributes into Oracle Business Intelligence, web services, the SOA infrastructure, and Oracle Social Network, as described in Section 4.8, "Integrating Custom Attributes."

You can create translations of the value sets and the custom attributes for different locales as described in the "Translating Custom Text" chapter in the *Oracle Fusion Applications Extensibility Guide for Developers*.

4.1.3 What You Cannot Do with Flexfields at Runtime

You cannot use flexfields for the following tasks:

- You cannot use the flexfield feature to add attributes to business objects that do not have a descriptive or extensible flexfield as described in the "Getting Started with Flexfields" chapter in the *Oracle Fusion Applications Developer's Guide*.
- You cannot use key flexfields to add custom attributes.
- You cannot use flexfields to add attributes for the following applications, which are part of the Oracle Fusion Customer Relationship Management product family. With those applications, you must use Application Composer to add custom attributes.

- Sales
- Marketing
- Customer Center
- Trading Community Architecture
- Order Capture

For more information, see the "Editing an Object: Explained" section in the *Oracle Fusion Applications CRM Extensibility Guide*

4.1.4 What You Cannot Do with Flexfields at Design Time

Do not use Oracle JDeveloper to customize flexfields. If you require flexfield changes that you cannot accomplish using the tasks and UI that are described in this chapter, contact My Oracle Support at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

4.1.5 Before You Begin Using Flexfields to Create Custom Attributes

Before you use flexfields to create custom attributes, you should be familiar with the Oracle Fusion application architecture that enables customization, as described in Chapter 1, "Customizing and Extending Oracle Fusion Applications." You should also understand the typical workflows for working with customizations, as described in Chapter 2, "Understanding the Customization Development Lifecycle."

4.1.6 How to Configure Custom Attributes

Perform the following steps to configure custom attributes. These steps are described in detail in the remaining sections of this chapter.

1. Find the flexfield on the page.
2. Plan the flexfield configuration.
3. Plan flexfield validation.
4. Define the attributes.

   For a simple descriptive flexfield configuration, such as adding a format-only field or adding a field with a basic list of values, you can use the Quick Create dialogs. For other configurations, complete the following tasks.

   a. Define the attributes using the appropriate task — Manage Extensible Flexfields or Manage Descriptive Flexfields

   b. Optionally, validate the flexfield configuration.

   c. Deploy the flexfield to the **flexfield sandbox** and test the configuration.

   d. After testing, deploy the flexfield to display the custom attributes on the application pages in the full test environment and to make them available for integration into the Oracle Fusion Middleware technology stack. The flexfield artifacts that are generated during deployment are saved to an Oracle Metadata Services (MDS) repository.

5. Perform the necessary steps to integrate the custom attributes into the technology stack.

4.2 Finding the Flexfields on a Page

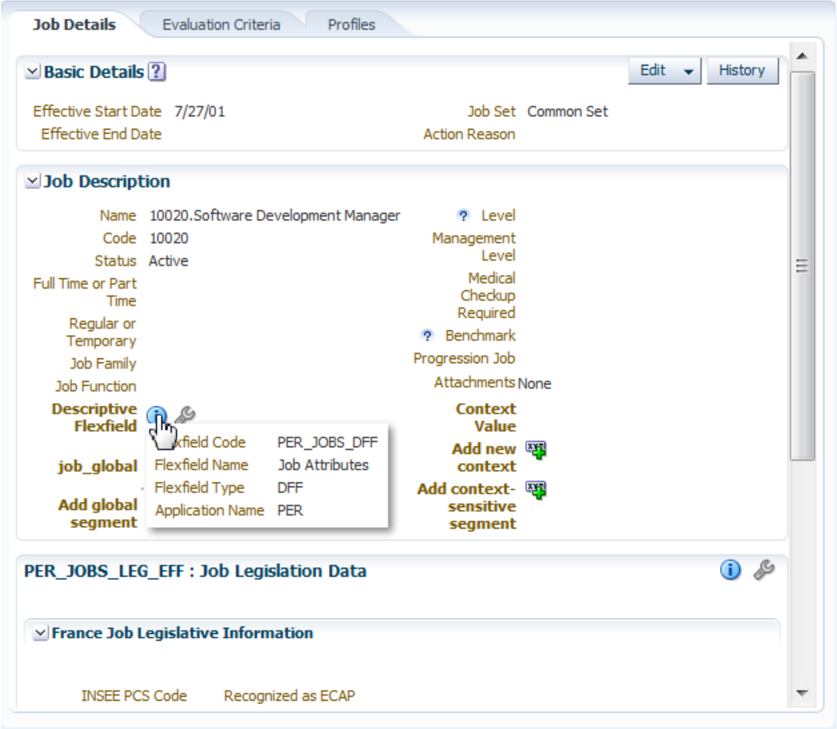
Before you begin the planning stage, you must first determine what type of flexfields — extensible or descriptive — are available for customizing. You also need the flexfield codes. You can choose **Highlight Flexfields** from the **Administration** menu or you can use Page Composer to find the flexfields on a page.

Task: Use Highlight Flexfields to Find the Flexfields on a Page

To obtain information about the flexfields on a page, open the page and choose **Highlight Flexfields** from the **Administration** menu. Hover over the **Information** icon next to the highlighted fields to display information about the flexfield, as shown in Figure 4–8. Choose **Unhighlight Flexfields** from the **Administration** menu when you no longer want to see the highlighted flexfields.

> **Tip:** When you click the wrench icon for a highlighted flexfield, the applicable Manage Flexfields task is displayed for that flexfield. If the flexfield is a descriptive flexfield, you can click the appropriate **Add** icon to add a context value, a global segment, or a context-sensitive segment that does not require advanced configurations. For more information, see Section 4.5.1, "Configuring Descriptive Flexfields,"

Figure 4–8 Highlight Flexfields UI



Basic Details

Effective Start Date 7/27/01
Effective End Date
Job Set Common Set
Action Reason

Job Description

Name 10020.Software Development Manager
Code 10020
Status Active
Full Time or Part Time
Regular or Temporary
Job Family
Job Function
Descriptive Flexfield
job_global
Add global segment

xfield Code PER_JOBS_DFF
Flexfield Name Job Attributes
Flexfield Type DFF
Application Name PER

Level
Management Level
Medical Checkup Required
Benchmark
Progression Job
Attachments None
Context Value
Add new context
Add context-sensitive segment

PER_JOBS_LEG_EFF : Job Legislation Data

France Job Legislative Information

INSEE PCS Code
Recognized as ECAP

> **Note:** Not all flexfields are available for creating custom attributes. Consult the product-specific documentation in Oracle Fusion Applications Help to verify whether there are any restrictions on using the flexfield.

Task: Use Page Composer to Find the Flexfields on a Page

Use Page Composer to get information about the flexfields on a page. Open the page in Source view and look for the following:.

- A region that is bound to an EffContextsPageContainer task flow. Open the properties panel for the region to view the flexfield name, flexfield code, and, optionally, the usage.
- `<descriptiveFlexfield>` elements. Open the properties panel for the element to view the flexfield name and flexfield code.

> **Note:** Not all flexfields are available for creating custom attributes. Consult the product-specific documentation in Oracle Fusion Applications Help to verify whether there are any restrictions on using the flexfield.

4.3 Planning Your Flexfields

The planning of your custom attributes is an important activity in preparation for flexfield configuration. The way in which you plan and configure a flexfield depends on whether it is descriptive or extensible.

4.3.1 Planning Descriptive Flexfields

If you are configuring a descriptive flexfield, complete the following tasks:

1. Identify existing context values.
2. Identify whether the context value is derived.
3. Identify available parameters.
4. List custom attributes.
5. Plan the descriptive flexfield structure.
6. Define attribute properties.
7. Define validation rules.
8. Define initial values.
9. Plan attribute mapping to Oracle Business Intelligence objects.

Task: Identify Existing Descriptive Flexfield Context Values

To identify existing context values, access the Manage Descriptive Flexfields task by completing one of the following steps:

- Choose **Setup and Maintenance** from the **Administration** menu in the global area of Oracle Fusion Applications and search for the Manage Descriptive Flexfields task. Next, search for and edit the flexfield to view the list of configured context values.
- Access the page that contains the flexfield, choose **Highlight Flexfields** from the **Administration** menu, click the **Edit Descriptive Flexfield** (wrench) icon for the highlighted flexfield, and view the list of configured context values.

If context values have been preconfigured, you should consult the product-specific documentation in Oracle Fusion Applications Help for information about the use of those values.

Task: Evaluate Whether the Context Value Is Derived

The context value for a descriptive flexfield might have been preconfigured to be derived from an external reference. For example, if the context is Marriage Status, then the value might be derived from an attribute in the employee business object. When the context value is derived, you might need to take the derived values and their source into consideration in your plan. To determine whether the context value is derived, look at the **Derivation Value** field in the Context Segment region in the Edit Descriptive Flexfields window for the flexfield. The dropdown list displays a list of available parameters. For more information about derived values, see Task: Identify Available Descriptive Flexfield Parameters.

Task: Identify Available Descriptive Flexfield Parameters

Some descriptive flexfields provide parameters, which enable you to set the initial value of an attribute from external reference data, such as a column value or a session variable. For example, if a flexfield has a user email parameter, then you would be able to configure the initial value for a customer email attribute to be derived from the parameter. You can view the **Derivation Value** dropdown list in the Create Segment window for the flexfield to see what parameters are available for a descriptive flexfield. If you decide to use one of the parameters to set an initial value, then you would select that parameter from the dropdown list when you add the attribute during the flexfield configuration process described in Section 4.5.1, "Configuring Descriptive Flexfields."

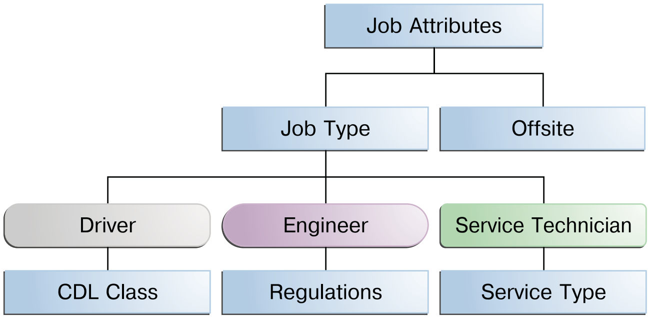
Task: List Custom Attributes

List all the custom attributes that you want to add. Later, as described in Task: Define Attribute Properties, you define the attribute details.

Task: Plan the Descriptive Flexfield Structure

When you configure the flexfield, you will use the flexfield's global, context, and context-sensitive segments to structure your custom attributes. For the custom attributes that you want to store for every instance of the business object, you will add them as global segments. For example, in Figure 4–9, the Offsite attribute is a global attribute that is stored for every job

Figure 4–9 Job Attributes Flexfield with Context-Sensitive and Global Attributes



For the custom attributes that you want to store based on a context (a condition or situation), define the context and list the valid values for the context. For example, you might want to use the Job Attributes descriptive flexfield on the Manage Jobs page to display different custom attributes depending on whether the job is for a service technician, an engineer, or a driver, as shown in Figure 4–9. You would define the context as Job Type, and you would list Service Technician, Engineer, and Driver as the list of valid values for the Job Type context. Next, list the custom attributes that you want to store for each of the valid values. For example, you might store the service type custom attribute for a service technician job and you might store the commercial driver's license (CDL) class for a driver job. Later, when you configure the flexfield, you would set the prompt for the context segment to Job Type, you would create the Service Technician, Engineer, and Driver context values, and you would define the context-sensitive segments for each of the three context values. For example, you would create a service type context-sensitive segment for the Service Technician context value and you would create a commercial driver's license (CDL) class context-sensitive segment for the Driver context value.

There is only one context segment available for descriptive flexfields. If you have more than one group of custom attributes where you could use the context segment, you will have to pick one group over the others, based on your company's needs and priorities, and add the other custom attributes as global segments.

The following list shows an example of a structural plan for the Job Attributes descriptive flexfield:

- Global segments
  - Offsite: YES or NO
- Context segment
  - Prompt: Job Type
  - Value set: JOB_TYPE
  - Context values:
    - Service Technician
    - Engineer
    - Driver
- Context-sensitive segments for Service Technician
  - Service type: such as appliance, heating, ventilation, and air conditioning (HVAC), facilities maintenance, and vehicle maintenance
- Context-sensitive segments for Engineer
  - Regulations: such as building codes and Occupational Safety and Health Administration (OSHA) regulations
- Context-sensitive segments for Driver
  - Commercial driver's license (CDL) class: such as A and B

Task: Define Attribute Properties

For each custom attribute that you want to add, define the attribute properties that are listed in Table 4–1.

Table 4–1 Attribute Properties

Property	Description
Sequence	The order the attribute appears in relation to the other configured attributes.
Enabled	Whether the attribute can be used.
Data type	Character, date, date and time, or number.

Table 4–1 (Cont.) Attribute Properties

Property	Description
Range type	If the attribute is part of a range specification, whether the attribute is the low value or the high value. For example, if adding minimum height and maximum height attributes, then the minimum height attribute has a range type of low.
Required	Whether the end user is required to provide a value.
Initial value	The initial default value for an attribute when the row is created. You can specify a constant value for both descriptive flexfields and extensible flexfields. You can specify a SQL statement for extensible flexfields, and for descriptive flexfields, you can specify a flexfield parameter. For more information, see Task: Define Initial Values for Descriptive Flexfield Custom Attributes and Task: Define Initial Values for Extensible Flexfield Custom Attributes.
Derivation value	The flexfield parameter from which to derive the attribute's value. Whenever the parameter value changes, the attribute's value is changed to match. If you derive an attribute value from a descriptive flexfield parameter, then consider making the attribute read-only, because user-entered values are lost whenever the parameter value changes.
Prompt	The string to be used for the attribute's label in the user interface.
Display type	The type of field in which to display the attribute. The available display types are described in Table 4–2.
Checked and unchecked values	If the display type is chooser (checkbox), the actual values to save. For example, Y and N or 0 and 1.
Display size	The character width of the field.
Display height	The height of the field.
Read only	Whether the field should display as read-only (noneditable) text.

Table 4–2 describes the display types that you can use for a segment.

Table 4–2 Display Types

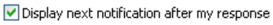
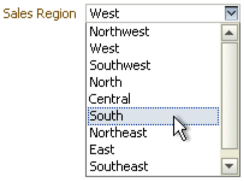
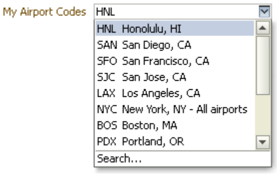
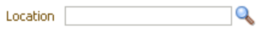
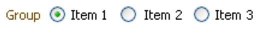
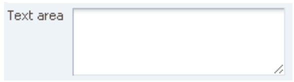
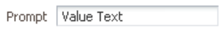
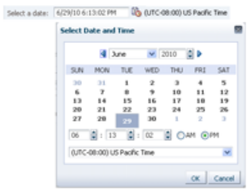
Display Type		Description
Checkbox		The field is displayed as a checkbox. If the end user selects the checkbox, the checked value is used. Otherwise, the unchecked value is used.
Drop-down List	 Sales Region West Northwest West Southwest North Central South Northeast East Southeast	The field displays a dropdown list of values from which the end user can select a value.
List of Values	 My Airport Codes HNL HNL Honolulu, HI SAN San Diego, CA SFO San Francisco, CA SJC San Jose, CA LAX Los Angeles, CA NYC New York, NY - All airports BOS Boston, MA PDX Portland, OR Search...	The field displays a dropdown list of values from which the end user can select a value. The user can also click **Search** to find more values.

Table 4–2 (Cont.) Display Types

Display Type		Description
Pop-up List of Values		The field displays as a text field with a **Search** icon. The end users can type a value in the text field or they can click the **Search** icon to open a dialog for searching.
Radio Button Group		The field is displayed as a set of radio buttons. The end user can select one button. Selecting a button deselects any previously selected button in the set.
Text Area		The field is displayed as a text area in which the end user can type multiple lines of text. The display width specifies the visible width of the text area. The display height specifies the visible number of lines in the text area.
Text Box		The field is displayed as a text field in which the end user can type a single line of text. The display width controls the width of the text box.
Date/Time	 Select a date: 6/29/10 6:13:02 PM (UTC-08:00) US Pacific Time Select Date and Time June 2010 SUN MON TUE WED THU FRI SAT OK Cancel	The field enables the end user to enter a date if the data type is Date, or a date and time if the data type is Date Time. The user can select the date from a calendar. If the data type is Date Time, the field also displays fields for specifying the hour, minutes, seconds, AM or PM, and time zone.
Hidden		The field is not displayed.

Task: Define Validation Rules for Descriptive Flexfield Custom Attributes

Define the validation rules for the custom attributes. For example, one attribute might need to match a specified format, while another attribute might be restricted to a list of values. You use a **value set** to specify the validation rules for an attribute, as described in Section 4.4, "Creating Custom Value Sets." Define each attribute's validation rules and check if value sets exist for those rules or you must create new ones. If you must create a value set, you can either create it before you configure the flexfield or you can create it at the same time that you create or edit a segment.

> **Note:** If you use the Quick Create dialogs to configure the flexfield, the value sets are created automatically. You cannot use existing value sets.

Note that validation for a descriptive flexfield context attribute is somewhat different from validation of global and context-sensitive attributes. While value sets are required for global and context-sensitive attributes, a value set is optional for descriptive flexfield context attributes. If a value set is not defined for a context attribute, then the application validates an input value against the context attribute's values. For example, if you created YES and NO context-attribute values, then you do not need an associated value set if YES and NO are the only valid values. However, if the list of valid values for a context is a superset of the context attribute's values, then a value set is required. For example, suppose that you need custom context-sensitive attributes for a YES value, but you do not need any custom attributes for a NO value. You need to add only a YES context attribute value. Because you do not have a NO context value, the context attribute requires a value set of YES and NO, as both values are valid. A value set is also required when the valid values are a subset of the context values. For example, you might have contexts for several countries, but the list of values from which the end user selects the context value might be confined to the countries in the user's region.

You can use only a table value set or an **independent value set** to validate context values. If you use a table value set, then you cannot reference other flexfield segments in the value set's `WHERE` clause. That is, the `WHERE` clause cannot reference `SEGMENT.`*segment_code* or `VALUESET.`*value_set_code*. For information about table and independent value sets, see Section 4.4, "Creating Custom Value Sets."

When determining an attribute's validation rules, consider the following questions:

- What is the data type — character, date, date and time, or number?
- Does the attribute require any validation beyond data type and maximum length?
- If the data type is character, should the value be restricted to digits, or are alphabetic characters allowed? If a number, should the value be zero-filled?
- If alphabetic characters are allowed, should they automatically be changed to uppercase?
- For numeric values, how many digits can follow the radix separator (in base ten numerical systems, the radix separator is often referred to as the decimal point)?
- Do you want to provide a list of values from which to select a valid value for the attribute? If so, consider the following questions:
  - Can you use an existing application table from which to obtain the list of valid values, or do you need to create a custom list?
  - If you are using an existing table, do you need to limit the list of values using a `WHERE` clause?
  - Does the list of valid values depend on the value in another flexfield attribute?
  - Is the list of valid values a subset of another flexfield attribute's list of values?
- Does the value need to fall within a range?

Task: Define Initial Values for Descriptive Flexfield Custom Attributes

When you configure a flexfield segment for a descriptive flexfield attribute, you can specify a constant to use for setting the initial value. If, in Task: Identify Available Descriptive Flexfield Parameters you identified available parameters, you can use those parameters to set initial values. For every custom attribute, list the constant value or parameter, if any, to use for the initial value.

Task: Plan Attribute Mapping to Oracle Business Intelligence Objects

If a descriptive flexfield has been enabled for Oracle Business Intelligence, then you can use segment labels to map attributes to logical objects. By mapping the attributes to logical objects before you deploy the flexfield, you minimize the steps for importing the flexfield into Oracle Business Intelligence. This mapping also equalizes similar context-sensitive attributes that are from different contexts but are mapped to a single logical object, as described in Task: Incorporate Custom Attributes into Oracle Business Intelligence in Section 4.8, "Integrating Custom Attributes." For information about objects in the logical model, see the "Working with Logical Tables, Joins, and Columns" chapter in the *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition)*.

Use the Manage Segment Labels page to view the preconfigured segment labels and to create new ones, as described in Task: Create Segment Labels in Section 4.5.1, "Configuring Descriptive Flexfields."

For each attribute that you want to make accessible to Oracle Business Intelligence, decide whether you want to map the attribute to a logical object. If a segment label does not exist for the logical object, then decide on a code, name, and description in preparation for adding that label. Choose a code, name, and description that will help end users select the correct label for any attribute that should be mapped to a logical object.

> **Note:** You are not required to map attributes to logical objects, but doing so minimizes the steps for importing the flexfield into Oracle Business Intelligence and helps to equalize context-sensitive segments across contexts, as described in Task: Incorporate Custom Attributes into Oracle Business Intelligence in Section 4.8, "Integrating Custom Attributes."

4.3.2 Planning Extensible Flexfields

If you are configuring an extensible flexfield, complete the following tasks:

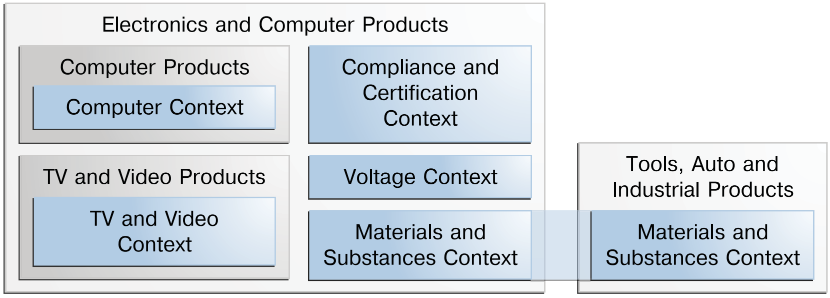
1. Identify the extensible flexfield's category hierarchy structure.
2. Identify existing context values.
3. List custom attributes.
4. Plan categories.
5. Plan the extensible flexfield structure.
6. Define attribute properties.
7. Define validation rules.
8. Define initial values.
9. Plan extensible flexfield security.

Task: Identify the Extensible Flexfield's Category Hierarchy Structure

Most extensible flexfields are shipped with only one category, which is often referred to as the root category, but there are a few extensible flexfields that are preconfigured with category hierarchies. In addition, some Oracle Fusion applications provide user interfaces to create and manage an extensible flexfield's category hierarchy, as described in the product-specific documentation in the Oracle Fusion Applications Help. If a category hierarchy exists for the flexfield, then you can take advantage of the hierarchy to reuse contexts for similar entities, such as similar items in a product catalog.

The Item Extended Attributes flexfield shown in Figure 4–11 is an example of a flexfield that uses the category hierarchy feature to reuse contexts. The flexfield's Electronics and Computers category has been customized to contain contexts for compliance and certification, voltage, and materials and substances, as shown in Figure 4–10. The TV and Video subcategory and the Computer Products subcategory inherit the Electronics and Computer contexts in addition to having their own contexts.

Figure 4–10 Category Hierarchy

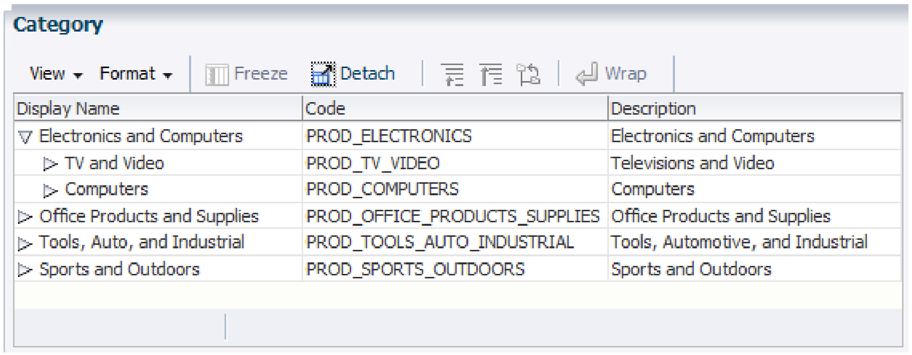

Electronics and Computer Products
Computer Products
Computer Context
Compliance and Certification Context
TV and Video Products
TV and Video Context
Voltage Context
Materials and Substances Context
Tools, Auto and Industrial Products
Materials and Substances Context

Contexts are reusable within a given extensible flexfield. For example, in Figure 4–10 the Materials and Substances context belongs to both the Electronics and Computer Products category and the Tools, Auto, and Industrial Products category.

If you were adding custom attributes to the Item Extended Attributes flexfield, then you would plan how to work them into the existing category hierarchy.

You can use the category hierarchy feature in your flexfield plan, as described in Task: Plan Extensible Flexfield Categories, only if a category hierarchy exists for the flexfield. For example, Figure 4–11 shows the Category region for the Item Extended Attributes flexfield.

Figure 4–11 Category Region



Display Name	Code	Description
▽ Electronics and Computers	PROD_ELECTRONICS	Electronics and Computers
▷ TV and Video	PROD_TV_VIDEO	Televisions and Video
▷ Computers	PROD_COMPUTERS	Computers
▷ Office Products and Supplies	PROD_OFFICE_PRODUCTS_SUPPLIES	Office Products and Supplies
▷ Tools, Auto, and Industrial	PROD_TOOLS_AUTO_INDUSTRIAL	Tools, Automotive, and Industrial
▷ Sports and Outdoors	PROD_SPORTS_OUTDOORS	Sports and Outdoors

To view whether a category hierarchy exists for an extensible flexfield, complete one of the following steps:

- Choose **Setup and Maintenance** from the **Administration** menu in the global area of Oracle Fusion Applications and search for the Manage Extensible Flexfields task. Next, search for and edit the flexfield to see its categories.
- Access the page that contains the flexfield, choose **Highlight Flexfields** from the **Administration** menu, click the **Configure Flexfield** (wrench) icon for the highlighted flexfield, and view the list of categories.

Task: Identify Existing Context Values

Some extensible flexfields have preconfigured context values. To identify an extensible flexfield's existing context values, display the user interface page or pages that contain the flexfield segments and look for region headers, which identify existing contexts. For example, the first region in Figure 4–3 contains the attributes that belong to the Recovery and Recycling context value.

You can also use the Manage Extensible Flexfields task to view the list of configured context values.

If context values have been preconfigured, then consult the product-specific documentation in Oracle Fusion Applications Help for information about the use of these contexts.

Task: List Custom Attributes

List all the custom attributes that you want to add using the extensible flexfield. Later, as described in Task: Define Attribute Properties, you define the attribute details.

Task: Plan Extensible Flexfield Categories

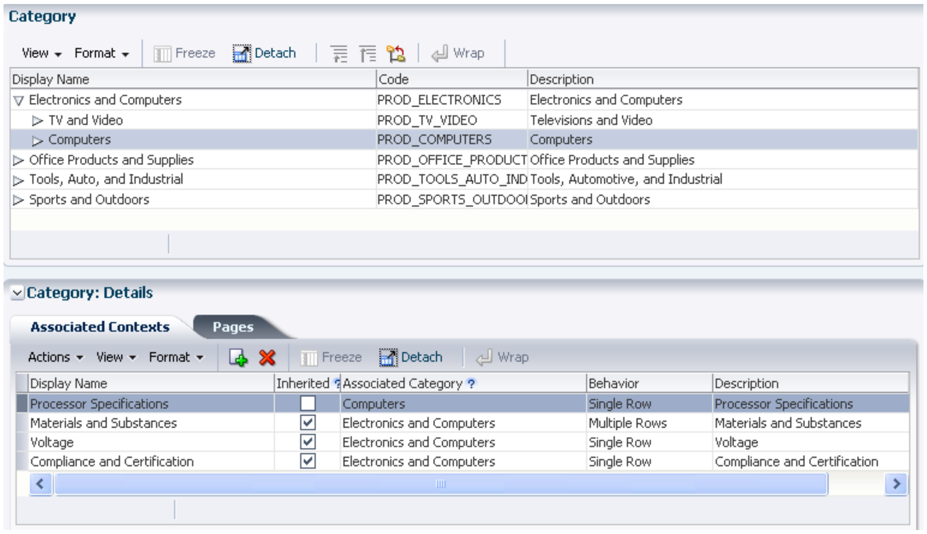
All extensible flexfields have at least one category, but some flexfields have been set up with a hierarchy of categories, as described in Task: Identify the Extensible Flexfield's Category Hierarchy Structure.

How you structure the flexfield configuration depends upon the way in which categories are defined for the flexfield. Most extensible flexfields are preconfigured with one category, and you associate all your contexts and pages with that category. Other extensible flexfields are preconfigured with several categories and you associate your contexts and pages with those categories as instructed by the product-specific documentation in Oracle Fusion Applications Help. A small number of extensible flexfields enable you to configure multiple categories by using provided user interfaces. In these cases, you can take advantage of the inheritance feature described in Task: Identify the Extensible Flexfield's Category Hierarchy Structure to associate a context with more than one category. For example, the Item Extended Attributes flexfield might be set up with the following category hierarchy:

- Electronics and Computers
  - TV and Video
  - Computers

Suppose you want to store voltage information for all electronic and computer items. If you associate a Voltage context with the Electronics and Computers category, then both the TV and Video subcategory and the Computers subcategory inherit the Voltage context from the parent Electronics and Computers category. Figure 4–12 shows the contexts that have been associated with a Computers subcategory. Three of the contexts are inherited from the parent category.

Figure 4–12 Contexts Associated with the Computer Category



Category

Display Name	Code	Description
▽ Electronics and Computers	PROD_ELECTRONICS	Electronics and Computers
▷ TV and Video	PROD_TV_VIDEO	Televisions and Video
▷ Computers	PROD_COMPUTERS	Computers
▷ Office Products and Supplies	PROD_OFFICE_PRODUCT	Office Products and Supplies
▷ Tools, Auto, and Industrial	PROD_TOOLS_AUTO_IND	Tools, Automotive, and Industrial
▷ Sports and Outdoors	PROD_SPORTS_OUTDOOI	Sports and Outdoors

Category: Details

Display Name	Inherited ?	Associated Category ?	Behavior	Description
Processor Specifications	□	Computers	Single Row	Processor Specifications
Materials and Substances	☑	Electronics and Computers	Multiple Rows	Materials and Substances
Voltage	☑	Electronics and Computers	Single Row	Voltage
Compliance and Certification	☑	Electronics and Computers	Single Row	Compliance and Certification

Task: Plan the Extensible Flexfield Structure

Extensible flexfields enable you to group similar custom attributes into contexts, as illustrated in Figure 4–2. A context's attributes are displayed together in a region, and the region's header is the context value. For example, the Item Extended Attributes flexfield might have the following contexts:

- Materials and Substances
- Compliance and Certification
- Voltage

However, for the Position EIT Information flexfield, you might group your custom attributes into the following contexts:

- Educational Requirements
- Certification and License Requirements
- Travel Requirements

To begin the planning process, group the custom attributes into contexts and determine the order in which the attributes should appear.

A context can optionally store multiple rows of data for a single entity, such as a specific job or position. For example, with the Certification and License Requirements context for the Position EIT Information flexfield, you might want to store values for all the certificates and licenses that are required to perform each position. For contexts that store multiple rows, decide how to uniquely identify each row. That is, identify which attributes form a unique key. For example, you might decide that the combination of certificate type and certificate name uniquely identifies a given row for the Certification and License Requirements context.

Next, for each category (or for the single category if the flexfield was not set up with multiple categories), group the category's contexts into logical pages and determine the sequence in which the logical pages should appear.

> **Note:** For hierarchical categories, the child categories inherit the logical pages that are defined for the parent categories.

The following list shows an example plan for custom computer attributes for the Item Extended Attributes flexfield. In this example, the Electronics Information page is inherited from the parent Electronics and Computers category.

- Page: Electronics Information
  - Context: Compliance and Certification, single row
    - ISO 14001 (International Organization for Standardization for an Environmental Management System)
    - ENERGY STAR (energy efficiency guidelines)
    - ROHS (Restriction of the use of certain hazardous substances in electrical and electronic equipment)
  - Context: Voltage, single row
    - Minimum voltage
    - Maximum voltage
    - Current type
  - Context: Materials and Substances, multiple rows
    - Material
    - Contain recyclate
    - Percent unit mass
- Page: Computer Information
  - Context: Processor Specifications, single row
    - Manufacturer
    - CPU type
    - Processor interface
    - Processor class
    - Processor speed
    - Cores

The following list shows a sample plan for the Position EIT Information flexfield:

- Page: Additional Position Information
  - Context: Educational Requirements, single row
    - Level (high school, bachelor, master, MD, Ph.D.)
  - Context: Certification and License Requirements, multiple rows
    - Type (certificate or license)
    - Name (for example, Automotive Service Excellence, NACE International Level II Coating Inspector, Cathodic Protection Specialist)
  - Context: Travel Requirements, single row
    - Overnight travel required
    - International travel required

Task: Define Attribute Properties

For each custom attribute that you want to add, define the attribute properties that are listed in Table 4–1.

Also define the indexed property for each attribute. This indicates whether the attribute should be marked as selectively required in search panels. That is, whether it is one of the attributes for which an end user must enter a value before conducting a search. Note that if you mark an attribute as indexed, you must ask your database administrator to create an index on the segment column that you configure for that custom attribute as described in Section 4.5.2, "Configuring Extensible Flexfields."

Task: Define Validation Rules for Extensible Flexfield Custom Attributes

Define the validation rules for every attribute as described in Task: Define Validation Rules for Descriptive Flexfield Custom Attributes in Section 4.3.1, "Planning Descriptive Flexfields."

Task: Define Initial Values for Extensible Flexfield Custom Attributes

When you configure a flexfield segment for an extensible flexfield attribute, you can specify a constant or a SQL statement to use for setting the initial value. For every custom attribute, list the constant value or SQL statement, if any, to use for the initial value.

If you use an SQL statement, it must be a valid statement that returns only one row and that returns a value of the correct type. You can use two types of SQL statements:

- SQL statement with no binding. For example, select MIN(SALARY) from EMPLOYEES.
- SQL statement with bind variables. You can use the following bind variables in the `WHERE` clause of the SQL statement.
  - **`:{SEGMENT.`**`segment_code`**`}`**: Identifies a segment in the same context.
  - **`:{CONTEXT.`**`context_code`**`;SEGMENT.`**`segment_code`**`}`**: Identifies a segment in a different context. The context must be in the same category or in an ancestor category, and it cannot be a multiple-row context.
  - **`:{VALUESET.`**`value_set_code`**`}`**: Identifies the closest prior segment in the same context that is assigned to the specified value set.
  - **`:{FLEXFIELD.`**`internal_code`**`}`**: Identifies a flexfield.

For information about how to use these bind variables, see Task: Create a Table Value Set in Section 4.4, "Creating Custom Value Sets."

Task: Plan Extensible Flexfield Security

Extensible flexfields enable you to specify view and edit privileges for a context's attributes. If an end user has a view **privilege** for a context, then the context's task flow and region appear in the user interface. Edit privileges enable an end user to edit the context's attribute values. If a user has view privileges but does not have edit privileges, then the context's attribute values are displayed, but the user cannot modify the values or add new rows. For example, you can set the privileges such that everyone can view the attributes but only managers can make changes.

When an end user performs a search, the user interface displays only the attribute values of the contexts for which the user has view privileges. The user is able to perform a search using all attributes for all contexts, regardless of view privileges.

If end users access a context through a web service, an exception is thrown if they perform an action for which they do not have privileges.

Some **data security** resources for extensible flexfields are preconfigured with actions that you can use to specify access privileges. If not, or if you need additional actions, then create the actions as described in the "Managing Oracle Fusion Applications Data Security Policies" chapter in the *Oracle Fusion Middleware Oracle Authorization Policy Manager Administrator's Guide (Oracle Fusion Applications Edition)*. You must create a policy to grant an action to a job **role**. You can optionally create a condition (rule) to filter the instance sets to which the action is granted.

All extensible flexfields have a base data security resource, which typically has a name with an _B suffix. Some extensible flexfields have a translatable option; these flexfields also have a translation data security resource. The translation data security resource, which is a view of a translation table, typically has a name with an _VL suffix. If a flexfield supports the translatable option and has a translation data security resource, you must ensure that you create the action for the appropriate data security resource. If you create a context-specific action for a nontranslatable context, add it to the base data security resource. If you create a context-specific action for a translatable context, add it to the translation data security resource. For information about the translatable option, see Task: Configure the Context in Section 4.5.2, "Configuring Extensible Flexfields."

> **Tip:** As shown in Table 4–3, the name of the base data security resource matches the name of the extensible flexfield's base extension table, which typically has an _B suffix. The name of the translation data security resource matches the name of the translation extension table, but with an _VL suffix instead of the table's _TL suffix. To find the names of the extension tables, view the flexfield in the **Search Results** region in the Manage Extensible Flexfields task and click the **Show Entity Usages** icon in the **Entity Usages** column. For example, the entity usages displayed for the ITEM_EFF flexfield would include ITEM_EFF_B, and ITEM_EFF_TL. Therefore, you would create actions for the ITEM_EFF_B and ITEM_EFF_VL data security resources.

If your security restrictions apply to several contexts, you can create generic actions. At a minimum, create the generic actions for the base data security resource. If the flexfield has a translatable option and you plan to use translatable contexts, then also create the generic actions for the translation data security resource. For example, if the Item flexfield supports the translatable option and has a data security resource ITEM_EFF_VL in addition to the base data security resource ITEM_EFF_B, then create actions for both data security resources, such as EDIT_NONTRANS_ATTRS for ITEM_EFF_B and EDIT_TRANS_ATTRS for ITEM_EFF_VL.

If your security restrictions are more fine-grained, for example, if you would like to secure each context with a different privilege, then you can create more fine-grained actions.

Table 4–4 shows an example of data security policies for the Item flexfield and Table 4–5 shows the privileges for three of the flexfield's contexts. In this example, anyone can view the contexts' attributes, but the edit privileges are restricted as follows:

- Voltage: Editable only by voltage specialists.
- Compliance and Certification: Editable only by compliance specialists.
- Materials and Substances: Only computer specialists can edit these attributes for items in the computer category. Only television specialists can edit these attributes for items in the TV category.

In this example, Materials and Substances is secured by a generic action with a condition applied to restrict access by category. Voltage and Compliance and Certification are secured by actions specific to each context.

Table 4–3 Data Security Resources for the Item Flexfield

Extension Table	Data Security Resource
ITEM_EFF_B	ITEM_EFF_B
ITEM_EFF_TL	ITEM_EFF_VL (defined for the view of the extension table)

Table 4–4 Example Security Policies for the Item Flexfield

Data Security Resource	Policy	Role	Actions	Condition
ITEM_EFF_B	A	VOLTAGE_SPEC	edit_nontrans_ voltage_ctx	All values
ITEM_EFF_VL	B	COMPLIANCE_ SPEC	edit_trans_ compliance_ctx	All values
ITEM_EFF_VL	C	COMPUTER_SPEC	edit_trans_attrs	ComputerCategoryFilter
ITEM_EFF_VL	D	TELEVISION_SPEC	edit_trans_attrs	TVCategoryFilter

Table 4–5 Example Context Privileges for the Item Flexfield

Context	Edit Privilege	View Privilege
Voltage	edit_nontrans_voltage_ctx	NONE
Compliance and Certification	edit_trans_compliance_ctx	NONE
Materials and Substances	edit_trans_attrs	NONE

When you configure a flexfield, you set the privileges for a context at the usage level by selecting actions from the **View Privileges** and **Edit Privileges** dropdown lists as described in Task: Configure Context Usages and Usage Security in Section 4.5.2, "Configuring Extensible Flexfields." By selecting an action from the **View Privileges** dropdown list, you are associating that action with the view operation. Selecting an action from the **Edit Privileges** dropdown list associates that action with the edit operation.

> **Note:** An extensible flexfield context can be associated with one or more flexfield usages. For example, a flexfield might be associated with an ITEM usage, an ITEM_REVISION usage, and an ITEM_SUPPLIER usage. Consult the product-specific documentation to understand the purpose of each usage.

4.4 Creating Custom Value Sets

You use the following types of value sets to control the values that can be stored for the custom attributes:

- Table
- Independent
- Dependent

- Subset
- Format only

A **value set** is a predefined group of values that you can use to validate custom attributes. Different custom attributes in the same flexfield can use the same value set, and custom attributes in different flexfields can share value sets.

Value sets enable you to enforce the following types of data validation:

- List of values: You can use one of the following types of lists to specify the valid values for an attribute:
  - Table column: If the valid values exist in a table column, use a **table value set** to specify the list of values. To limit the valid values to a subset of the values in the table, use a SQL `WHERE` clause. Table value sets also provide some advanced features, such as enabling validation to depend upon custom attributes in the same structure.
  - Custom list: Use an **independent value set** to specify a custom set of valid values. For example, you can use an independent value set of Mon, Tue, Wed, and so forth to validate the day of the week.
  - Dependent custom list: Use a **dependent value set** when the available values in the list and the meaning of a given value depend on which independent value was selected for a prior custom attribute. For example, the valid holidays depend on which country you are in. A dependent value set is a collection of value subsets, with one subset for each value in a corresponding independent value set.

  You can further limit the valid values that an end user can select or enter by specifying format, minimum value, and maximum value.
- Subset: Use a **subset value set** when you want to use a subset of values from an existing independent value set. For example, if you have an independent value set for the days of the week, then a weekend subset can be composed of entries for Saturday and Sunday.
- Format: Use a **format-only value set** when you want to allow end users to enter any value so long as that value conforms to formatting rules. For example, if you specify a maximum length of 3 and numeric-only, then end users can enter 456, but not 4567 nor 45A. You can also specify the minimum and maximum values, whether to right-justify, and whether to zero-fill. With a format-only value set, no other types of validation are applied.
- Range of values: You can use either a format-only, independent, or dependent value set to specify a range of values. For example, you might create a format-only value set with format type of Number where the end user can enter only the values between 0 and 100. Or, you might create a format-only value set with format type of Date where the end user can enter only dates for a specific year (a range of 01–JAN–93 to 31–DEC–93, for example). Because the minimum and maximum values enforce these limits, you need not define a value set that contains each of these individual numbers or dates.

Note: You can use only table and independent value sets to validate context values. The data type must be character and the maximum length of the values being stored must not be larger than the context's column length. If you use a table value set, the value set cannot reference flexfield segments in the value set's `WHERE` clause other than the flexfield segment to which the value set is assigned. You learn about assigning value sets to flexfield segments in Section 4.5, "Configuring Flexfields."

If you are creating an independent, dependent, or subset value set, you must also define the set of valid values. For table, independent, dependent, and subset value sets, you can optionally implement value set **data security**. If the Oracle Fusion applications are running in different locales, you might need to provide different translations for the values and descriptions. For more information, see the "Translating Flexfield and Value Set Configurations" section in the *Oracle Fusion Applications Extensibility Guide for Developers*.

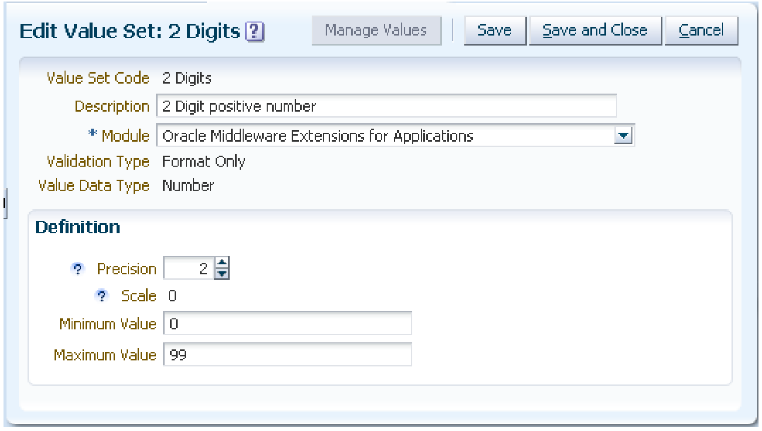
For independent, dependent, and subset value sets, if you need to capture additional information about each valid value, such as its purpose, you can configure the value set to add additional attributes, as described in Task: Add Attributes to a Value Set.

Use the Manage Value Sets task as shown in Figure 4–13 to create and manage value sets.

For more information, see the "Manage Value Sets" section in the *Oracle Fusion Applications Common Implementation Guide*. In Section 4.5, "Configuring Flexfields" you learn how to assign the value sets to the flexfield segments that you preconfigure for your custom attributes as well as how to create value sets at the same time that you create segments for the custom attributes.

Note: If you use the Quick Create dialogs to configure a flexfield, a value set is created automatically. If the display type is a checkbox, radio button group, dropdown list, list of values, or pop-up list of values, an independent value set is created and you specify its values and descriptions (the set of valid values) using the Quick Create UI. If the display type is text field, multi-line text area, or date picker, a format-only value set is created and you specify the format rules, such as length, minimum value, and maximum value, using the Quick Create UI.

Figure 4–13 Edit Value Set Page in the Manage Value Sets Task



Value Set Code: 2 Digits
Description: 2 Digit positive number
* Module: Oracle Middleware Extensions for Applications
Validation Type: Format Only
Value Data Type: Number

Precision: 2
Scale: 0
Minimum Value: 0
Maximum Value: 99

> **Note:** The management of value sets cannot be performed in a sandbox. For a list of the customizations that can be performed in a sandbox, see Chapter 2.2, "Using the Sandbox Manager."

When you change an existing value set, the deployment status for all affected flexfields changes to Edited. You must redeploy all flexfields that use that value set to make the flexfields reflect the changes. The **Usages** tabs show which flexfields are affected by the value set changes. You learn about deploying flexfields in Section 4.7, "Deploying Flexfield Configurations."

Before you begin:

You will need to do the following before you can begin creating custom value sets:

- Ensure that you have specific privileges to access the Manage Value Sets task. Contact your security administrator for details.
- Access the Manage Value Sets task by completing one of the following steps:
  - Choose **Setup and Maintenance** from the **Administration** menu in the global area of Oracle Fusion Applications and search for the task.
  - In the Manage Descriptive Flexfield task or the Manage Extensible Flexfield task, click **Create Value Set** from either the Edit Segment or Create Segment window.

Task: Define Format Specifications

Regardless of which type of validation you use for an attribute, the attribute will most likely require some sort of format specification. Before you create a value set, consider how you will specify the required format. Depending on the validation type and the value data type, you will be able to specify one or more of the options shown in Table 4–6.

Table 4–6 Format Options

Option	Description
Value data type	Character, Number, Date, Date Time.

Table 4–6 (Cont.) Format Options

Option	Description
Value subtype	Text, Translated text, Numeric digits only, Time (20:08), Time (20:08:08). An additional data type specification for the Character data type for the Dependent, Independent, and Format validation types. See the Caution note following this table for information about the Text and Translated text subtypes.
Maximum length	Maximum number of characters or digits for Character data type.
Precision	Maximum number of digits the user can enter.
Scale	Maximum number of digits that can follow the decimal point.
Uppercase only	Lowercase characters automatically changed to uppercase.
Zero fill	Automatic right-justification and zero-filling of entered numbers (affects values that include only the digits 0-9).

Caution: When choosing between a value subtype of Text or Translated text, choose Translated text if your application has more than one language installed, or there is any possibility that you might install additional languages, and you might want to translate the displayed values into the other languages. Choosing the Translated text subtype does not require you to provide translated values now, but you cannot change this option if you decide to provide them later.

Task: Create a Format-Only Value Set

If you do not need to validate input against a list of valid values, then create a value set with the **Validation Type** of Format. You can also use this validation type to create a range specification, such as all numbers between 1 and 100.

Task: Create an Independent Value Set

If you need to validate the input against a custom list of values, where the list is not stored in an application table and the values are not dependent upon nor a subset of another independent value set, then create a value set with a **Validation Type** of Independent.

After you create the value set, define the set of valid values as described in Task: Define the Set of Valid Values.

Note: If the independent value set has dependent value sets, then define the dependent values sets before you define the valid values for the independent value set.

Task: Create a Dependent or Subset Value Set

If you need to validate the input against a custom list of values where the values are dependent upon or are members of an independent value set, then create a value set with a **Validation Type** of Dependent or Subset, as appropriate. You must specify the independent value set on which dependent or subset value set depends.

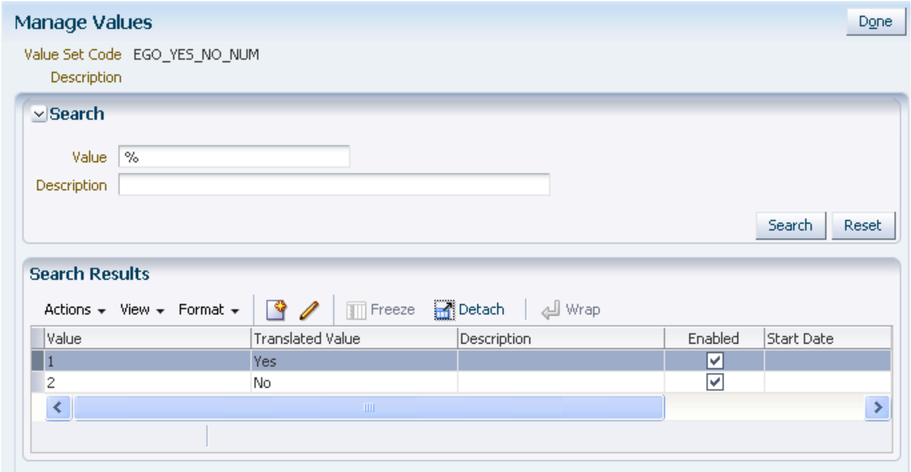
After you create the value set, define the set of valid values as described in Task: Define the Set of Valid Values.

> **Note:** If the independent value set has dependent value sets, then define the dependent values sets before you define the valid values for the independent value set.

Task: Define the Set of Valid Values

Independent, dependent, and subset value sets require a customized list of valid values. Use the Manage Values page as shown in Figure 4–14 to create and manage a value set's valid values and the order in which they appear. To access the Manage Values page, select the value set from the Manage Value Sets page and click **Manage Values**. Alternatively, click **Manage Values** from the Edit Value Set page.

Figure 4–14 Manage Values Page



Manage Values

Value Set Code EGO_YES_NO_NUM

Description

Search

Value: %

Description:

Search Results

Value	Translated Value	Description	Enabled	Start Date
1	Yes		☑	
2	No		☑	

If you are adding a value to a dependent value set, then you must associate it with a value from the parent independent value set.

If you are adding a value to a subset value set, then you must select it from the parent independent set.

If the value set's subtype is Translated text, then you can enter a translated value for a locale by logging in using that locale (or choosing **Set Preferences** from the **Personalization** menu in the global area) and entering the translated value. If you do not provide a translation for a given locale, then the value that was first entered is used for that locale.

Task: Create a Table Value Set

If you need to validate the input against a list of values from an application table, then create a value set with the Table validation type. You define which table you want to use and you specify the column that contains the valid value. You can optionally specify the description and ID columns, a `WHERE` clause to limit the values to use for your set, and an `ORDER BY` clause.

If you specify an ID column, then the flexfield saves the ID value, instead of the value from the value column, in the associated flexfield segment. You learn about assigning value sets to flexfield segments in Section 4.5, "Configuring Flexfields."

If the underlying table supports translations, you can enable the display of translated text by basing the value set's value column on a translated attribute of the underlying table. You should also define an ID column that is based on an attribute that is not language-dependent so that the value's invariant ID (an ID that does not change) is saved in the transaction table. This allows the runtime to display the corresponding translated text from the value column for the runtime session's locale. For more information, see the "Using Multi-Language Support Features" section in the *Oracle Fusion Applications Developer's Guide*.

After you assign a value set to a flexfield, you can use the following bind variables in the `WHERE` clause.

- **`:{SEGMENT.`**`*segment_code*`**`}`**

  This bind variable refers to the ID (if the value set is ID-validated) or value (if not ID-validated) of a segment where `*segment_code*` identifies the segment. The data type of the bind value is the same as the data type of the segment's column.

  For descriptive flexfields, if the segment is global, then the source segment (the segment with the WHERE clause) must be global. Otherwise, it must be in the same context as the source segment. For extensible flexfields, the segment must be in the same context as the source segment.

  The source segment must have a sequence number that is less than the sequence number of the target segment (segment with this bind variable).

  This bind variable is useful when the set of valid values depends on the value in another segment. For example, the values to select from a CITIES table might depend upon the selected country. If SEGMENT1 contains the country value, then the `WHERE` clause for the CITIES table might be `country_code = :{SEGMENT.SEGMENT1}`.

- **`:{CONTEXT.`**`*context_code*`**`;SEGMENT.`**`*segment_code*`**`}`**

  This bind variable, which is valid only for extensible flexfields, refers to the ID (if the value set is ID-validated) or value (if not ID-validated) of a segment that is in a different context than the target segment (the segment with the `WHERE` clause). The `*context_code*` identifies the context and `*segment_code*` identifies the segment. The data type of the bind value is the same as the data type of the segment's column.

  The source context must be a single-row context that is in the same category or ancestor category as the target segment.

  > **Tip:** The target segment should appear in the UI after the source segment to ensure the source segment has a value. If the target segment's context is a single-row context, the source and target segments must be on separate pages and the target page must follow the source page.

  This bind variable is useful when the set of valid values depends on the value of a segment in another context. For example, the values to select from a CERTIFICATION table for a segment in the Compliance and Certification context might depend on the value of the country segment in the Manufacturing context.

- **`:{VALUESET.`**`*value_set_code*`**`}`**

  This bind variable refers to the ID (if the value set is ID-validated) or value (if not ID-validated) of the segment that is assigned to the value set that is identified by the `*value_set_code*`. The data type of the bind value is the same as the data type of the segment's column.

  The segment must have a sequence number that is less than the sequence number of the segment with this bind variable. If more than one segment is assigned to the value set, the closest prior matching segment will be used to resolve the bind expression. A matching segment must exist in the current flexfield context.

This bind variable is useful when the set of valid values depends on the value in another segment and that segment code can vary, such as when the value set is used for more than one context or flexfield. For example, the values to select from a CITIES table might depend upon the selected country. If the value set for the segment that contains the country value is COUNTRIES, then the `WHERE` clause for the CITIES table might be `country_code = :{VALUESET.COUNTRIES}`.

- **`:{FLEXFIELD.`*`internal_code`*`}`**

  This bind variable refers to an internal code of the flexfield in which the value set is used, or to a validation date. The *`internal_code`* must be one of the following:

  – `APPLICATION_ID` — the application ID of the flexfield in which this value set is used. The data type of `APPLICATION_ID` and its resulting bind value is NUMBER.

  – `DESCRIPTIVE_FLEXFIELD_CODE` — the identifying code of the flexfield in which this value set is used. The data type of `DESCRIPTIVE_FLEXFIELD_CODE` and its resulting bind value is VARCHAR2. Note that you use this string for both descriptive and extensible flexfields.

  – `CONTEXT_CODE` — the context code of the flexfield context in which this value set is used. The data type of `CONTEXT_CODE` and its resulting bind value is VARCHAR2.

  – `SEGMENT_CODE` — the identifying code of the flexfield segment in which this value set is used. The data type of `SEGMENT_CODE` and its resulting bind value is VARCHAR2.

  – `VALIDATION_DATE` — the current database date. The data type of `VALIDATION_DATE` and its resulting bind value is DATE.

- **`:{PARAMETER.`*`parameter_code`*`}`**

  This bind variable refers to the value of a flexfield parameter where *`parameter_code`* identifies the parameter. The data type of the resulting bind value is the same as the parameter's data type.

> **Note:** You cannot assign a table value set to a context segment if the `WHERE` clause uses `VALUESET.`*value_set_code* or `SEGMENT.`*segment_code* bind variables.

Task: Add Attributes to a Value Set

For independent, dependent, and subset value sets, you can use the FND_VS_VALUES_B descriptive flexfield to add the attributes to a value set. The additional attributes appear in the Manage Value Sets UI, and can be used to capture additional information about each valid value, such as its purpose. These attributes are typically used to capture internal information. If you need to display the attributes on an application page, you must modify the application to programmatically access them.

To add attributes to a value set, edit the FND_VS_VALUES_B flexfield from the Manage Descriptive Flexfields task, and then click Manage Contexts. Create a new context and use the value set code for the context code. Add the desired attributes as context-sensitive segments. Next, deploy the FND_VS_VALUES_B flexfield, sign out, and then sign back in to see the new attributes for that value set in the Manage Value Sets UI.

Task: Implement Value Set Data Security

If you need to control end user access to the values in an independent, dependent, subset, or table value set, then you can create **data security** policies for that value set. These policies enable you to specify what values each end user can enter and view, based on that user's **role**. End users will not be able to enter values for which they do not have access, and only the values they have access to appear in the list of values for the associated segments. Any security rules that you define for a value set affect every segment that uses the value set.

A data security resource identifies the data that must be secured by the policies, which in this case is the value set. Value set data security follows a "deny all, allow some" approach. This means that access to all of a resource's data is denied by default and, based on policies, end users have access only to the data that is appropriate for their role. Use conditions to define the data that can be accessed and use policies to identify the roles that have the authority to access the data that is identified by the condition.

You can implement value set security after you create the value set, or you can implement it later by editing the value set. To implement security, select the **Security Enabled** checkbox and then provide the data security resource name. You can enter the name of an existing resource or type the name of a data security resource that you want to create. This will be the name used for the data security resource in the data security system. The name typically matches the code value for the value set.

> **Note:** You cannot edit the data security resource name after you save your changes.

After you save your changes to the data security resource name, you can optionally click the **Edit Data Security** button to access the Edit Data Security page, provided that you have access to the Manage Database Security Policies task. On the Edit Data Security page, you can specify conditions, such as filters or SQL predicates, and you can define policies where you associate roles with conditions. You can use a filter for simple conditions. For more complex conditions, use a SQL predicate.

The data security conditions and policies that you define are similar to those that you can define for business objects. Value set data security policies differ in the following ways:

- You can grant only read access to end users.You cannot specify any other action.
- When defining a condition that is based on a SQL predicate, use `VALUE`, `VALUE_NUMBER`, `VALUE_DATE`, `VALUE_TIMESTAMP`, or `VALUE_ID` to reference the value from a dependent, independent, or subset value set. For table value sets, use a table alias to define the table, such as `&TABLE_ALIAS.category_id=70`.

For more information about data security resources, roles, conditions, and policies, see the "Managing Oracle Fusion Applications Data Security Policies" chapter in the *Oracle Fusion Middleware Oracle Authorization Policy Manager Administrator's Guide (Oracle Fusion Applications Edition)*.

4.5 Configuring Flexfields

Use the Manage Extensible Flexfields task to configure an extensible flexfield. For descriptive flexfields, use either the Manage Descriptive Flexfields task or the Quick Create dialogs.

Note: Do not use JDeveloper to customize flexfields. If you require flexfield changes that you cannot accomplish using the Manage Flexfields tasks or the Manage Value Sets tasks as described in this chapter, contact My Oracle Support at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

4.5.1 Configuring Descriptive Flexfields

Use the Manage Descriptive Flexfields task or the Quick Create dialogs to configure a descriptive flexfield. Both the Manage Descriptive Flexfield task and the Quick Create dialogs are available from the Highlight Flexfields UI. The Manage Descriptive Flexfields task is also available from the Setup and Maintenance work area.

When you configure a descriptive flexfield, you define the global segments, the context segment, and the context-sensitive segments. For each segment, you define its display properties and you specify how to validate its values.

If the descriptive flexfield has been enabled for Oracle Business Intelligence, you can enable segments for Oracle Business Intelligence and you can select segment labels for the segments. For more information, see Task: Plan Attribute Mapping to Oracle Business Intelligence Objects in Section 4.3.1, "Planning Descriptive Flexfields."

Note: A descriptive flexfield might be used for more than one application table. For example, a flexfield might be associated with a USER table and a USER_HISTORY table. These associations are called flexfield usages. When you configure a flexfield, the configuration applies to all its usages.

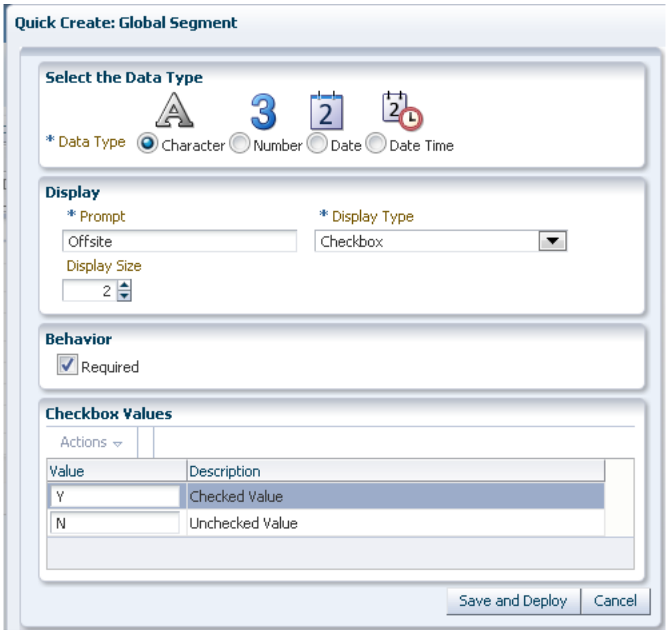
If the Oracle Fusion applications are running in different locales, then you might want to provide different translations for the translatable text, such as prompts and descriptions. To create the translations, log in with each of the different locales (or choose **Set Preferences** from the **Personalization** menu in the global area to set the locale) and change the text to the translated text for that locale. For more information, see the "Translating Flexfield and Value Set Configurations" section in the *Oracle Fusion Applications Extensibility Guide for Developers*.

For more information about using the Manage Descriptive Flexfields task, see the "Manage Descriptive Flexfields" section in the *Oracle Fusion Applications Common Implementation Guide*.

4.5.1.1 Configuring Descriptive Flexfields Using Quick Create Dialogs

For basic configurations, you can use the three Quick Create dialogs — Create Context Value, Quick Create: Global Segment, and Quick Create: Context-Sensitive Segment — to quickly add and configure a context value or segment from the page on which the flexfield appears. Figure 4–15 shows the Quick Create: Global Segment dialog.You access the Quick Create dialogs from the Highlight Flexfields UI.

Figure 4–15 Quick Create: Global Segment Dialog



Quick Create: Global Segment

Select the Data Type

* Data Type ● Character ○ Number ○ Date ○ Date Time

Display

* Prompt: Offsite

* Display Type: Checkbox

Display Size: 2

Behavior

☑ Required

Checkbox Values

Value	Description
Y	Checked Value
N	Unchecked Value

When you use a Quick Create dialog to create a segment, the segment code, name, description, table column, and sequence number are set automatically.

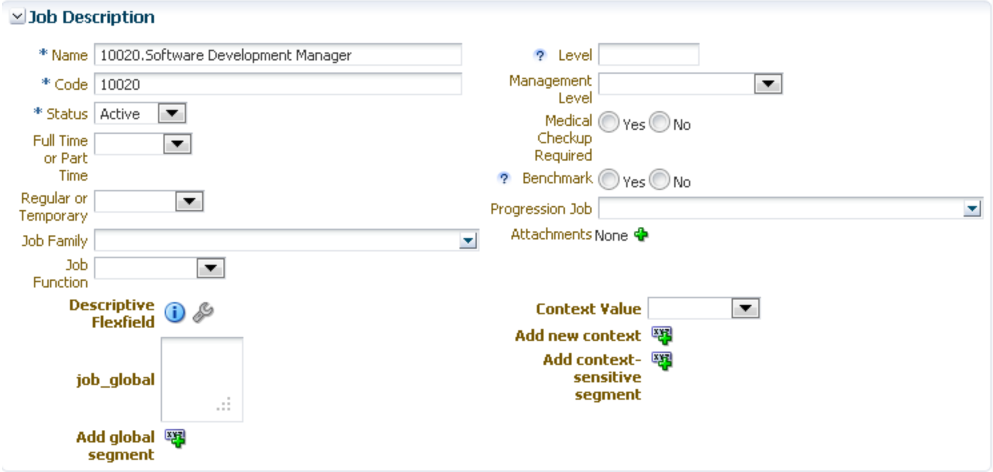
To change any default configuration that is editable, such as sequence number, name, and description, or to configure advanced options such a business-intelligence label, click the segment's **Edit Descriptive Flexfield** (wrench) icon in the Highlight Flexfields UI to access the Manage Descriptive Flexfields task for the segment.

Before you begin:

You will need to do the following before you can begin configuring the descriptive flexfield:

- Plan your flexfield configuration as described in Section 4.3.1, "Planning Descriptive Flexfields."
- Compile a list of the UI pages that are affected by the descriptive flexfield as well as other artifacts in the Oracle Fusion Middleware technology stack. See Section 4.8, "Integrating Custom Attributes" for more information about how flexfield configuration affects these artifacts. Using this list, plan how you will integrate and test your flexfield configuration
- Ensure that you have the necessary privileges to access the Highlight Flexfields UI. Contact your security administrator for details.
- Open the page that contains the flexfield and choose **Highlight Flexfields** from the **Administration** menu to identify the flexfield and to display the **Add global segment**, **Add new context**, and **Add context-sensitive segment icons**, as shown in Figure 4–16.

Figure 4–16 Page After Highlight Flexfields is Chosen From the Administration Menu



* Name 10020.Software Development Manager

* Code 10020

* Status Active

Attachments None

Note: Some Oracle Fusion applications provide product-specific tasks for configuring and managing certain flexfields, such as the Manage Purchasing Descriptive Flexfields task. See the product-specific documentation in Oracle Fusion Applications Help for information about product-specific flexfield management tasks.

Task: Use the Quick Create Dialogs to Add Basic Segments

To add a context value, click the **Add new context** icon that appears next to the **Context** field. To add a global segment, click the **Add global segment** icon. Click the **Add context-sensitive segment** icon to add a context sensitive segment.

> **Tip:** After you add a context value, refresh the page to see the new value.

A value set is created automatically for a segment. The type of value set that is created depends upon the display component that you select.

- Radio button group, checkbox, dropdown list, list of values, or pop-up list of values: An independent value set is created for the segment. You can enter the valid values, their descriptions, and the default value for the set.
- Text field, multi-line text area, or date picker: A format-only value set is created for the segment. You can specify the formatting constraints, such as minimum and maximum values.

Click **Save and Deploy** to save your changes and to deploy the flexfield configuration.

Use the Manage Descriptive Flexfield task to edit the segments that you create using Quick Create dialogs.

Note: You must log out and log in again to see your changes.

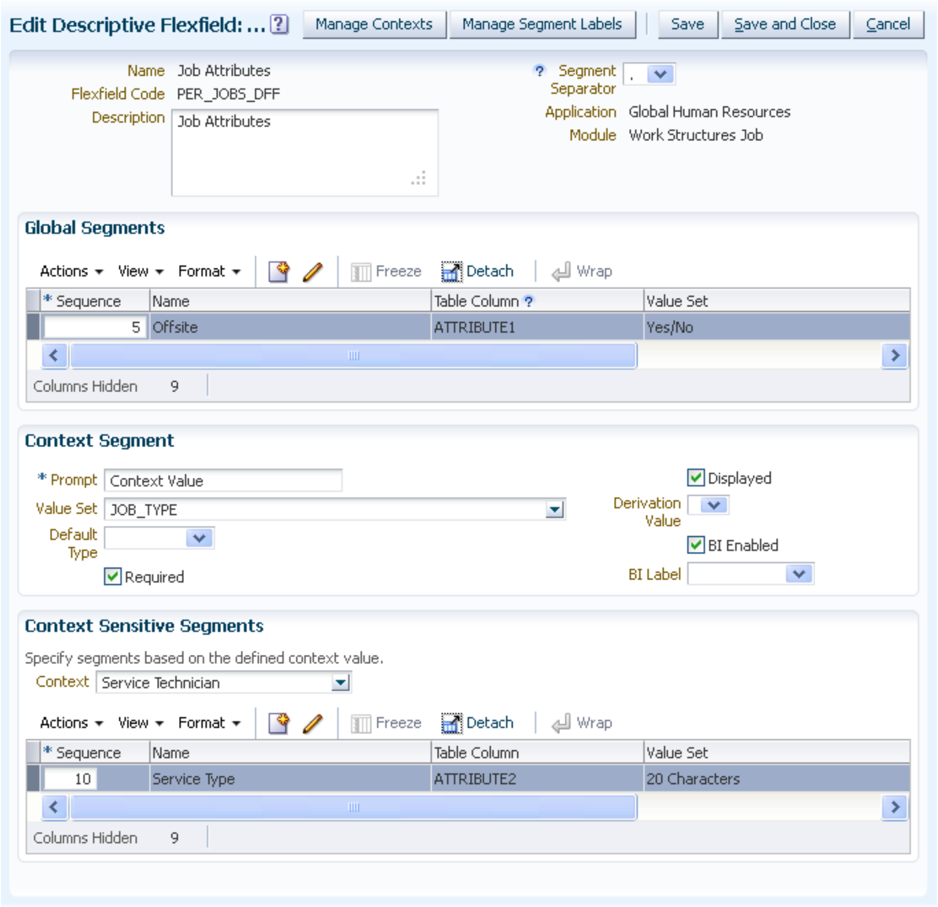
4.5.1.2 Configuring Descriptive Flexfields Using the Manage Descriptive Flexfields Task

The Manage Descriptive Flexfields task enables you to define the global segments, the context segment, and the context-sensitive segments. For each segment, you can define its display properties, its initial default value, and how to validate its values. You can also use this task to business-intelligence enable the flexfield and to configure BI labels.

You can access the Manage Descriptive Flexfields task from either the Highlight Flexfields UI or from the Setup and Maintenance work area.

Figure 4–17 shows the Edit Descriptive Flexfield page, which you use to define the context, add segments, and manage contexts, and manage segment labels.

Figure 4–17 Edit Descriptive Flexfield Page in the Manage Descriptive Flexfields Task



Name Job Attributes
Flexfield Code PER_JOBS_DFF
Description Job Attributes
Segment Separator .
Application Global Human Resources
Module Work Structures Job

Global Segments

* Sequence	Name	Table Column	Value Set
5	Offsite	ATTRIBUTE1	Yes/No

Columns Hidden 9

Context Segment

* Prompt Context Value
Value Set JOB_TYPE
Default Type
☑ Required
☑ Displayed
Derivation Value
☑ BI Enabled
BI Label

Context Sensitive Segments

Specify segments based on the defined context value.
Context Service Technician

* Sequence	Name	Table Column	Value Set
10	Service Type	ATTRIBUTE2	20 Characters

Columns Hidden 9

Before you begin:

You will need to do the following before you can begin configuring the descriptive flexfield:

- Plan your flexfield configuration as described in Section 4.3.1, "Planning Descriptive Flexfields."
- Optionally, create the required value sets as described in Section 4.4, "Creating Custom Value Sets." If you do not create the value sets before hand, you can create them at the time that you add the segments.
- Compile a list of the UI pages that are affected by the descriptive flexfield as well as other artifacts in the Oracle Fusion Middleware technology stack. See Section 4.8, "Integrating Custom Attributes" for more information about how flexfield configuration affects these artifacts. Using this list, plan how you will integrate and test your flexfield configuration

- Ensure that you have the necessary privileges to access the Manage Descriptive Flexfields task. Contact your security administrator for details.
- Access the Manage Descriptive Flexfields task by completing one of the following steps:
  - Choose **Setup and Maintenance** from the **Administration** menu in the global area of Oracle Fusion Applications and search for the Manage Descriptive Flexfields task.
  - Access the page that contains the flexfield, choose **Highlight Flexfields** from the **Administration** menu, and click the **Edit Descriptive Flexfield** (wrench) icon for the highlighted flexfield.

> **Note:** Some Oracle Fusion applications provide product-specific tasks for configuring and managing certain flexfields, such as the Manage Purchasing Descriptive Flexfields task. See the product-specific documentation in Oracle Fusion Applications Help for information about product-specific flexfield management tasks.

Task: Create Segment Labels

If the descriptive flexfield has been enabled for Oracle Business Intelligence and you want to create segment labels for the flexfield, click **Manage Segment Labels** to create the labels. For more information, see Task: Plan Attribute Mapping to Oracle Business Intelligence Objects in Section 4.3.1, "Planning Descriptive Flexfields."

Task: Configure Global Segments

Using the information that you gathered in your planning stage, configure a global segment for every global attribute that you identified in Task: Plan the Descriptive Flexfield Structure in Section 4.3.1, "Planning Descriptive Flexfields." For each segment, provide the identifying information, the initial default value, and the display properties.

> **Tip:** A flexfield column is assigned to a new segment automatically, but you can change the assignment. If you need to set a specific column assignment for a segment, create that segment first to ensure that the intended column is not automatically assigned to a different segment.

If the value set to be used to validate the segment already exists, select it from the dropdown list. Otherwise, click **Create Value Set** and configure it.

Select **BI Enabled** if you want to make the segment available to Oracle Business Intelligence. Select a label from the **BI Label** dropdown list if you want to equalize the segment with segments in other contexts or you want to map the segment to a logical object as described in Task: Plan Attribute Mapping to Oracle Business Intelligence Objects in Section 4.3.1, "Planning Descriptive Flexfields."

Note: The segment code is used in the flexfield's element in the XML schema for web services. You can maximize the readability of the schema by naming segment codes with a leading alphabetic character followed by alphanumeric characters. The use of spaces, underscores, multibyte characters, and leading numeric characters, which are all encoded in XML schemas, make the codes in the schema element difficult to read.

Tip: When you create segments for your attributes, you specify the sequence number for the segment. This sequence affects the order in which the attribute is displayed on the page. Consider numbering the segments in multiples, such as 4, 5, or 10, to make it easy to insert new attributes.

Task: Configure the Context Segment

On the Edit Descriptive Flexfield page, specify the prompt for the context segment, whether the context segment should be displayed, and whether a value is required.

Select **BI Enabled** if you want to make the context segment available to Oracle Business Intelligence. Select a label from the **BI Label** dropdown list if you want to map the segment to a logical object as described in Task: Plan Attribute Mapping to Oracle Business Intelligence Objects in Section 4.3.1, "Planning Descriptive Flexfields."

As explained in Task: Define Validation Rules for Extensible Flexfield Custom Attributes in Section 4.3, "Planning Your Flexfields," you do not need to specify a value set if the set of valid values is the same as the set of context values. If you need to associate a context with a value set, then the value set must be an independent or table value set. The data type must be Character and the maximum length of the values being stored must not be larger than the context's column length.

If the value set to be used to validate the segment already exists, select it from the dropdown list. Otherwise, to create a value set, click **Create Value Set** and configure it.

Note: You cannot assign a table value set to a context segment if the `WHERE` clause uses `VALUESET.`*value_set_code* or `SEGMENT.`*segment_code* bind variables.

Task: Configure Contexts

Using the list of valid context values that you prepared in the planning stage, access the Manage Contexts page and specify for each context value a context code, description, and name, and add its context-sensitive segments.

For each context-sensitive segment that you add, provide the identifying information, the column assignment, the initial default value, and the display properties.

Tip: A flexfield column is assigned to a new segment automatically, but you can change the assignment. If you need to set a specific column assignment for a segment, create that segment first to ensure that the intended column is not automatically assigned to a different segment.

If the value set to be used to validate the segment already exists, select it from the dropdown list. Otherwise, click **Create Value Set** and configure it.

Select **BI Enabled** if you want to make the segment available to Oracle Business Intelligence. Select a label from the **BI Label** dropdown list if you want to equalize the segment with segments in other contexts or you want to map the segment to a logical object as described in Task: Plan Attribute Mapping to Oracle Business Intelligence Objects in Section 4.3.1, "Planning Descriptive Flexfields."

Note: The context and segment codes are used in the flexfield's element in the XML schema for web services. You can maximize the readability of the schema by naming context and segment codes with a leading alphabetic character followed by alphanumeric characters. The use of spaces, underscores, multibyte characters, and leading numeric characters, which are all encoded in XML schemas, make the codes in the schema element difficult to read.

Tip: When you create segments for your attributes, you specify the sequence number for the segment. This sequence affects the order in which the attribute is displayed on the page. Before you begin creating segments, you might want to plan how you will number sequences. Consider numbering the segments in multiples, such as 4, 5, or 10, to make it easy to insert new attributes.

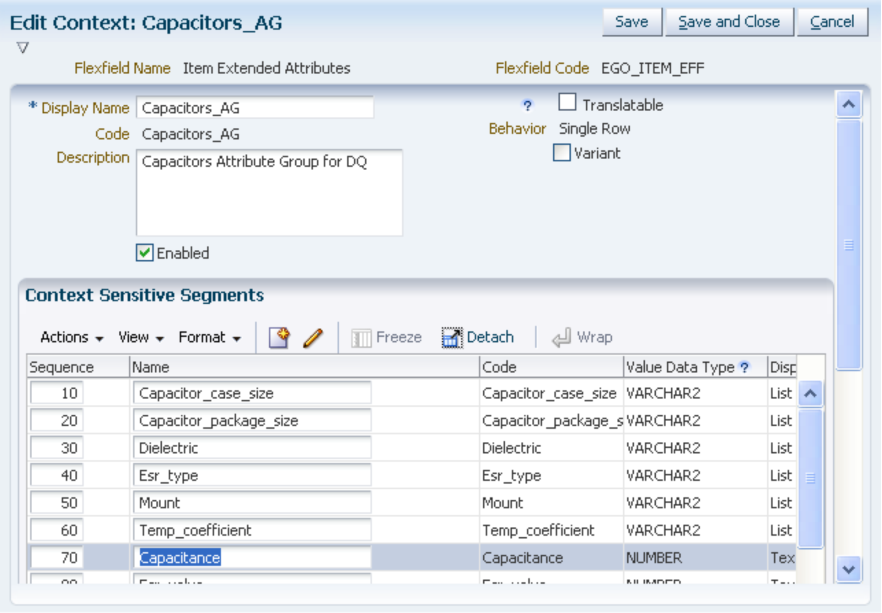
4.5.2 Configuring Extensible Flexfields

Use the Manage Extensible Flexfields task shown in Figure 4–18 to configure an extensible flexfield. You can access the Manage Extensible Flexfields task from the Highlight Flexfields UI as well as from the Setup and Maintenance work area.

Configure an extensible flexfield by defining its contexts and associated context-sensitive segments and usages, defining the access privileges for each usage, associating the context with categories and pages, and defining the display properties, initial value, and value validation for each context-sensitive segment.

If the Oracle Fusion applications are running in different locales, you might want to provide different translations for the translatable text, such as prompts and descriptions. To create the translations, log in with each of the different locales (or choose **Personalization** from the **Set Preferences** menu in the global area to set the locale) and change the text to the translated text for that locale. For more information, see the "Translating Flexfield and Value Set Configurations" section in the *Oracle Fusion Applications Extensibility Guide for Developers*.

Figure 4–18 Edit Context Page in the Manage Extensible Flexfields Task



Flexfield Name Item Extended Attributes

Flexfield Code EGO_ITEM_EFF

* Display Name Capacitors_AG

Code Capacitors_AG

Description Capacitors Attribute Group for DQ

☑ Enabled

□ Translatable

Behavior Single Row

□ Variant

Context Sensitive Segments

Sequence	Name	Code	Value Data Type	Disp
10	Capacitor_case_size	Capacitor_case_size	VARCHAR2	List
20	Capacitor_package_size	Capacitor_package_s	VARCHAR2	List
30	Dielectric	Dielectric	VARCHAR2	List
40	Esr_type	Esr_type	VARCHAR2	List
50	Mount	Mount	VARCHAR2	List
60	Temp_coefficient	Temp_coefficient	VARCHAR2	List
70	Capacitance	Capacitance	NUMBER	Tex
80	Esr_value	Esr_value	NUMBER	Tex

For more information, see the "Manage Extensible Flexfields" section in the *Oracle Fusion Applications Common Implementation Guide*.

Before you begin:

You will need to do the following before you can begin configuring an extensible flexfield:

- Plan your flexfield configuration as described in Section 4.3.2, "Planning Extensible Flexfields."
- Optionally, create the required value sets as described in Section 4.4, "Creating Custom Value Sets." If you do not create the value sets before hand, you can create them at the time that you add the segments.
- Compile a list of the UI pages that are affected by the extensible flexfield as well as other artifacts in the Oracle Fusion Middleware technology stack. See Section 4.8, "Integrating Custom Attributes" for more information about how flexfield configuration affects these artifacts. Using this list, plan how you will integrate and test your flexfield configuration
- Ensure that you have the necessary privileges to access the Manage Extensible Flexfields task. Contact your security administrator for details.
- Access the Manage Extensible Flexfields task by completing one of the following steps:
  - Choose **Setup and Maintenance** from the **Administration** menu in the global area of Oracle Fusion Applications and search for the Manage Extensible Flexfields task.
  - Access the page that contains the flexfield, choose **Highlight Flexfields** from the **Administration** menu, and click the **Configure Flexfield** (wrench) icon for the highlighted flexfield.

> **Note:** Some Oracle Fusion applications provide product-specific tasks for configuring and managing certain flexfields, such as the Manage Orchestration Extensible Flexfields task. See the product-specific documentation in Oracle Fusion Applications Help for information about product-specific flexfield management tasks.

Task: Configure the Context

Using the list of contexts that you prepared in the planning stage, access the Manage Contexts page and create the desired contexts. Create the context-sensitive segments for each context and provide the identifying information, the column assignment, the initial default value, and the display properties.

> **Tip:** A flexfield column is assigned to a new segment automatically, but you can change the assignment. If you need to set a specific column assignment for a segment, create that segment first to ensure that the intended column is not automatically assigned to a different segment.

If the value set to be used to validate a segment already exists, select it from the dropdown list. Otherwise, click **Create Value Set** and configure it. If the value set is a table value set, independent value set, dependent value set, or subset value set, optionally select **Show Value Description** to display the description of the selected value to the right of the segment.

Some extensible flexfields have a **Translatable** option. If the context will be used to store free-form user-entered text in the language of the user's locale, and if different translations of that text can be stored for other languages, then select **Translatable**. If you select **Translatable**, then the context can have only format-only validated segments.

> **Note:** The context and segment codes are used in the flexfield's element in the XML schema for web services. You can maximize the readability of the schema by naming context and segment codes with a leading alphabetic character followed by alphanumeric characters. The use of spaces, underscores, multibyte characters, and leading numeric characters, which are all encoded in XML schemas, make the codes in the schema element difficult to read.

> **Tip:** When you create segments for your attributes, you specify the sequence number for the segment. This sequence affects the order in which the attribute is displayed on the page. Consider numbering the segments in multiples, such as 4, 5, or 10, to make it easy to insert new attributes.

Task: Configure Context Usages and Usage Security

Typically, an extensible flexfield has one usage. If a flexfield has more than one usage, you can associate a context with several usages. Consult the product-specific documentation in Oracle Fusion Applications Help to understand the purpose of each usage.

Using the usage security information that you gathered from Task: Plan Extensible Flexfield Security in Section 4.3, "Planning Your Flexfields," select the appropriate actions from the **View Privileges** and **Edit Privileges** dropdown lists for each usage that you associated with the context. Select **None** to specify that no special privileges should be enforced.

> **Note:** If you change a privilege for a usage after you deploy the flexfield, you must redeploy the flexfield for the change to take place.

Some flexfields might have other usage fields. Consult the product-specific documentation from Oracle Enterprise Repository for Oracle Fusion Applications for more information about the additional usage fields.

Task: Configure Categories and Category Details

While most extensible flexfields have a single category, some extensible flexfields have multiple categories. When an extensible flexfield has multiple categories, you can further group the contexts into categories. Some flexfields provide an activity or task for creating your own categories. Consult the product-specific documentation in Oracle Fusion Applications Help to determine whether you can create categories for the flexfield.

Task: Associate Contexts with a Category

Most extensible flexfields have a single category that contains all the flexfield's contexts. Some flexfields enable you to work with multiple categories. To associate contexts with a category, select the category on the Edit Extensible Flexfield page. Then, in the Associated Contexts tab, add the desired contexts to the selected category.

Task: Create Logical Pages for a Category

Using the information from your plan, group the contexts into logical pages and determine the sequence in which the logical pages should appear. To build the logical pages, select the category on the Edit Extensible Flexfield page. Then, in the Pages tab, add the desired contexts and specify their sequence.

4.6 Validating Flexfield Configurations

You can verify that a flexfield's metadata is complete and correctly configured by choosing **Validate Metadata** from the **Actions** menu in either the Maintain Extensible Flexfields task or the Maintain Descriptive Flexfields task. The validations that this action performs are the same validations that are performed when you deploy a flexfield.

4.7 Deploying Flexfield Configurations

You must deploy a flexfield before you can access the custom attributes from the user interface and other parts of the Oracle Fusion Middleware technology stack. You use the appropriate flexfield management task — Manage Descriptive Flexfields or Manage Extensible Flexfields — to deploy a flexfield.

> **Note:** If you use the Quick Create dialogs to configure a flexfield, the flexfield is deployed when you click **Save and Deploy**.

When you deploy descriptive or extensible flexfields, the following occurs:

- The deployment process validates the flexfield metadata. If any errors are found, then a popup window is displayed with a list of the errors that were encountered, and the flexfield is not deployed. Note that the flexfield's status is not changed if metadata errors are encountered.

- The deployment process generates ADF Business Components and ADF Faces runtime artifacts that are based on the flexfield configurations.
- The deployment process adds the custom attributes to the Web Services Description Language (WSDL) schemas that are exposed by Oracle ADF services and that are used by SOA composites. For information about SOA, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.
- If the flexfield is enabled for business intelligence, the deployment process redeploys the flexfield's business intelligence artifacts.
- The web services that expose the business object's data also expose the business object's flexfield segment data.
- End users see the new custom attributes the next time they log in to the application. The custom attributes will appear on all user interface pages that contain the flexfield, and in the search screens and Excel worksheets with ADF Desktop Integration in which they have been integrated.

> **Note:** Some products provide tasks for creating and maintaining categories for an extensible flexfield. Some of these tasks enable the end user to disable a category while they are setting it up. In the Edit Extensible Flexfield page, the **Enabled** column in the Category table indicates which categories are enabled. When you deploy an extensible flexfield that has a disabled category, that category and its descendant categories are not deployed. Contexts and their segments are deployed only if they belong to at least one enabled category.

Before you deploy the flexfield to the full test environment, deploy the flexfield configurations to the **flexfield sandbox** by choosing the Deploy Flexfield to Sandbox action. The changes that you deploy to a sandbox are isolated from the full test environment and can be seen only by those who make the flexfield sandbox active in their session. After you are satisfied with the changes in the sandbox, deploy the changes to the full test environment. Note that this feature if not available if you use the Quick Create dialogs to deploy the flexfield.

In addition to the Deploy Flexfield and Deploy Flexfield to Sandbox actions, extensible flexfields provide a Deploy Offline action. You must use this action to deploy extensible flexfields that have more than 30 categories. You can also use this action if you want to deploy several extensible flexfields, or if you want to continue working in your session without having to wait for a deployment to complete.

After deployment, the custom attributes are available for incorporating into the SOA infrastructure, such as business process and **business rule** integration. For example, you can now write business rules that depend on the custom attributes.

Descriptive flexfields that are enabled for business intelligence are available for integrating with business intelligence (BI) technology, such as Oracle BI Enterprise Edition (Oracle BI EE) and Oracle Essbase.

For more information about flexfield deployment and flexfield sandboxes, see the "Flexfield Deployment" section in the *Oracle Fusion Applications Common Implementation Guide*. For more information about integrating the deployed flexfield into the Oracle Fusion Middleware technology stack, see Section 4.8, "Integrating Custom Attributes."

Before you begin:

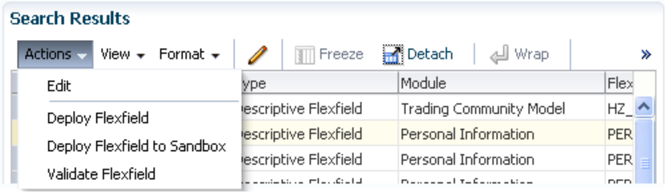
You will need to do the following before you can begin deploying a descriptive or extensible flexfield:

- Configure the flexfield as described in Section 4.5, "Configuring Flexfields."
- Ensure that you have the necessary privileges to access the flexfield management task. Contact your security administrator for details.
- Access the appropriate flexfield management task — Manage Descriptive Flexfields or Manage Extensible Flexfields — by choosing **Setup and Maintenance** from the **Administration** menu in the global area of Oracle Fusion Applications and searching for the task.
- Use the Search region to access the desired flexfield.

Task: Deploy a Flexfield to a Sandbox

Before you deploy the flexfield configuration to the full test environment, deploy the flexfield to the flexfield sandbox and test the configuration. To deploy a flexfield to a sandbox, select the flexfield in the Search Results region, and choose **Deploy Flexfield to Sandbox** from the **Actions** menu, as shown in Figure 4–19. A progress window shows when the process has completed and the completion status.

Figure 4–19 Deploying a Flexfield to a Sandbox



ype	Module	Flex
escriptive Flexfield	Trading Community Model	HZ_
escriptive Flexfield	Personal Information	PER
escriptive Flexfield	Personal Information	PER

The sandbox to which you deploy the flexfield is different from the standard sandboxes that are described in Section 2.2, "Using the Sandbox Manager." Each flexfield has its own sandbox. You do not need to create a flexfield sandbox because the deployment process manages flexfield sandbox creation. When you deploy to the flexfield sandbox, a dialog box shows the name of the flexfield sandbox, and that flexfield sandbox is set as your current active sandbox.

> **Note:** When you deploy to the flexfield sandbox, the `CreatedBy` value is set to `FlexfieldDeployment`.

When you next log in to the application, you can see the updated flexfield configurations (you must log out and log back in to see the changes). The Oracle Fusion Applications global area displays your current session sandbox. When you hover over the sandbox name, the sandbox details appear. To exit a sandbox, hover over the sandbox name and click **Exit Sandbox**. To view additional details about the sandbox, hover over the sandbox name and click **More Details**.

To make an existing flexfield sandbox active for your session, access the flexfield sandbox from the Manage Sandboxes page, select the flexfield sandbox, and click **Set as Active**. For information about accessing this page, see Section 2.2, "Using the Sandbox Manager."

After you are satisfied with the changes, deploy the flexfield to the full test environment as described in Task: Deploy a Flexfield to the Full Test Environment.

Task: Deploy a Flexfield to the Full Test Environment

To deploy a flexfield to the full test environment, select the flexfield in the Search Results region, and choose **Deploy Flexfield** from the **Actions** menu.

A progress window shows when the process has completed and the completion status. If a sandbox exists for the flexfield, then this sandbox is deleted automatically after the flexfield successfully deploys to the full test environment.

When you next log in to the application, you can see the updated flexfield configurations (you must log out and log back in to see the changes).

Task: Deploy Extensible Flexfields Offline

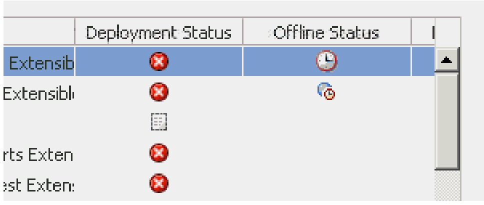
When an extensible flexfield has more than 30 categories, you must deploy the flexfield offline. You can also use this action if you want to deploy several extensible flexfields, or if you want to continue working in your session without having to wait for a deployment to complete.

To add an extensible flexfield to your offline deployment queue, select the flexfield and click **Deploy Offline**. The **Queued** icon appears in the Offline Status column, as shown in the second row in Figure 4–20. You can add several flexfields to your queue, one after the other. The flexfields are deployed, one at a time, in the order that you deploy them to the queue. To remove an extensible flexfield from your queue, select the queued flexfield and click **Cancel Offline**. You cannot remove flexfields from other users' queues.

While an extensible flexfield is in an offline deployment process, the **Deploying** icon displays in the Offline Status column, as shown in the first row in Figure 4–20. After an extensible flexfield completes the offline deployment process, its offline status becomes blank and its deployment status indicates if the deployment was successful or not.

> **Note:** The Offline Status column does not refresh automatically when a queued flexfield begins the offline deployment process or when its offline deployment process completes. To display the current offline status, you must perform a new search in the Manage Extensible Flexfields task, or exit and reenter the task.

Figure 4–20 Offline Status Icons

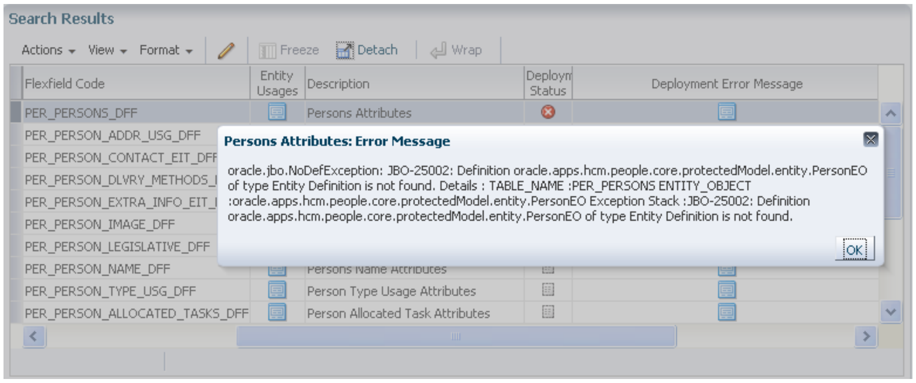


Task: Check Deployment Status

The Deployment Status column in the Search Results region shows the current status for each flexfield.

If a deployment error occurs, then the Deployment Error Message column in the Search Results region provides details about the error, as shown in Figure 4–21.

Figure 4–21 Deployment Error Message



Flexfield Code	Entity Usages	Description	Deploym Status	Deployment Error Message
PER_PERSONS_DFF	▤	Persons Attributes	⊗	▤
PER_PERSON_ADDR_USG_DFF				
PER_PERSON_CONTACT_EIT_DFF				
PER_PERSON_DLVRY_METHODS_I				
PER_PERSON_EXTRA_INFO_EIT_I				
PER_PERSON_IMAGE_DFF				
PER_PERSON_LEGISLATIVE_DFF				
PER_PERSON_NAME_DFF	▤	Persons Name Attributes	▤	▤
PER_PERSON_TYPE_USG_DFF	▤	Person Type Usage Attributes	▤	▤
PER_PERSON_ALLOCATED_TASKS_DFF	▤	Person Allocated Task Attributes	▤	▤

Persons Attributes: Error Message

oracle.jbo.NoDefException: JBO-25002: Definition oracle.apps.hcm.people.core.protectedModel.entity.PersonEO of type Entity Definition is not found. Details : TABLE_NAME :PER_PERSONS ENTITY_OBJECT :oracle.apps.hcm.people.core.protectedModel.entity.PersonEO Exception Stack :JBO-25002: Definition oracle.apps.hcm.people.core.protectedModel.entity.PersonEO of type Entity Definition is not found.

4.8 Integrating Custom Attributes

After you have deployed a flexfield, you can begin incorporating its custom attributes into the Oracle Fusion Middleware technology stack.

For information about the technology stack, see the "Oracle Fusion Middleware Components" chapter in the *Oracle Fusion Applications Concepts Guide*.

Task: Customize Flexfield Pages

After you deploy an extensible or descriptive flexfield, the new attributes appear on all pages that contain the flexfield. You can use Page Composer to configure the custom attribute properties on a page-by-page basis. For example, you can hide some custom attributes. For more information, see Chapter 3, "Customizing Existing Pages."

Note: The custom attributes appear only on pages that include the flexfield. For information about adding flexfields to a page, see the "Adding Descriptive Flexfield UI Components to a Page" section or the "Employing an Extensible Flexfield on a User Interface Page" section in the *Oracle Fusion Applications Developer's Guide*.

Task: Incorporate Custom Attributes into Oracle Business Intelligence

If a descriptive flexfield is enabled for Oracle Business Intelligence, the Manage Descriptive Flexfields task displays a **Manage Segment Labels** button and displays **BI Enabled** checkboxes for the global, context, and context-sensitive segments. Select a segment's **BI Enabled** checkbox to specify that the segment is available for use in Oracle Business Intelligence. You can optionally select a value from the **BI Label** dropdown list to specify the logical object to which the segment should be mapped when the flexfield is imported into Oracle Business Intelligence. This mapping also equalizes context-sensitive segments across contexts, as explained later in this section. For more information, see Task: Plan Attribute Mapping to Oracle Business Intelligence Objects in Section 4.3.1, "Planning Descriptive Flexfields."

When you deploy a flexfield with business intelligence-enabled segments, the deployment process generates a set of **flattened** ADF Business Components in addition to the usual ADF Business Components and ADF Faces runtime artifacts that are generated during deployment. A flattened business component includes a single attribute for the context segment if it is business intelligence-enabled, as well as one attribute for each business intelligence-enabled global segment. For the business intelligence-enabled context-sensitive segments that have labels, the segments are **equalized** across contexts. That is, the flattened business component includes one attribute for each label. For example, a United States context might have a Passport segment and a Canada context might have Visa segment. If you assign the NationalID segment label to both the Passport and Visa segments, they are equalized into the same NationalID attribute in the flattened business component.

For the business intelligence-enabled context-sensitive segments that do not have labels, the business component includes a separate attribute for each non-labeled context-sensitive segment for each context value. Non-labeled context-sensitive segments are not equalized across context values.

After you deploy a business intelligence-enabled flexfield, import the flexfield changes into the Oracle Business Intelligence repository to make use of the newly flattened business components in business intelligence and then propagate the flex object changes. When you import the metadata into the Oracle Business Intelligence repository, you must do so as the `FUSION_APPS_BI_APPID` user.

Note: To import flexfield changes into the Oracle Business Intelligence repository in Oracle Cloud implementations, log a service request using My Oracle Support at `https://support.oracle.com`.

Tip: When you import a flexfield into the Oracle Business Intelligence repository, you see both *name*_ and *name*_c attributes for each segment, along with some other optional attributes. The *name*_ attribute contains the value. The *name*_c attribute contains the code of the value set that the value comes from, and is used for linking to "value dimension." You must import both attributes.

For more information about importing and propagating your flexfield changes, see Task: Configuring Descriptive Flexfields and Key Flexfields for Oracle Business Intelligence in Section 7.3.3, "Customizing the Oracle BI Repository (RPD)." For information about adding flexfields to an Oracle Business Intelligence Publisher data model, see the "Adding Flexfields" chapter in the *Oracle Fusion Middleware Data Modeling Guide for Oracle Business Intelligence Publisher (Oracle Fusion Applications Edition)*.

Task: Incorporate Custom Attributes into Web Services and SOA Infrastructure

When an extensible or descriptive flexfield is deployed, the deployment process regenerates the XML schema definition (XSD), which makes the custom attributes available to web services and the SOA infrastructure.

After deploying a flexfield configuration, you must synchronize the updated XSD files in the MDS repositories for each SOA application. For more information, see "Customizing SOA Composite Applications" section in the *Oracle Fusion Applications Extensibility Guide for Developers*.

Note: To synchronize the updated XSD files in the MDS repositories in Oracle Cloud implementations, log a service request using My Oracle Support at `https://support.oracle.com`.

Task: Enable the Custom Attributes for Oracle Social Network

Use the Manage Oracle Social Network Objects task to enable the flexfield attributes for Oracle Social Network. To access the task, choose **Setup and Maintenance** from the

Administration menu in the global area of Oracle Fusion Applications and search for the task. Next, search for the business object and select the attributes that you want to enable.

5 Customizing Menus

This chapter describes how to use the Manage Menu Customizations task to customize the navigator menu and the home page menu in Oracle Fusion Applications.

This chapter includes the following sections:

5.1 About Menu Configuration

You use the Manage Menu Customizations task to customize the navigator and home page menus. This task is available from the Setup and Maintenance work area, which is accessible from the **Administration** menu in the Oracle Fusion Applications global area.

You customize the menus at the site level and your changes affect all users (or all users of a tenant if in a **multi-tenant environment**).

In non-Cloud implementations, you can localize menu customizations. For more information, see the "Translating Menu Customizations" section in the *Oracle Fusion Applications Extensibility Guide for Developers*.

> **Note:** While you can customize the navigator menu and the home page menu at runtime, you cannot customize the global preferences menu at runtime.
>
> For information about customizing the navigator and home page menus using Oracle JDeveloper, see the "Customizing Menus" section in the *Oracle Fusion Applications Extensibility Guide for Developers*. This functionality is not available in Oracle Cloud implementations.

5.1.1 About Navigator Menu Configuration

The **navigator menu** is the global menu that is accessible from the Oracle Fusion Applications global area. It allows an end user to navigate directly to pages inside of Oracle Fusion Applications as well as to outside web pages. The menu is composed of links (items) that are organized in a hierarchy of groups.

You can customize the navigator menu to address needs that are specific to your organization. For example, you might want to add specialized groupings for cross-functional teams or add links to web pages or external applications. You can add groups and links to the navigator menu, as well as hide and show them.

The navigator menu typically appears when the end user clicks the Navigator link, as shown in Figure 5–1. However, in some non-Cloud applications, the page template can be customized to present the top level groups and items as dropdown buttons, as shown in Figure 5–2, in place of the single Navigator link. For information about how to display the navigator menu as a series of dropdown buttons see the "Rendering the Navigator Menu as Dropdown Buttons" section in the *Oracle Fusion Applications Developer's Guide*.

Figure 5–1 Navigator Menu Example — Navigator Link

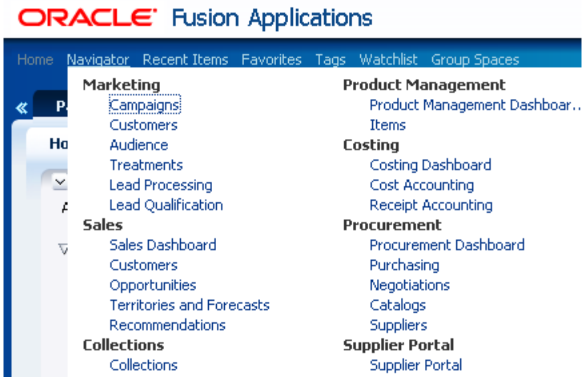
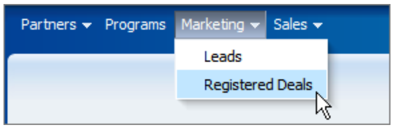
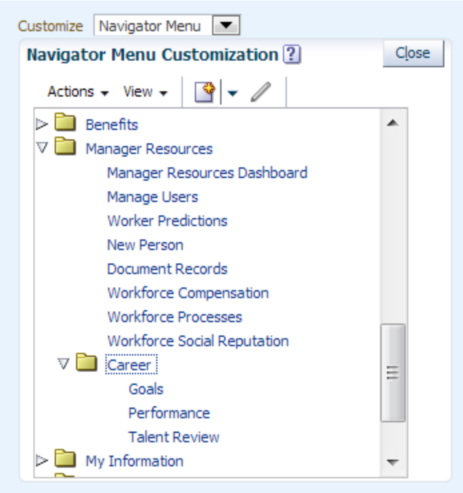


Figure 5–2 Navigator Menu Example — Navigator Dropdown Buttons



The Manage Menu Customizations task displays the menu groups as expandable nodes, as shown in Figure 5–3, with which you can traverse the menu hierarchy.

Figure 5–3 Expandable Group Nodes in Manage Menu Customizations Task



In non-Cloud implementations you can also customize the navigator menu from Oracle JDeveloper. For more information, see the "Customizing Menus" section in the *Oracle Fusion Applications Extensibility Guide for Developers*

5.1.1.1 What You Can Do with the Navigator Menu at Runtime

If you have the required privileges, you can perform the following tasks to customize the menu:

- Add and delete custom groups.
- Edit any group.
- Add and delete custom items.
- Edit any item.
- Specify navigation for an item:
  - Specify navigation to a UI Shell page in an Oracle Fusion application.
  - Specify navigation to an external web page using either a dynamic URL or a static URL.
- Hide or show groups and items.

5.1.1.2 What You Cannot Do with the Navigator Menu at Runtime

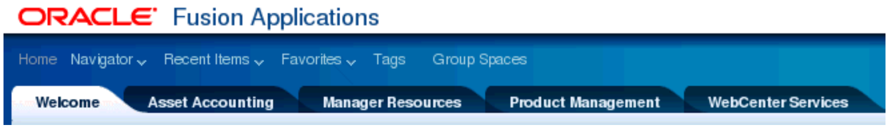
You cannot make the following menu customizations at runtime:

- You cannot add menu items (links) as top-level nodes. You can add nodes to only the groups in the top level and subgroups.
- You cannot delete nodes that are delivered with the product.
- You cannot move nodes. Instead, you must duplicate the node and hide the original node.

5.1.2 About Home Page Menu Configuration

The home page menu is the set of tabs that are displayed in the Oracle Fusion Applications global area at the top of a **desktop page**, as shown in Figure 5–4. The home page menu displays tabs for all the items in the menu for which the end user has access privileges. You can add tabs to the home page menu, as well as hide and show them.

Figure 5–4 Home Page Menu Tabs



In a **multi-tenant environment**, you customize the home page menu at the site level and your changes affect the tenant at that site level. Otherwise, you customize the home page menu at the site level and your customizations affect all end users.

In non-Cloud implementations you can also customize the home page menu from Oracle JDeveloper. For more information, see the "Customizing Menus" section in the *Oracle Fusion Applications Extensibility Guide for Developers*

5.1.2.1 What You Can Do with the Home Page Menu at Runtime

If you have the required privileges, you can perform the following tasks to customize the menu:

- Add and delete custom items.
- Edit any item.
- Specify navigation to a home page in an Oracle Fusion application.
- Hide or show items.

In non-Cloud implementations, you can localize home page menu customizations. For more information, see the "Customizing Menus" section in the *Oracle Fusion Applications Extensibility Guide for Developers*

5.1.2.2 What You Cannot Do with the Home Page Menu at Runtime

You cannot make the following menu customizations at runtime:

- You cannot add menu items (links) as sub-nodes. All nodes are top-level nodes.
- You cannot delete nodes that are delivered with the product.
- You cannot move nodes. Instead, you must duplicate the node and hide the original node.

5.1.3 Before You Begin Customizing a Menu

Before you customize a menu, you should be familiar with the Oracle Fusion application architecture that enables customization, as described in Chapter 1, "Customizing and Extending Oracle Fusion Applications." You should also understand the typical workflows for working with customizations, as described in Chapter 2, "Understanding the Customization Development Lifecycle."

You will need to do the following before you can begin customizing a menu:

- If you are making minor changes, such as adding or editing one or two nodes, then you can hide the changes until you have completed your customizations. However, if you are making more than minor changes, such as rearranging several nodes, you might want to instead create a sandbox and make your changes in the sandbox. For more information see Section 2.2, "Using the Sandbox Manager."
- You must have specific privileges to access the Manage Menu Customizations task. Contact your security administrator for details.
- Open the Manage Menu Customizations task. To access this task, choose **Setup and Maintenance** from the **Administration** menu in the Oracle Fusion Applications global area and search for the task.
- Select the menu type — Navigator Menu or Homepage Menu — from the **Customize** dropdown list.

5.2 Adding Navigator Menu Groups

You arrange the navigator menu by building a hierarchy of nested groups. For example, in Figure 5–3 the Manager Resources group contains the Career group.

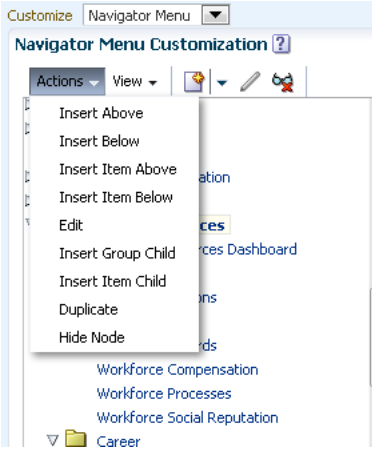
Task: Navigate the Menu Hierarchy

Use the View menu to expand or collapse a group of nodes. You can also right-click a node and access similar actions to facilitate tree navigation.

Task: Add a Group

As shown in Figure 5–5, you can insert a group above or below a peer group or insert a child group. You edit a group by defining a label and specifying whether the group should be rendered. You typically hide the group until all changes have been completed.

Figure 5–5 Actions Menu



5.3 Adding Menu Items

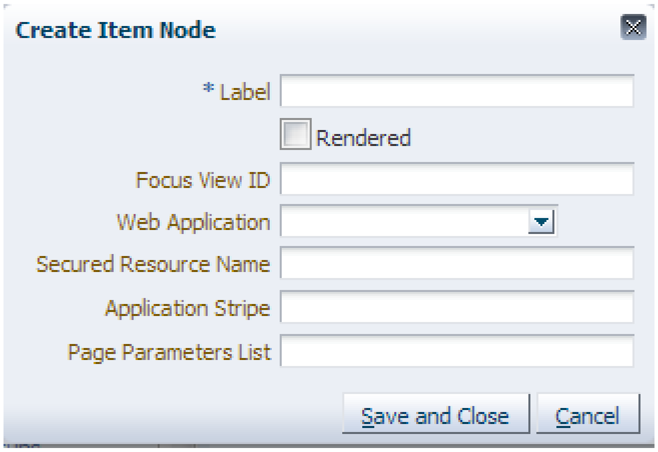
The home page menu items are URL links to Oracle Fusion Applications home pages. Navigator menu items can be links to UI Shell pages or links to external applications and web sites.

You can add navigator menu items to top-level groups and to their subgroups. You cannot add navigator menu items as top-level nodes. Home page menu items are always top-level items.

Task: Adding a Home Page Menu Item

To add a home page menu item, navigate to the place where you want the item to appear and choose **Insert Above** or **Insert Below** from the **Actions** menu. You can also choose to duplicate an existing item. You must supply the menu label and a link to a UI Shell page as described Task: Linking to a UI Shell Page. The Create Item Node dialog for the home page menu is shown in Figure 5–7.

Figure 5–6 Create Item Node Dialog for the Home Page Menu



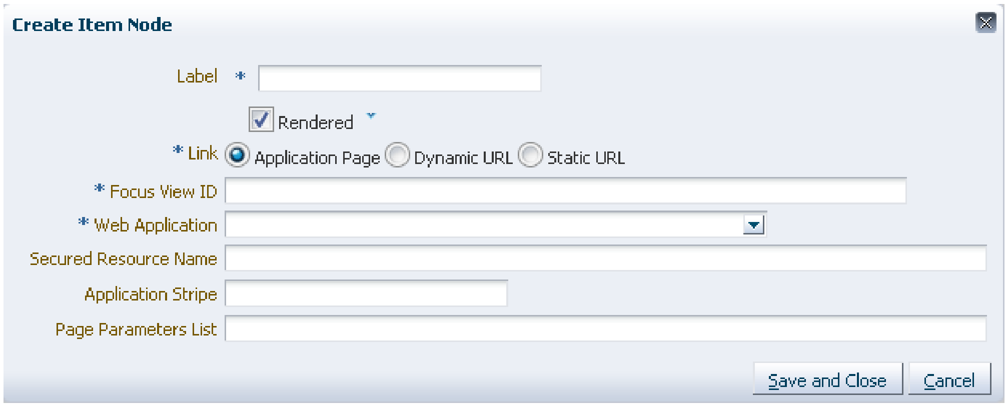
□ Rendered

Task: Adding a Navigator Menu Item

To add a navigator menu item, navigate to the item's group and insert the item above or below another item, as shown in Figure 5–5. You can also choose to duplicate an existing item. When the Create Item Node dialog appears, as shown in Figure 5–7, select the type of link that you want to create.

- Application Page: Use this choice to link to a UI Shell page in an Oracle Fusion application.
- Dynamic URL: Use this choice to link to a page outside of Oracle Fusion Applications where the host, port, or context root might change. To use this choice, you must first register the application in the topology using the Register Enterprise Applications task. The advantage of using a dynamic URL is that when the host, port, or context root changes, you can make the change to the web application in the topology registration and the change affects all menu items that have dynamic links to that web application. An example of a need for a dynamic URL is when you need to link to a test version of an application and you will need to change the host and port when you move the application to production.
- Static URL: Use this choice to link to a page outside of Oracle Fusion Applications where you do not expect the host, port, or context root to change. For example, you can use a static URL to link to http://www.oracle.com.

Figure 5–7 Create Item Node Dialog for the Navigator Menu



☑ Rendered

* Link ● Application Page ○ Dynamic URL ○ Static URL

Task: Linking to a UI Shell Page

To add a menu item that links to a UI Shell page, create a new item to display the Create Item Node dialog shown in Figure 5–7. Select **Application Page**, select the web application from the dropdown list, and provide the view ID of the target page.

The quickest way to obtain the web application name and view ID is to copy them from an existing menu item that links to the same page. In a non-Cloud implementation, you also can obtain the web application name from the context root for the application, and you can obtain the view ID from the `id` attribute for the page's `<view>` tag in the product's `public_html/WEB-INF/adfc-config.xml` file.

If you want to secure access to the target UI Shell page from the menu item, then you must provide the name of the secured resource and the name of the policy store's **application stripe**. When an end user clicks the link, Oracle Fusion Applications checks the secured resource and Lightweight Directory Access Protocol (LDAP) policy store to determine whether the user the **privilege** to view the page.

If there is another menu item that points to the same page, then you can get the secured resource name and application stripe from that item. In a non-Cloud implementation, you also can obtain the application stripe from the `jps.policystore.applicationid` parameter in the application's `weblogic-application.xml` file. Examples of application stripes are `crm`, `fscm`, and `hcm`.

For non-Cloud applications, you can determine the secured resource name by obtaining the name of the web page's page definition file. By default, the page definition files are located in the `view.PageDefs` package in the Application Sources directory of the view project. If the corresponding JavaServer Faces (JSF) page is saved to a directory other than the default (`public_html`), or to a subdirectory of the default, then the page definition will also be saved to a package of the same name. An example of a secured resource name is `oracle.apps.view.pageDefs.CaseList_Form_Attach_UIShellPagePageDef`.

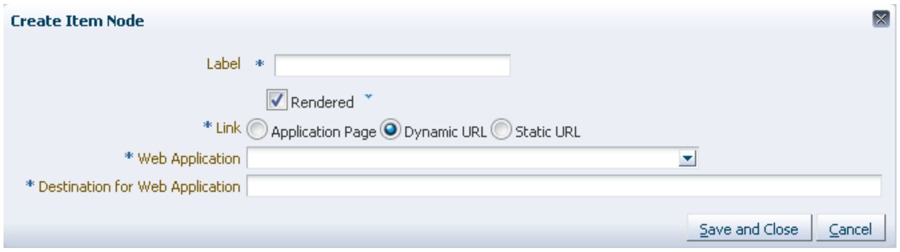
A UI Shell page might take parameters and display or act differently based on the parameters that are passed in. For example, if accessing a page from one group in the menu hierarchy, the parameter might be set to `status="Open"` and if accessing the page from a different group, the parameter might be set to `status="Closed"`. If the page takes parameters, you can use the **Page Parameters List** text box to provide a semicolon-delimited string of name-value pairs, such as `org=m1;context=s1`. You can use expression language (EL) to specify the parameters. If the EL evaluates to an Object, the `toString` value of that Object is passed as the value of the parameter.

Task: Dynamically Linking to an External Web Site or Application

To link to a page outside of Oracle Fusion Applications where the host, port, or context root might change, first use the Register Enterprise Applications task to add the web application to the topology.

Next, create a new menu item, and, in the Create Item Node dialog, select **Dynamic URL** as shown in Figure 5–8. From the Web Application dropdown list, select the web application that you added to the topology. In the **Destination for Web Application** text box, type the remainder of the URL. For example, if you wanted to link to `http://example:9011/myApp/faces/Page1`, you would add MyApp to the topology with a protocol, host, port, and context root of `http://example:9011/myApp`. In the Create Item Node dialog, you would select MyApp from the dropdown list, and you would type `/faces/Page1` in the **Destination for Web Application** text box.

Figure 5–8 Creating an Item Node with a Dynamic URL



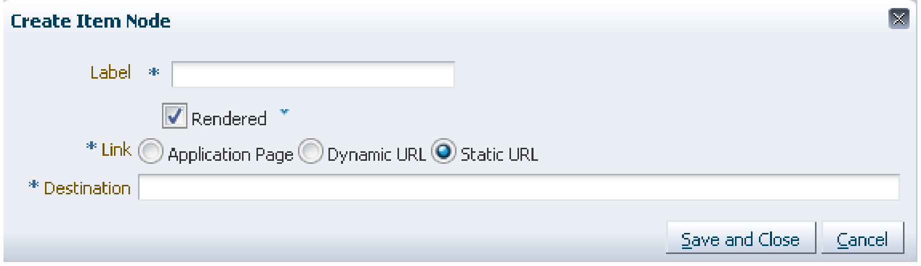
☑ Rendered

* Link ○ Application Page ● Dynamic URL ○ Static URL

Task: Statically Linking to an External Web Site or Application

To link to a page outside of Oracle Fusion Applications where the host, port, or context will not change, create a new menu item, and, in the Create Item Node dialog, select **Static URL** as shown in Figure 5–8. Next, type the full URL, for example `http://www.oracle.com`, in the **Destination** text box.

Figure 5–9 Creating an Item Node with a Static URL



☑ Rendered

* Link ○ Application Page ○ Dynamic URL ● Static URL

5.4 Hiding and Showing Nodes

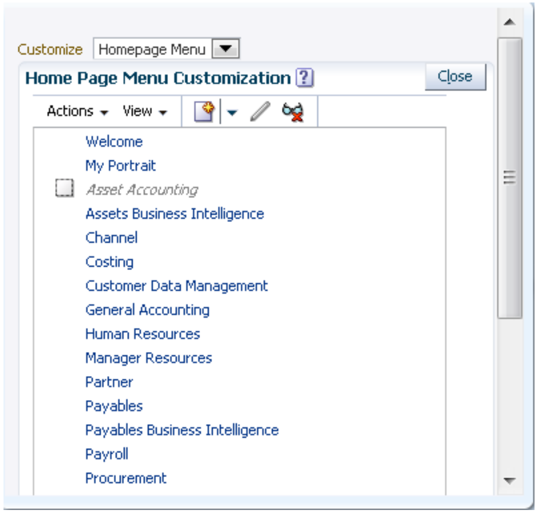
While you are creating or working with a group or item, you might want to prevent end users from accessing the node. You can hide the group or item while you are working with it, and then show the node when you have completed the task.

If you want a group or item to appear only if certain conditions are met, you can use an EL expression to indicate when the node should appear.

> **Tip:** For major changes that need to be tested and approved, you might want to use the sandbox manager instead of hiding and showing nodes. For more information, see Section 2.2, "Using the Sandbox Manager."

The Manage Menu Customizations page shows all nodes. A node that appears in italics either has an EL expression or the **Rendered** check box was deselected, as shown in Figure 5–10.

Figure 5–10 Hidden Node Shown in Italics



Task: Hiding or Showing a Node

When you add a node, the **Rendered** check box is selected by default. To hide the node, deselect the check box. You can edit the node later to change how it is rendered.

If you want the node to appear only if certain conditions exist, click the down arrow on the right side of the **Rendered** check box to open the EL Expression Editor dialog. Type the expression that must be met for the node to appear, such as `#{securityContext.userInRole['ADMIN']}`.

6

Customizing and Extending BPMN Processes

This chapter describes how to use Oracle Business Process Composer (Business Process Composer)to customize and extend Business Process Modeling Notation (BPMN) processes. Several Oracle Fusion applications use BPMN processes to define process flows within the application. This chapter also describes how to edit BPMN processes by creating and modifying BPM projects based on project templates and deploying those projects at runtime.

> **Note:** The Business Process Composer and Oracle Business Process Management Studio functionality described in this chapter is not available in Oracle Cloud implementations of Oracle Fusion Human Capital Management.

This chapter includes the following sections:

- Section 6.1, "About Customizing BPMN Processes"
- Section 6.2, "Creating a BPM Project"
- Section 6.3, "Customizing BPMN Processes"
- Section 6.4, "Saving a BPM Project to the BPM Repository"
- Section 6.5, "Deploying a BPM Project"
- Section 6.6, "Configuring Oracle Fusion Applications to Use BPMN Processes"

6.1 About Customizing BPMN Processes

The Oracle Fusion CRM Application Composer (Application Composer) and Oracle Fusion Human Capital Management product families of Oracle Fusion Applications use Business Process Management Notation (BPMN) processes to define some of the process flows used within their applications.

BPMN is a standard notation for modeling the behavior of business processes. It enables process analysts to create graphical models of a business process. Oracle Business Process Management Suite provides an environment for implementing BPMN processes by enabling process developers to integrate them with other systems and services. Oracle BPM Suite also provides a runtime environment for integrating the running processes within Oracle Fusion Applications. For general information about BPMN, see the "Modeling and Testing Business Processes" part in the *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management*.

Business Process Composer enables you to customize the BPMN processes used within Application Composer and Oracle Fusion Human Capital Management. To customize these processes you must create, modify, and deploy Oracle BPM projects created from a project template. BPMN processes are contained within a BPM project. BPM projects contain all of the resources required for a functioning BPM application, including BPMN processes and Service Oriented Architecture artifacts such a **business rule** or **human task**.

Oracle Fusion applications provide default project templates that you can use to create new process flows. See the product-specific documentation in Oracle Fusion Applications Help for more information about the default BPM project templates provided by Oracle Fusion Applications.

For general information about working with project templates, see the "Working with BPM Projects" section in the *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management*.

A BPM project template is prepopulated with all of the required resources for implementing BPMN processes within an Oracle Fusion application. This includes the BPMN processes that define the process flow as well as the necessary implementation details that enable the processes to communicate with other parts of the application.

After customizing a BPM project, you can deploy it to runtime in one of several ways. See Section 6.5, "Deploying a BPM Project" for more information.

The reusable elements contained within a BPM project are called the business catalog. The business catalog contains different reusable services that a BPMN process can use to connect to other components of the application, including other processes, systems, and databases.

Table 6–1 lists the reusable components of the business catalog.

Table 6–1 Business Catalog Components Available in Business Process Composer

Business Catalog Component	Description
Services	Use services to connect a BPMN process with other processes, systems, and services, including a **BPEL process**, and databases. See the "Working with Services" section in the *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management* for more information.
Human tasks	Human tasks enable you to define how end users interact with your BPMN processes. Human tasks are implemented in a BPMN process using the user task. See the "Working with Human Tasks" chapter in the *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management.*
Business rules	Oracle business rules are statements that describe business policies or describe key business decisions. See the "Using Oracle Business Rules" chapter in the *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management* for information on implementing business rules within a BPMN process.

6.1.1 Oracle Tools for Customizing and Extending BPMN Processes

Oracle Fusion applications provide multiple tools for customizing and extending BPMN processes. These tools are described in Table 6–2.

Table 6–2 Oracle Tools for Customizing and Extending BPMN Processes

This tool...	Enables you to...
Business Process Composer	Customize BPMN processes by creating and deploying BPM projects based on project templates. This functionality is described in the current chapter.
Oracle BPM Studio (Oracle JDeveloper)	Customize project templates. See the "Customizing and Extending Oracle BPM Project Templates" chapter in the *Oracle Fusion Applications Extensibility Guide for Developers*.
Oracle SOA Composer	Customize a business rule, **domain value map**, or approval configuration and assignment rules at runtime. See the "Customizing SOA Composite Applications" section in the *Oracle Fusion Applications Extensibility Guide for Developers* for more information. These customizations are performed directly on a running application. They do not require redeployment of the BPM project containing the BPMN process.
Oracle BPM Worklist	Customize approval configuration and assignment rules. See the "Customizing SOA Composite Applications" section in the *Oracle Fusion Applications Extensibility Guide for Developers* for more information. These customizations are performed directly on a running application. They do not require redeployment of the BPM project containing the BPMN process.

6.1.2 What You Can Do with BPMN Processes at Runtime

There are two types of runtime customization that you can make to BPMN processes. These are described in the following sections.

6.1.2.1 What You Can Customize Using Oracle SOA Composer and Oracle BPM Worklist

BPMN processes use multiple SOA components, including business rules and approval workflow. Using Oracle SOA Composer and Oracle BPM Worklist, you can customize the following components used by the BPMN processes of a running Oracle Fusion application:

- Oracle business rules
- Domain value maps
- Approval assignment rules in human workflows such as customizing the approval flow for a specific customer

These customizations can be made directly to a running Oracle Fusion application without having to redeploy the BPM project. See the "Customizing SOA Composite Applications" section in the *Oracle Fusion Applications Extensibility Guide for Developers* for information about using Oracle SOA Composer or Oracle BPM Worklist to make these customizations.

> **Note:** Any changes you make to the deployed, running BPMN processes of an Oracle Fusion Customer Relationship Management application will not be preserved if you later redeploy the BPM project that contains them.

6.1.2.2 What You Can Customize Using Business Process Composer

Using Business Process Composer, you can make the following customizations to a BPM project created from a project template:

- Customize an existing BPMN process.
- Create new BPMN processes.
- Create simple data objects.
- Create and modify some business catalog components.

  Table 6–3 lists which business catalog components can be customized or created using Business Process Composer.

Table 6–3 List of Business Catalog Components

Business Catalog Component	Can be created using Business Process Composer?	Can be customized using Business Process Composer?
Business rules	Yes	Yes
Human tasks	Yes You can create human tasks using Business Process Composer, however not all functionality of a human task can be customized.	Yes
Services	Yes	Yes

  See Section 6.2, "Creating a BPM Project" for information about creating a new BPM project based on a project template.

  Changes to the business catalog that cannot be made using Business Process Composer must be made to the project template using Oracle BPM Studio. See Section 6.1.3, "What You Cannot Do with BPMN Processes at Runtime" for more information.

6.1.3 What You Cannot Do with BPMN Processes at Runtime

Often, it is necessary to make changes to a BPMN process that cannot be performed using Business Process Composer. The following tasks must be performed using JDeveloper:

- Modify project templates.
- Create or customize some business catalog components, including:
  - Configuring advanced properties of web services
  - Creating new adapters and mediators
  - Creating or customizing error messages
  - Configuring advanced features of human tasks, including:
    - Complex assignment support
    - Automatically generated task flow
    - Business rules within human tasks

To create or customize these components, process developers must modify the project template and republish it to the Oracle BPM repository. After a project template is revised, you can use Business Process Composer to create and deploy BPM projects. See the "Customizing and Extending Oracle BPM Project Templates" chapter in the *Oracle Fusion Applications Extensibility Guide for Developers* for more information.

6.1.4 Before You Begin Customizing BPMN Processes

Before customizing BPMN processes using Business Process Composer, you should be familiar with the Oracle Fusion application architecture that enables customization, as described in Chapter 1, "Customizing and Extending Oracle Fusion Applications."

You should also understand the typical workflows for customization and extensibility, as described in Chapter 2, "Understanding the Customization Development Lifecycle."

You should be familiar with how to model business processes using the Business Process Modeling and Notation (BPMN) standard. See the "Modeling and Testing Business Processes" part in the *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management*.

To create and deploy projects based on project templates, you must have specific privileges. Contact your security administrator for details. See the "Performing Administrative Tasks" chapter in the *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management* for more information.

How you access Business Process Composer depends on which Oracle Fusion application you are using:

- To access Business Process Composer from Oracle Fusion Human Capital Management, access the **Business Process Composer** task by choosing **Setup and Maintenance** from the **Administration** menu in the global area of Oracle Fusion Applications and searching for the task.
- To access Business Process Composer from Oracle Fusion Customer Relationship Management:

  1. Access Application Composer.

     To access Application Composer, in the **Navigator** menu, choose **Application Composer**.

     To edit or create business processes, you will need specific privileges. Contact your security administrator for details.

  2. From the Application Composer navigator, click **Business Processes**.

     A page displays showing a list of BPMN processes.

  3. From the **Actions** menu, select **Create**.

     To create a new business process choose **Create** from the **Actions** menu.

     To modify an existing business process, choose a process from the list, then choose **Update** from the **Actions** menu.

  4. Sign on to Business Process Composer.

     After selecting one of the preceding steps, the Business Process Composer login screen appears. For more information, see the "Signing On to Oracle Business Process Composer" section in the *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management*.

6.2 Creating a BPM Project

You can use Business Process Composer to customize or extend BPMN processes. BPMN processes are contained within a BPM project. After launching Business Process Composer the first step in modifying a BPMN process is to create a new project based on a project template or to open an existing BPM project.

Task: Create a New BPM Project Based on a Project Template

Oracle Fusion applications do not ship with running BPMN processes out-of-the-box. To integrate BPMN processes within Oracle Fusion applications, you must create a new BPM project based on a project template, deploy the project to runtime, and then configure the Oracle Fusion application to use the BPMN processes of the deployed BPM project.

Business Process Composer enables you to create new BPM projects based on project templates. For information about creating a new BPM project, see the "How to Create a New Project from a Project Template" section in the *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management*.

See the product-specific documentation in Oracle Fusion Applications Help for a list of the default BPM project templates provided by Oracle Fusion Applications. By default, project templates are stored in the Templates folder of the BPM repository.

Task: Open an Existing BPM Project

If you have already created a new BPM project based on a template, you can continue to customize the project before deploying it to runtime.

For information about opening an existing BPM project, see the "How to Open a Project Using the Application Welcome Page" section in the *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management*.

6.3 Customizing BPMN Processes

After creating a new BPM project based on a project template, you can modify the BPMN processes within the project. Additionally, you can customize the business catalog components within the BPM project.

Task: Open a BPMN Process

See "How to Open a Business Process" in the *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management* for information about opening a BPMN process.

Task: Customize a BPMN Process

You can alter the flow of your BPMN process by adding, removing, or modifying BPMN flow objects.

See the "Working with Flow Elements" section in the *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management* for more information about using the process editor to modify BPMN processes.

See the "Modeling and Testing Business Processes " part in the *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management* for specific information about BPMN flow objects.

Task: Assign Business Catalog Components to a BPMN Flow Object

You can use Business Process Composer to assign reusable services from the business catalog to different BPMN flow objects.

The business catalog components and their corresponding flow objects are shown in Table 6–4.

Table 6–4 BPMN Flow Objects and Their Corresponding Services

BPMN Flow Object	Business Catalog Component
User task	Human tasks
Service task	Services, including web services and adapters
Business rules task	Business rules

See the "The Business Catalog" section in the *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management* for more information about working with reusable services and the business catalog.

Task: Add Milestones to the Activity Guide

The activity guide of a BPM project defines a set of milestones. Each BPM project contains one activity guide. An activity guide can contain multiple milestones.

A milestone is a specific set of tasks that the process participant has to complete. A milestone is complete when the participant successfully runs a specific set of tasks in the milestone.

Each milestone is a defined by a set of human workflow tasks. Each human workflow task is itself a task flow that may require the collaboration of multiple participants in various roles.

See the "Using Guided Business Processes to Set Project Milestones" section in the *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management.* for information about creating milestones.

Task: Customize Business Rules

Business rules enable dynamic decisions to be made at runtime that allow you to automate policies, computations, and reasoning while separating rule logic from the underlying application code.

> **Note:** You can use Business Process Composer to modify business rules within a BPM project. These changes are made to the runtime application when the project is deployed.
>
> You can use Oracle SOA Composer to make runtime changes directly to the runtime environment, without deploying a project. See the "Customizing SOA Composite Applications" section in the *Oracle Fusion Applications Extensibility Guide for Developers* for more information about making runtime changes to a BPMN process.

For more information about customizing rules using Business Process Composer, see the "Using Oracle Business Rules" chapter in the *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management*.

Task: Create or Customize Human Tasks

Human tasks enable you to integrate human interaction with connectivity to systems and services as part of an end-to-end process flow. Human tasks are responsible for handling all interactions with the users or groups participating in the business process.

Using Business Process Composer, you can create and customize human tasks. See the "Working with Human Tasks" section in the *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management* for information about creating and configuring human tasks.

Task: Customize Expressions

Expressions are used to evaluate the data used within your process. Different flow objects use expressions to determine which path within the process to follow.

Expressions are used to configure the following BPMN elements:

- Conditional Sequence Flows
- Complex Gateways
- Timer Events
- Data Associations
- Loop Markers
- Multi-Instance Markers
- User Task Advanced Properties

You can customize the expressions used within a BPMN process using Business Process Composer.

See the "Working with Expressions" section in the *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management*.

6.4 Saving a BPM Project to the BPM Repository

You can save the changes you make to a BPM project to the BPM repository. In addition to saving your work in progress, saving a project in the repository enables you to share BPM projects with process developers using Business Process Composer and Oracle BPM Studio.

The BPM repository can also be used to share project templates created in Oracle BPM Studio. In the BPM repository, projects and project templates are stored in a default folder as shown in Table 6–5.

Table 6–5 Default Location of BPM Projects and Project Templates in the BPM Repository

Type	Location
BPM Projects	/bpm/drafts
BPM Project Templates	/bpm/templates

Within these default folders, you can create additional subfolders to organize your projects and project templates.

Task: Save a Project

See the "How to Save Changes to a Project" section in the *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management* for information about how to save a project to the BPM repository.

6.5 Deploying a BPM Project

After creating a new BPM project, you must deploy it to runtime to make the BPMN processes contained within it accessible to Oracle Fusion applications.

After customizing a project and publishing it, you must deploy the project to runtime. After the project is deployed, the BPMN processes within it are accessible to the Oracle Fusion applications that implement them.

Task: Deploy a BPM Project

There are three methods of deploying a BPM project to Oracle Fusion applications using Business Process Composer. The specific method you use depends on whether you are in a production or test environment:

- Using a **SAR file**

  This is the recommended deployment method if you are deploying a BPM project to a production environment. A SAR file is an archive of a **SOA composite application** that, like a BPM project, contains all of the required resources of a deployable application.

  You can use Business Process Composer to export a BPM project as a SAR file. See the "Deploying a BPM Project" chapter in the *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management.*

  After the project is exported as a SAR file, your system administrator must deploy it to runtime using Oracle Enterprise Manager.

- Using a deployment plan

  You can use Business Process Composer to generate a deployment plan for your project. Like a SAR file, system administrators can use a deployment plan to deploy a BPM project to runtime.

  See the "How to Generate a Deployment Plan" section in the *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management* for information about generating a deployment plan using Business Process Composer.

- Directly from Business Process Composer

  You can deploy a BPM project to runtime directly from Business Process Composer. See the "How to Deploy a Project to Run Time" section in the *Oracle Fusion Middleware Business Process Composer User's Guide for Oracle Business Process Management* for information about deploying a BPM project to runtime.

  To deploy project using Business Process Composer you must have specific privileges. Contact your system administrator for details.

  You may want to deploy a BPM project using the same name as the project and project template to make it easier to associate the deployed project with its design time version

Task: Configure Oracle Fusion Applications to Use BPMN Processes

After deploying a BPM project, you must configure the Oracle Fusion application to use the BPMN processes within the application. See Section 6.6, "Configuring Oracle Fusion Applications to Use BPMN Processes" for more information.

6.6 Configuring Oracle Fusion Applications to Use BPMN Processes

After deploying a BPM project to runtime, you must configure your Oracle Fusion applications to use the BPMN processes.

6.6.1 Configuring BPMN Processes in CRM Applications

After deploying a BPM project, you must use Application Composer to add an object workflow that conditionally responds to a record modification event for the **business object**.

6.6.2 Configuring BPMN Processes in HCM Applications

After deploying your BPM project to Oracle BPM runtime, you must register the BPMN processes using the Register Workforce Process page in Oracle Fusion Workforce Lifecycle Manager. When you register a process, you must provide the name of the process as it appears in the Start menu of the Manage Workforce Process page. Depending on how your process is designed, you may also need to specify the parameters that are passed to the workforce process on startup.

For more information, see product-specific documentation in Oracle Fusion Applications Help.

7 Customizing Reports and Analytics

This chapter describes how to use Oracle Business Intelligence Suite (Oracle BI Suite) to customize and extend reports and analytics for Oracle Fusion Applications, including customizing Oracle BI Publisher layouts and data models, customizing Oracle BI Enterprise Edition analyses and dashboards, and extending the Oracle BI repository.

This chapter includes the following sections:

- Section 7.1, "About Customizing Reports and Analytics"
- Section 7.2, "Customizing Reports"
- Section 7.3, "Customizing Analytics"

7.1 About Customizing Reports and Analytics

In Oracle Fusion Applications, reports and analytics are built using Oracle Business Intelligence:

- Reports are built with Oracle Business Intelligence Publisher (Oracle BI Publisher) and are usually highly-formatted, detailed documents. Oracle BI Publisher is also better suited for reports with substantial volume (more than a few hundred rows of data).
- Analytics are analyses and dashboards built with Oracle Business Intelligence Presentation Services. Analyses are queries based on real-time, transactional or operational data that provide answers to business questions. Dashboards provide personalized views of corporate and external information. A dashboard consists of one or more pages that contain content, such as analyses, links to websites, BI Publisher reports, and so on.

7.2 Customizing Reports

This section describes how to use Oracle BI Publisher to customize and extend reports for Oracle Fusion Applications. It includes the following sections:

- Section 7.2.1, "About Customizing Reports"
- Section 7.2.2, "Customizing Layouts"
- Section 7.2.3, "Customizing Data Models"
- Section 7.2.4, "Creating Custom Reports"
- Section 7.2.5, "Adding Translations"

- Section 7.2.6, "Tasks Required to Run Custom Reports with Oracle Enterprise Scheduler"
- Section 7.2.7, "Securing Custom Reports and Related Components"
- Section 7.2.8, "Making Reports Available to Users in the Reports and Analytics Pane"
- Section 7.2.9, "Enabling Reports for Scheduling from the Reports and Analytics Pane"

7.2.1 About Customizing Reports

Reports extract the data from your applications and present it in the formats required for your enterprise. Reports provide the information you need for internal operations and statutory compliance; reports also provide the business documents for communicating with your customers. Many product-specific reports are provided with Oracle Fusion Applications; for example, the invoice register, the pick slip report, the payroll summary, the journals report, and the customer credit memo. To meet the specific needs of your enterprise, you may need to customize the reports provided or you may need to create new reports to capture and present different data.

Understanding the BI Publisher reporting architecture will help you to understand the report customization scenarios and tasks. A report in Oracle BI Publisher consists of the following **components**:

- A data model that defines the data source, data structure, and parameters for the report. A data model can be used by multiple reports. Each report has one data model.
- One or more layouts to define the presentation, formatting, and visualizations of the data. A report may have multiple layouts of the data model.
- A set of properties that specify runtime and formatting options.

Optionally, a report may also include:

- Translations to provide localized versions of a report.

7.2.1.1 What You Can Customize

In many cases, Oracle BI Publisher reports delivered with Oracle Fusion applications contain the appropriate data elements you expect, but may not provide the presentation of the data just as you would like it. Oracle BI Publisher enables you to customize the layouts for reports leveraging the prebuilt data models. If the reports provided by Oracle Fusion applications do not include the data you require, you can create a new report based on a custom data model.

Important: Oracle recommends that you do not directly edit the predefined report objects. If you change a report and a subsequent patch includes a new version of the report object, the patch overwrites any customizations. When customizing reports, use the Customize feature to seamlessly create the copy for you. See Section 7.2.1.4, "Using the Customize Feature."

If you choose not to use the Customize feature or you are creating new reports, place your custom report objects in the Shared/Custom folder in the catalog.

Some common report customization scenarios are shown in the following table.

Customization Use Case	Described in
Edit the layout of a report provided with an application For example: Add your company logo to the Receivables Credit Memo report	"Customizing Layouts" on page 7-9
Add a new layout to a report provided with an application For example: Design a new form letter users can select when they run the Receivables Credit Memo	"Customizing Layouts" on page 7-9
Create a new report based on a data model provided with an application	"Creating Custom Reports" on page 7-29
Edit a data model provided with an application For example: Add a field to a data model	"Editing Existing Data Models" on page 7-26
Create a new data model For example: Define a new query against Oracle Fusion applications tables	"Creating a New Data Model" on page 7-27
Create a new report based on a custom data model	"Creating Custom Reports" on page 7-29

7.2.1.2 Related Report Customization Tasks

Depending on how a report is implemented in Oracle Fusion Applications and the type of customization you make you may also have to perform additional tasks to implement your custom report in the system.

When you create a new report and you wish to schedule this report through Oracle Enterprise Scheduler, you must create an Oracle Enterprise Scheduler job for the report. If you require Oracle Enterprise Scheduler to send parameter values to the Oracle BI Publisher report you must also create a parameter view object to pass the values.

If you create a custom layout and you require translations of the layout, you must also provide the translations. Oracle BI Publisher provides a tool for extracting the translation file for some layout types. The translation file can be translated into the required languages then uploaded to the report.

You must also ensure that the proper security settings are applied to the report and data model to enable the intended report consumers to run the report.

Table 7–1 provides links to related report customization tasks.

Table 7–1 Related Report Customization Tasks

Related Report Customization Task	Described in
Create the Parameter View Object for Oracle Enterprise Scheduler	"Customizing Parameters for Reports Submitted Through Oracle Enterprise Scheduler" on page 7-31
Create the Oracle Enterprise Scheduler job definition and job	"Tasks Required to Run Custom Reports with Oracle Enterprise Scheduler" on page 7-30
Provide translations	"Adding Translations" on page 7-29

Table 7–1 (Cont.) Related Report Customization Tasks

Related Report Customization Task	Described in
Secure reports and related components	"Securing Custom Reports and Related Components" on page 7-31

7.2.1.3 Tools for Customizing Reports

Customize reports either within the Oracle BI Publisher application or using one of the tools or applications listed in Table 7–2. For the list of certified versions of third-party applications, see the "System Requirements and Certification" section in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.

Table 7–2 Tools for Customizing Report Components

Component	Tool for Customizing
Report Data Model	BI Publisher's data model editor
Report properties	BI Publisher's report editor
Layout	See Table 7–3, " Tools for Customizing Layouts"

Table 7–3 shows the tools required to customize each layout type.

Table 7–3 Tools for Customizing Layouts

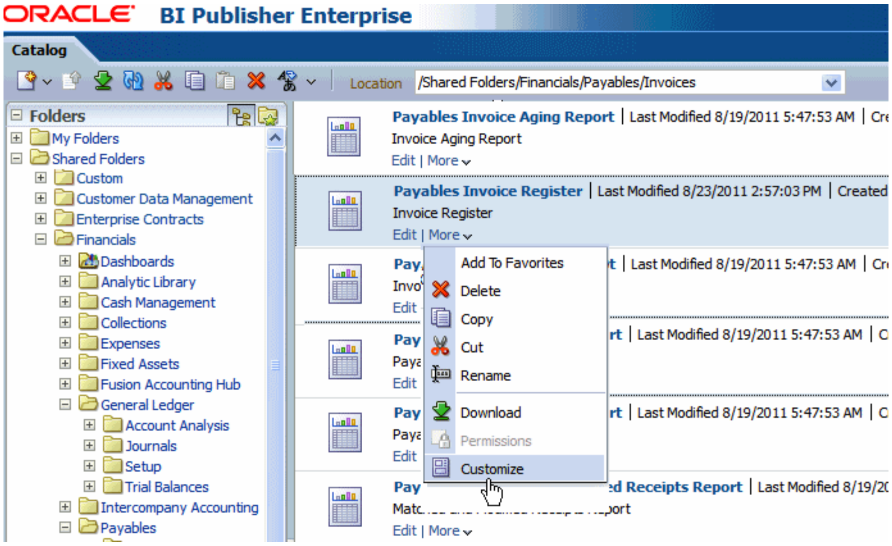
Layout Template Type	Tool for Customizing
RTF template	Microsoft Word with BI Publisher's Template Builder for Word (see Section 7.2.2.1, "Customizing RTF Templates")
BI Publisher template (XPT)	BI Publisher's layout editor (see Section 7.2.2.2, "Customizing BI Publisher Templates")
PDF template	Adobe Acrobat Professional
Excel template	Microsoft Excel with BI Publisher's Template Builder for Excel
eText Template	Microsoft Word

7.2.1.4 Using the Customize Feature

To facilitate the customization of Oracle Fusion Applications reports, BI Publisher provides the **Customize** feature. When you select the **Customize** option for a report, BI Publisher creates a copy of the report in the "Custom" folder. This custom copy is linked internally to the original report. You can customize the custom copy of the report, leaving the original report intact. When users initiate a request to run the original report, whether from Oracle Enterprise Scheduler, from the BI catalog or through an application process, BI Publisher detects the customized version and runs your custom version instead.

Figure 7–1 shows the **Customize** task available from the report **More** menu. In this example, the Payables Invoice Register is automatically copied to Shared Folders/**Custom**/Financials/Payables/Invoices.

Figure 7–1 Selecting the Customize Feature for a Report



The Customize feature provides the following benefits:

- Enables a customization process that mimics the convenience of an "in-place" customization. BI Publisher automatically creates the copy and the mapping to the original report.
- Removes the requirement to edit calling processes or applications to execute the custom report. BI Publisher automatically sends all requests to run the original report to the custom copy. Therefore Oracle Enterprise Scheduler jobs that point to the original report or applications that call the original report will automatically run your custom report with no additional configuration required.
- Automatically copies the security settings of the original report.
- Removes the risk of patches overwriting your customizations of Oracle-provided reports.

User Requirements

Following are requirements for a user to customize a report:

- A user must be granted the BI Author role (or a role that includes the permission oracle.bi.publisher.developReport). For more information, see "Permissions Required to Edit Reports, Style Templates, and Subtemplates."
- A user must be granted these permissions on the original report in the catalog:
  - Read
  - Run Publisher Report
- A user must be granted these permissions on the Custom folder in the catalog:
  - Read
  - Write

Following are requirements for a user to view a report in the Custom folder:

- A user must be granted the BI Consumer role
- A user must be granted these permissions on the original report in the catalog:

  - Read
  - Run Publisher Report
- A user must be granted these permissions on the Custom report in the catalog:
  - Read
  - Run Publisher Report
- The user must be granted access to the data source and be granted these permissions on the data model of the original report:
  - Read
  - Traverse

Limitations

Following are limitations of the Customize report option:

- The Customize option is only available through direct access to the BI Publisher server using the `/xmlpserver` URL (for example: `http://hostname.com:7001/xmlpserver`). The Customize option is not available through the `/analytics` URL used to access Oracle Business Intelligence Enterprise Edition.
- The Customize option is available only for reports. The Customize option is not available for data models, style templates, or sub templates.

  To customize data models, style templates or sub templates and insulate them from potential changes from patching: Make a copy of the data model, style template, or sub-template and either rename it or place it in a custom folder. Ensure that you update any reports to point to the customized data model, style template, or sub template.

7.2.1.5 Considerations When Using the Customize Feature

When you click the Customize link, the report, folder structure, and original report permissions are copied to the Custom folder. The custom report still references the original data model. The data model is not copied. Because the data model referenced by the custom report is the original data model, an Oracle patch that updates the data model may impact the running of your custom report if the data structure is changed.

When an Oracle patch is applied that updates the original report, the custom report is not updated in any way.

When a custom version of a report exists, performing tasks on the original report have the following results:

Task Performed on the Original Report	Behavior When a Custom Report Is Present
Open	Opens the custom report.
Schedule	Creates a report job for the custom report.
Edit	Edits the custom report.
Delete	Deletes the original report (only).
Copy	Copies the original report.
Cut/Paste	Cuts and pastes the original report.

Task Performed on the Original Report	Behavior When a Custom Report Is Present
Rename	Renames the original report. The custom report name is not changed.
Download	Downloads the custom report.
Customize	Edits the custom report.

The Job History for the custom report is maintained under the original report.

To apply further edits to a report in the Custom folder perform one of the following

- Select the Customize or Edit option for the original report; this action opens the existing customized report.
- Navigate to the custom report in the Custom folder and select Edit.

If you manually create the identical folder path to a report under the Custom folder and create a report with the same name, BI Publisher treats the identically named report as a customized report and will run this report instead of the original report, as if you had used the Customize option to create it.

If you delete the original report, the custom report is not deleted. If you delete the custom report, the original report is not deleted.

7.2.1.6 Before You Begin Customizing Reports

Before you customize reports, you should be familiar with the Oracle Fusion application architecture that enables customization, as described in Chapter 1, "Customizing and Extending Oracle Fusion Applications." You should also understand the typical workflows for working with runtime customizations, as described in Chapter 2, "Understanding the Customization Development Lifecycle."

In addition, be familiar with the following Oracle BI Publisher-specific requirements:

Ensure that you have proper permissions for editing and creating Oracle Business Intelligence Publisher objects

To view and perform actions on report objects in the BI Presentation catalog your role must be granted the appropriate combination of functional permissions to enable the actions and access permissions to the object in the catalog. The functional permissions are granted in the definition of the role; the access permissions are granted in the catalog.

Permissions Required to Edit Reports, Style Templates, and Subtemplates

To edit, create, or copy reports, style templates, and subtemplates you must be granted the BI Author role. You must be granted additional permissions on the objects in the catalog to enable the actions on the specific objects.

The following security settings are configured by default in the BI Presentation catalog:

- All roles are granted the Traverse permission on all reports.
- The BI Author role is granted Full Control on the Custom folder and product family subfolders to enable editing and deleting of the objects copied there to customize.

  The preceding settings are configured to enable users with the BI Author role and Read permission to copy seeded reports, subtemplates, and style templates to the

Custom folder and to edit these objects. These settings also enable BI Author to create new reports, subtemplates, and style templates under the Custom folder.

Permissions Required to Customize Data Models

Creating and editing data models requires a role that includes the BI Author role and the BI Publisher Data Model Developer role. The BI Publisher Data Model Developer role is included in the Application Developer role. Note that the ability to create Oracle BI Publisher data models enables the user to write and execute SQL, therefore implementors must consider carefully to whom they grant the BI Publisher Data Model Developer role.

> **Important:** Because this role enables the ability to write SQL, it carries substantial access privileges and should, therefore, only be assigned when needed. As a guideline: Developers should be granted this access in development environments; only trusted individuals should be granted this access in test environments. Implementers should consider not extending this privilege to any user in a production environment.

To grant the role for editing data models, follow the guidelines in the "Configuring Roles" section in the *Oracle Fusion Applications Administrator's Guide* in conjunction with the guidelines in the *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*. For Oracle Cloud implementations, request this role by logging a service request using My Oracle Support at `https://support.oracle.com`.

For more information about setting permissions in the catalog, see the "Managing Objects in the Oracle BI Presentation Catalog" chapter in the *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition)*.

Understand how the patching process for catalog objects impacts customizations

If a patch includes an update to a catalog object that was delivered with an Oracle Fusion application (for example, the Payables Invoice Register report) the patch will overwrite any customizations applied to the original report. To avoid overwriting a customization, do not customize a predefined Oracle Fusion application object in place, instead use the Customize option see Section 7.2.1.4, "Using the Customize Feature."

Be aware of any application-specific guidelines for customizing reports

See application-specific documentation in the Oracle Enterprise Repository for Oracle Fusion Applications.

Be aware of property settings that determine how the report can be run and viewed

Some reports are configured to run only through an external application or through Oracle Enterprise Scheduler. While customizing a report, you may want to configure it temporarily for viewing online to facilitate testing. See Task: Review Report Settings for Online Viewing for information about these settings.

Know how to access Oracle BI Publisher report components

Your entry point to customize report components will depend on what report component you are customizing.

Navigate to the Oracle BI Presentation Catalog as follows:

On the **Navigator**, under **Tools**, click **Reports and Analytics**. In the **Reports and Analytics** pane, click **Browse Catalog**.

Alternatively, log in to Oracle Business Intelligence directly (example: http://host:port/analytics/saw.dll).

Oracle Fusion Applications reporting objects are organized by product family in the catalog typically as follows:

- Top Level: Shared Folders
  - Product Family Folder (example: Financials)
    - Product folder (example: Payables)

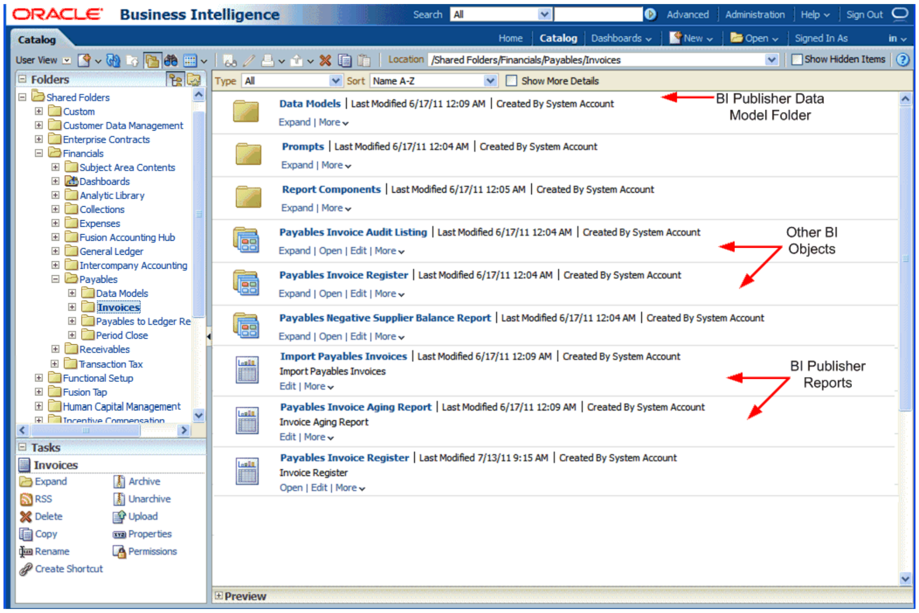
      --Report group folders (example: Invoices)

        -- Reports

        -- Data Model folder

Figure 7–2 shows the Oracle BI Presentation Catalog.

Figure 7–2 Oracle BI Presentation Catalog



Location: /Shared Folders/Financials/Payables/Invoices

Type: All | Sort: Name A-Z | □ Show More Details

Data Models | Last Modified 6/17/11 12:09 AM | Created By System Account
Expand | More

Prompts | Last Modified 6/17/11 12:04 AM | Created By System Account
Expand | More

Report Components | Last Modified 6/17/11 12:05 AM | Created By System Account
Expand | More

Payables Invoice Audit Listing | Last Modified 6/17/11 12:04 AM | Created By System Account
Expand | Open | Edit | More

Payables Invoice Register | Last Modified 6/17/11 12:04 AM | Created By System Account
Expand | Open | Edit | More

Payables Negative Supplier Balance Report | Last Modified 6/17/11 12:04 AM | Created By System Account
Expand | Open | Edit | More

Import Payables Invoices | Last Modified 6/17/11 12:09 AM | Created By System Account
Import Payables Invoices
Edit | More

Payables Invoice Aging Report | Last Modified 6/17/11 12:09 AM | Created By System Account
Invoice Aging Report
Edit | More

Payables Invoice Register | Last Modified 7/13/11 9:15 AM | Created By System Account
Invoice Register
Open | Edit | More

BI Publisher Data Model Folder

Other BI Objects

BI Publisher Reports

If you are customizing a data model, sub template or style template, initiate the customization from this entry point.

The Customize option for BI Publisher layouts is only available through direct access to the BI Publisher server using the `/xmlpserver` URL (for example: `http://hostname.com:7001/xmlpserver`).

7.2.2 Customizing Layouts

The layout defines the presentation of the report data. All reports delivered with Oracle Fusion Applications include at least one predefined layout template file that defines the presentation components (such as tables and labeled fields) and maps the elements from the data model to these components. The layout also defines font sizes, styles, borders, shading, and can also include images, such as a company logo.

To customize a layout, you edit the layout template. Oracle BI Publisher supports several types of templates to support different report layout requirements. Most of the templates delivered with Oracle Fusion Applications are rich text format (RTF) templates created using Microsoft Word. Some predefined templates are BI Publisher layout templates created using Oracle BI Publisher's layout editor. These are for interactive and more visually appealing layouts. A third type is the eText template, which is used specifically for electronic data interchange (EDI) and electronic funds transfer (EFT).

Oracle BI Publisher templates can also be created using Adobe PDF, Microsoft Excel, Adobe Flash, and XSL-FO.

Some examples of layout customizations are:

- Style changes only, no changes to data mapping

  This is the simplest type of customization. Examples are removing the predefined logo from the report and inserting your own or simply modifying colors and font styles. For these changes you can download the predefined template and edit it. Because there are no changes to the data mapping, style changes do not require sample data from your report; however, having sample data available will enable testing of the template from your desktop.

- Changes to mapped data elements within the existing layout

  An example of this type of customization is adding or removing a table column or data field from the report layout. For these changes you must have sample data to load to the layout editing tool. You can download the predefined template, load your sample data, insert the required elements, preview your template, then upload your customized template back to the report definition.

- New presentation of the data

  To create a new layout, start by opening the layout editing tool and loading the sample data to begin designing your custom layout.

Customizing a layout consists of the following tasks:

1. Task: Create a Custom Copy of the Original Report
2. Task: Review Report Settings for Online Viewing
3. Task: Generate Sample Data from the Report
4. Task: Edit or Create the Layout
5. Task: Upload the Template File to the Report Definition
6. Task: Configure the Layout Settings

Task: Create a Custom Copy of the Original Report

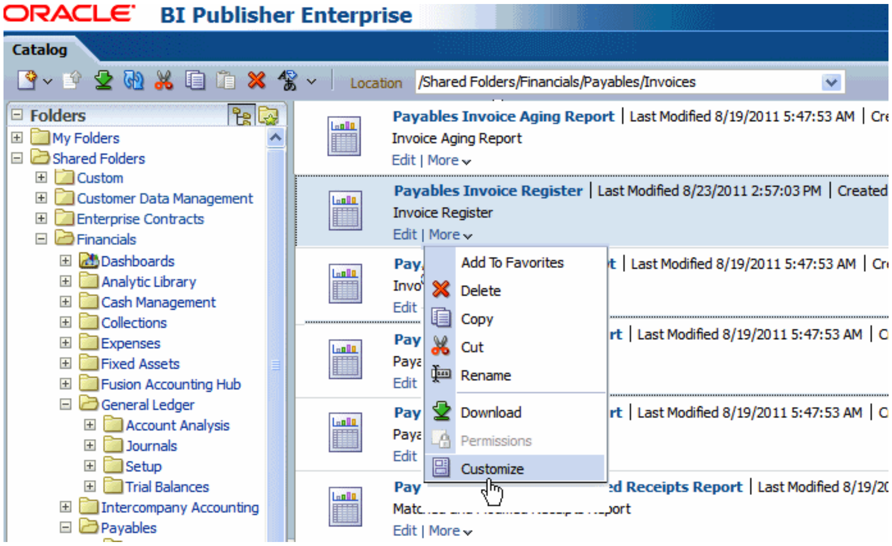
To create the custom copy using the Customize feature:

1. Log in to BI Publisher Enterprise through the /xmlpserver URL.

   One method of navigation to the BI Publisher server is: On the **Navigator**, under **Tools**, click **Reports and Analytics**. In the **Reports and Analytics** pane, click **Browse Catalog**. This opens Oracle Business Intelligence (http://host:port/analytics/saw.dll). Change the final node of the URL to `xmlpserver`: `http://host:port/xmlpserver.`

2. Navigate to your report in the catalog. Click **More** and then click **Customize**, as shown in Figure 7–3.

Figure 7–3 The Customize Task Option



Location: /Shared Folders/Financials/Payables/Invoices

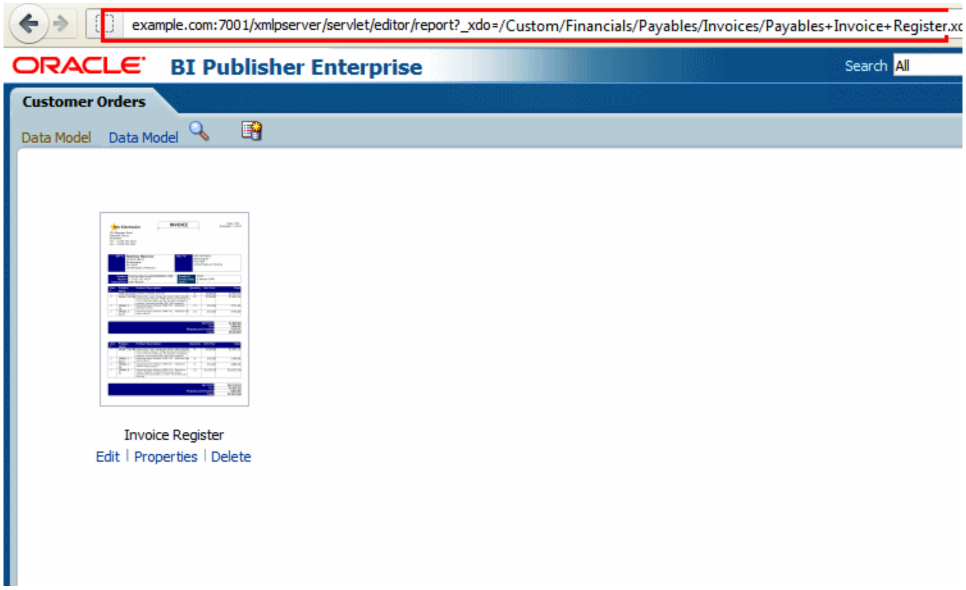
Payables Invoice Aging Report | Last Modified 8/19/2011 5:47:53 AM | Cre
Invoice Aging Report

Payables Invoice Register | Last Modified 8/23/2011 2:57:03 PM | Created
Invoice Register

BI Publisher creates a copy of the report in the Custom folder, under the identical folder hierarchy and opens the report copy in the report editor. In this example the report is created under: Shared Folders/**Custom**/Financials/Payables/Invoices.

Figure 7–4 shows the custom report in the report editor. Notice that the report URL now references the report in the Custom folder.

Figure 7–4 Custom Copy Opened in the Report Editor



example.com:7001/xmlpserver/servlet/editor/report?_xdo=/Custom/Financials/Payables/Invoices/Payables+Invoice+Register.xd

Task: Review Report Settings for Online Viewing

Some reports are configured to view only through an external application or through Oracle Enterprise Scheduler. To view your report online while you are customizing it, ensure that the following properties are set as shown in this section. When finished customizing your report, ensure that you reset these properties appropriately for production.

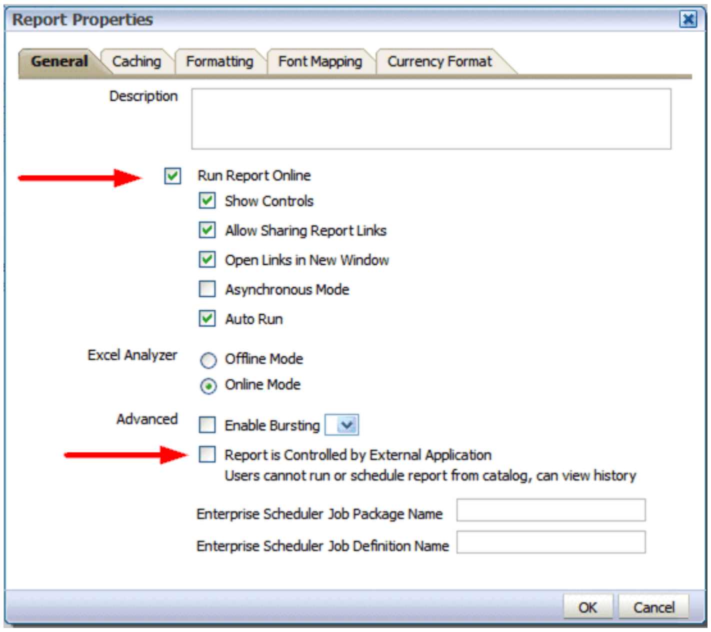
- Report Properties Settings

- **Run Report Online** - must be enabled
- **Report is Controlled by External Application** - must be disabled

To navigate to the report **Properties** dialog:

1. Navigate to your report copy in the catalog and click **Edit**.
2. In the report editor, click the **Properties** link at the top of the page.
3. In the **Properties** dialog, select **Run Report Online** and clear **Report is Controlled by External Application**. These properties are shown in Figure 7–5.

Figure 7–5 Report Properties Dialog



Description

☑ Run Report Online

☑ Show Controls

☑ Allow Sharing Report Links

☑ Open Links in New Window

□ Asynchronous Mode

☑ Auto Run

Excel Analyzer ○ Offline Mode

● Online Mode

Advanced □ Enable Bursting

□ Report is Controlled by External Application
Users cannot run or schedule report from catalog, can view history

Enterprise Scheduler Job Package Name

Enterprise Scheduler Job Definition Name

- Layout Setting

  The layout setting **View Online** must be enabled.

  To view the layout settings:

  In the report editor, click **View a List**. Ensure that the **View Online** property is enabled.

Task: Generate Sample Data from the Report

The layout tools require sample data to enable the mapping of data fields to layout components in the report. You can generate sample data in the following ways:

- From the report data model
- From the report viewer
- From the scheduler

To generate sample data from the data model:

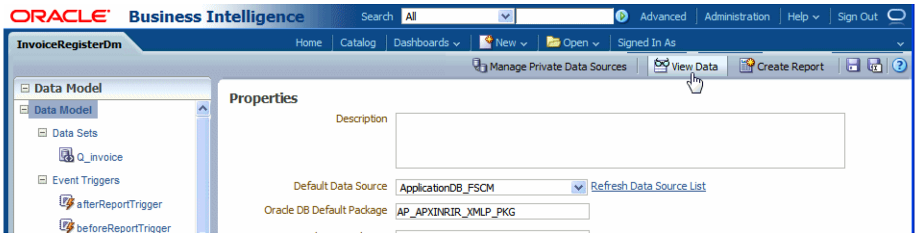
This procedure requires the BI Publisher Data Model Developer to open the data model.

1. Navigate to the report data model in the Oracle BI Presentation Catalog and click **Edit**.

> **Tip:** If you are not sure which data model is the source for a particular report, click the report **Edit** link. The data model is displayed in the upper left corner of the report editor.

2. In the data model editor, click **View Data** as shown in Figure 7–6.

Figure 7–6 Getting XML Output from a Data Model



Default Data Source: ApplicationDB_FSCM

Oracle DB Default Package: AP_APXINRIR_XMLP_PKG

3. Enter values for any required parameters, select the **Number of rows to return**, and click **View**.

4. To save the sample data to the data model click **Save As Sample Data**.

   If you are designing an RTF template, click **Export** to save the file to a local client for use with the Template Builder for Microsoft Word.

5. Click **Save**. To save the data model.

To save sample data from the report viewer:

This procedure requires that the report is enabled for online viewing. For information on enabling a report for online viewing, see Task: Review Report Settings for Online Viewing.

1. Navigate to the report in the Business Intelligence catalog.
2. Click **Open** to run the report in the report viewer with the default parameters.
3. On the **Actions** menu, click **Export**, then click **Data**.
4. Save the data file.

To save sample data from the scheduler:

This procedure requires that the report is enabled for scheduling through BI Publisher. For information, see Task: Review Report Settings for Online Viewing.

1. Navigate to the report in the Business Intelligence catalog.
2. Click **Schedule** to open the BI Publisher scheduler.
3. On the **General** tab, enter values for any report parameters.
4. On the **Output** tab, ensure that **Save Data for Republishing** is selected.
5. Click **Submit**.
6. Navigate to the **Report Job History** page: On the global header click **Open**, then click **Report Job History**.

7. Locate your report job in the **Report Job Histories** table and click the job name. On the details page, under **Output and Delivery** click the **XML Data** download icon.

Task: Edit or Create the Layout

To design an RTF layout, see Section 7.2.2.1, "Customizing RTF Templates."

To design a layout using the Oracle BI Publisher layout editor, see Section 7.2.2.2, "Customizing BI Publisher Templates."

To design one of the other supported layout types, see the corresponding chapter in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*:

- Creating PDF Templates
- Creating Excel Templates
- Creating eText Templates

Task: Upload the Template File to the Report Definition

If you created a layout using the layout editor, the layout is automatically saved to the report definition and you can skip this step. For all other layout types upload the template file to the report definition.

To upload the template file to the report definition:

1. Navigate to your custom report under the Custom folder in the catalog and click **Edit**.
2. On the report definition page, click **View a List**. On the table that lists the layouts, click **Create** (the "+" button).
3. Under **Upload or Generate Layout,** click **Upload**.
4. In the Upload Template dialog perform the following:
   - Enter a **Layout Name**.
   - Browse for and select the **Template File**.
   - Select the template file **Type**.
   - Select the **Locale**. The locale is not editable once the template file is saved to the report definition.
   - Click **Upload**.
5. Save the report definition.

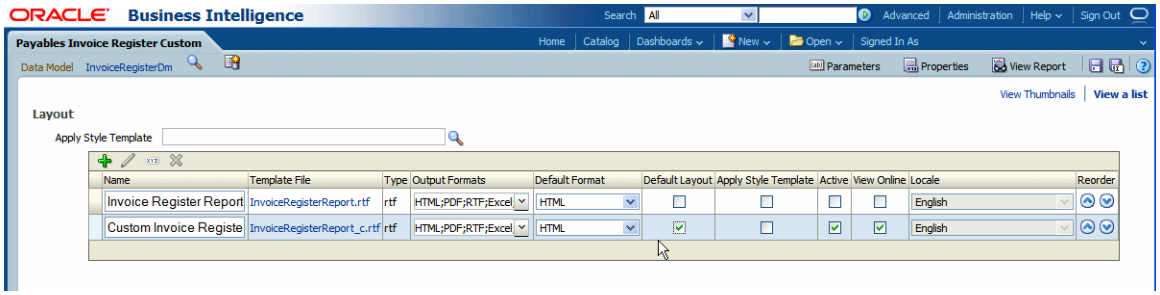
Task: Configure the Layout Settings

> **Tip:** To hide the original layout from users, clear the **Active** box.

To edit the layout settings:

1. In the report editor, click **View a List**. The List View is shown in Figure 7–7.

Figure 7–7 Report Layouts Shown in the List View



Layout

Apply Style Template

Name	Template File	Type	Output Formats	Default Format	Default Layout	Apply Style Template	Active	View Online	Locale	Reorder
Invoice Register Report	InvoiceRegisterReport.rtf	rtf	HTML;PDF;RTF;Excel	HTML	□	□	□	□	English	
Custom Invoice Registe	InvoiceRegisterReport_c.rtf	rtf	HTML;PDF;RTF;Excel	HTML	☑	□	☑	☑	English	

Set the following properties for your custom layout:

- Output Formats

  Output formats are the file formats available for the generated report, such as PDF, HTML, RTF, Excel. Depending on the requirements of a report you may want to limit the output formats available to users. The output formats available vary according to the template file type.

- Default Output Format

  When multiple output formats are available for the report, the default output format is generated by default when the report is run in the report viewer.

- Default Layout

  When multiple layouts are available for the report, the default layout is presented first in the report viewer. Select this box for your custom layout when you want it displayed first. One layout must be selected as the default layout.

- Apply Style Template

  If a style template is assigned to this report, use this field to apply the style template to the layout.

- Active

  Active layouts are displayed to report consumers. Inactivate the layout to make it unavailable to report consumers.

- View Online

  Layouts that can be viewed online are available to report consumers from the report viewer. If this property is not checked, the layout is available only for scheduled jobs.

Task: Delete a Layout

To delete a layout from the report:

1. In the report editor, click **View a List**.
2. Locate the layout that you wish to delete in the table and click anywhere within its row to select it.
3. Click the **Delete** toolbar button. Click **OK** in the confirmation dialog.

7.2.2.1 Customizing RTF Templates

Most templates delivered with Oracle Fusion Applications are RTF templates. An RTF template is a rich text format file that contains the layout instructions for BI Publisher to use when generating the report. RTF templates are created using Microsoft Word.

Oracle BI Publisher provides an add-in to Microsoft Word to facilitate the coding of layout instructions. For more information see the "Creating RTF Templates Using the Template Builder for Word" chapter in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.

> **Note:** If you are designing a new layout for the report, consider using the BI Publisher layout editor. The layout editor is an online layout editing tool launched from the report editor. See Section 7.2.2.2, "Customizing BI Publisher Templates."

Before you begin:

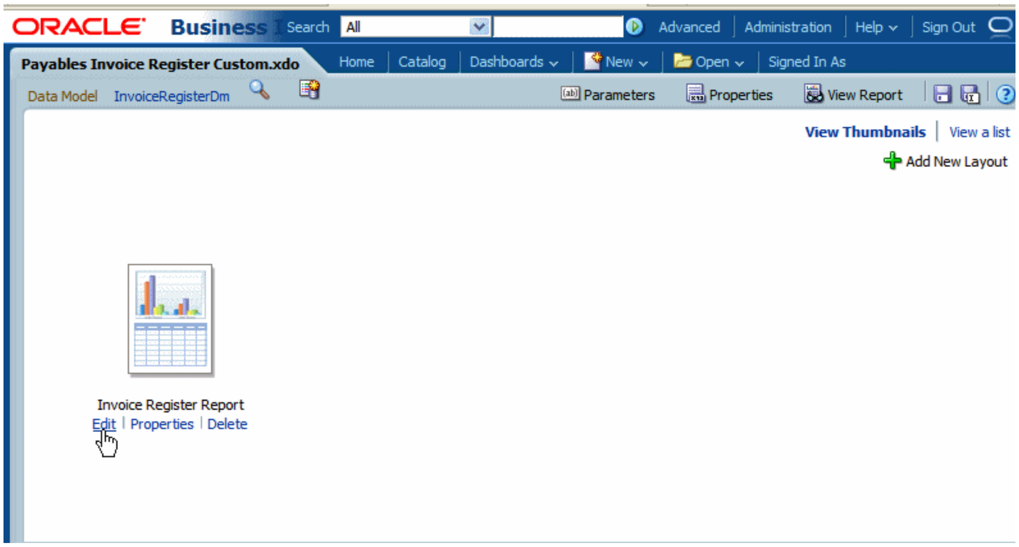
Ensure that your local client has a supported version of Microsoft Word. Oracle BI Publisher provides the Template Builder for Microsoft Word to facilitate RTF template design. Download the tool from the Oracle Business Intelligence home page. For more information see the "Creating RTF Templates Using the Template Builder for Word" chapter in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.

Task: Download the Template File

If you are creating a new layout, skip this step.

If you are modifying a predefined layout, navigate to the copy of the report under Custom in the catalog and click **Edit**. In the report editor, click the **Edit** link of the layout to download the RTF file to your local client, as shown in Figure 7–8.

Figure 7–8 Downloading the Predefined Layout



Task: Edit the RTF File in Microsoft Word

Open the downloaded RTF template file in Microsoft Word; or, if you are creating a new template, open Microsoft Word.

Load the sample data to the Template Builder for Word Add-in.

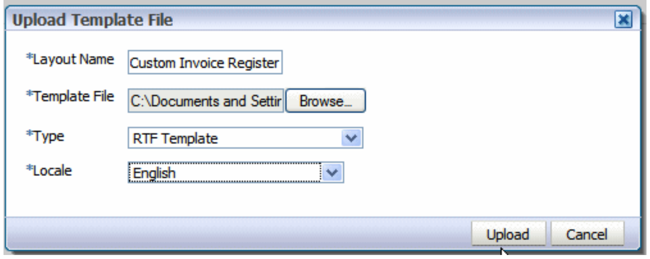
Edit or create the layout following the guidelines in the Template Builder help or in the "Creating RTF Templates Using the Template Builder for Word" chapter in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.

Ensure that you save the file as Rich Text Format (rtf).

Task: Upload the Template File to the Report Definition

In the catalog, open the report in the report editor and click **Add New Layout**. Click **Upload** to upload the RTF file from your local directory. Enter the fields in the **Upload Template File** dialog as shown in Figure 7–9.

Figure 7–9 Uploading the Template File



Upload Template File

*Layout Name	Custom Invoice Register
*Template File	C:\Documents and Settir
*Type	RTF Template
*Locale	English

Task: Add Translations for the Layout

If this report requires translations, see Section 7.2.5, "Adding Translations."

7.2.2.1.1 Customizing an RTF Template: Examples This section includes two examples of RTF template customizations:

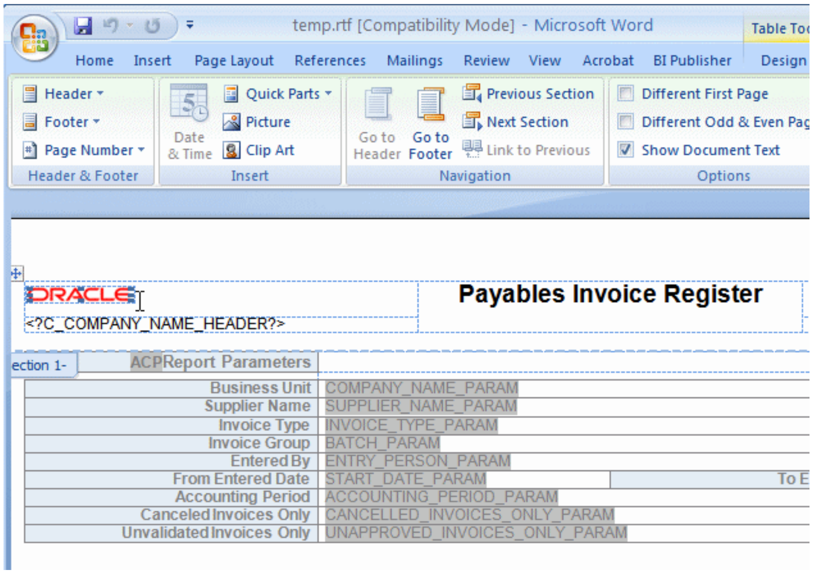
- Changing the Inserted Logo in a Predefined Template
- Customizing an RTF Template Using an Existing Data Model

Example 7–1 Changing the Inserted Logo in a Predefined Template

The Payables Invoice Register report layout includes a standard logo in the report header. To change the inserted logo using Microsoft Word 2007:

1. In Microsoft Word, delete the Oracle logo as shown in Figure 7–10.

Figure 7–10 Selecting the Oracle Logo Prior to Deletion



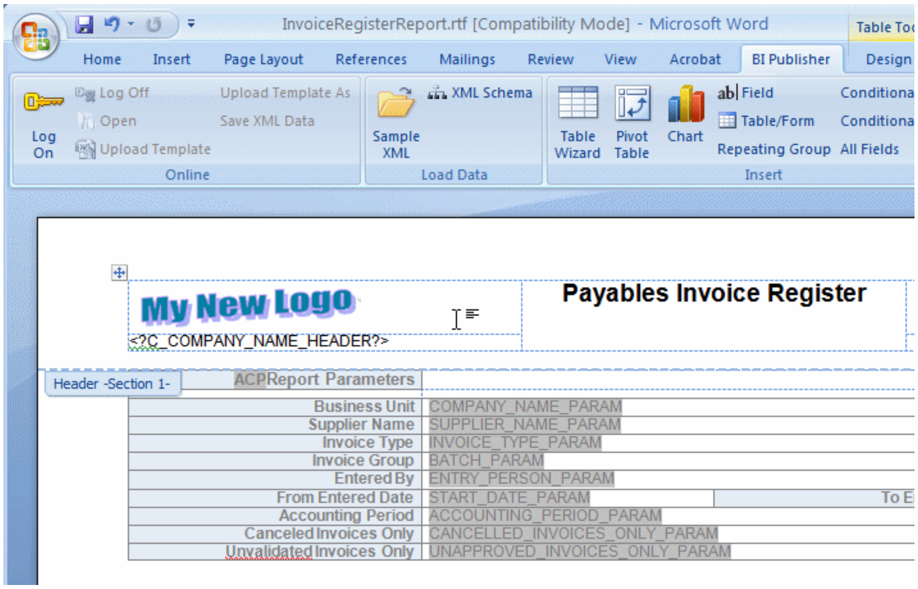
<?C_COMPANY_NAME_HEADER?>

Payables Invoice Register

ACPReport Parameters		
Business Unit	COMPANY_NAME_PARAM	
Supplier Name	SUPPLIER_NAME_PARAM	
Invoice Type	INVOICE_TYPE_PARAM	
Invoice Group	BATCH_PARAM	
Entered By	ENTRY_PERSON_PARAM	
From Entered Date	START_DATE_PARAM	To E
Accounting Period	ACCOUNTING_PERIOD_PARAM	
Canceled Invoices Only	CANCELLED_INVOICES_ONLY_PARAM	
Unvalidated Invoices Only	UNAPPROVED_INVOICES_ONLY_PARAM	

2. On the **Insert** tab in the **Illustrations** group click **Picture**. Select your company logo image file to insert it to the Word document. Resize the image as necessary. An example is shown in Figure 7–11.

Figure 7–11 Inserting the Logo in Microsoft Word



<?C_COMPANY_NAME_HEADER?>

Payables Invoice Register

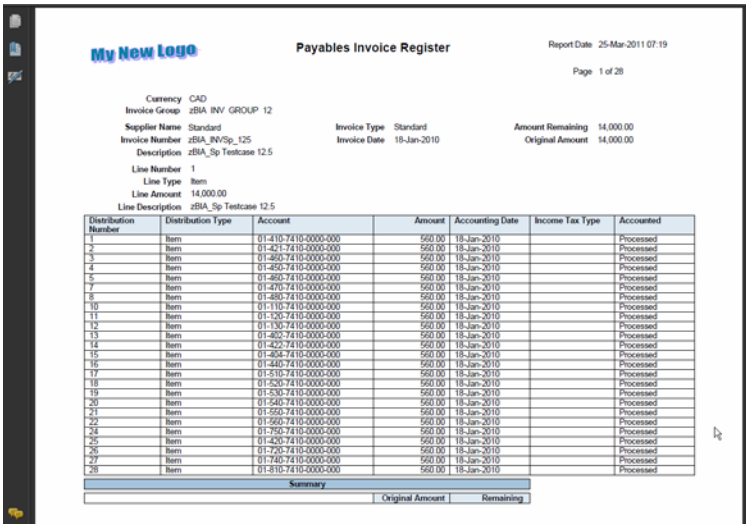
ACPReport Parameters	
Business Unit	COMPANY_NAME_PARAM
Supplier Name	SUPPLIER_NAME_PARAM
Invoice Type	INVOICE_TYPE_PARAM
Invoice Group	BATCH_PARAM
Entered By	ENTRY_PERSON_PARAM
From Entered Date	START_DATE_PARAM To E
Accounting Period	ACCOUNTING_PERIOD_PARAM
Canceled Invoices Only	CANCELLED_INVOICES_ONLY_PARAM
Unvalidated Invoices Only	UNAPPROVED_INVOICES_ONLY_PARAM

Tip: If the template file includes section breaks, you must insert the new logo for each section header.

---

3. If you downloaded sample data, you can test the template on your desktop: On the Oracle BI Publisher tab, in the **Preview** group click **PDF**. The Template Builder will apply the sample data you loaded and generate a PDF output document, as shown in Figure 7–12.

Figure 7–12 Preview of Custom Layout Template



Payables Invoice Register

Report Date 25-Mar-2011 07:19

Page 1 of 28

Currency CAD
Invoice Group zBIA INV GROUP 12

Supplier Name Standard
Invoice Number zBIA_INVSp_125
Description zBIA_Sp Testcase 12.5

Invoice Type Standard
Invoice Date 18-Jan-2010

Amount Remaining 14,000.00
Original Amount 14,000.00

Line Number 1
Line Type Item
Line Amount 14,000.00
Line Description zBIA_Sp Testcase 12.5

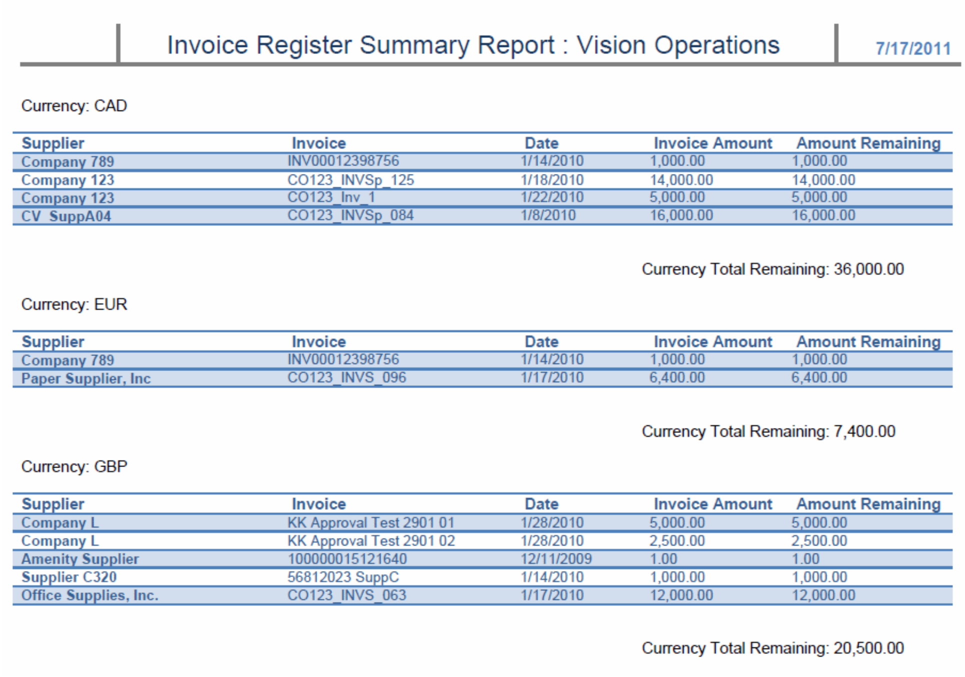
Distribution Number	Distribution Type	Account	Amount	Accounting Date	Income Tax Type	Accounted
1	Item	01-410-7410-0000-000	560.00	18-Jan-2010		Processed
2	Item	01-421-7410-0000-000	560.00	18-Jan-2010		Processed
3	Item	01-460-7410-0000-000	560.00	18-Jan-2010		Processed
4	Item	01-450-7410-0000-000	560.00	18-Jan-2010		Processed
5	Item	01-460-7410-0000-000	560.00	18-Jan-2010		Processed
7	Item	01-470-7410-0000-000	560.00	18-Jan-2010		Processed
8	Item	01-480-7410-0000-000	560.00	18-Jan-2010		Processed
10	Item	01-110-7410-0000-000	560.00	18-Jan-2010		Processed
11	Item	01-120-7410-0000-000	560.00	18-Jan-2010		Processed
12	Item	01-130-7410-0000-000	560.00	18-Jan-2010		Processed
13	Item	01-402-7410-0000-000	560.00	18-Jan-2010		Processed
14	Item	01-422-7410-0000-000	560.00	18-Jan-2010		Processed
15	Item	01-404-7410-0000-000	560.00	18-Jan-2010		Processed
16	Item	01-440-7410-0000-000	560.00	18-Jan-2010		Processed
17	Item	01-510-7410-0000-000	560.00	18-Jan-2010		Processed
18	Item	01-520-7410-0000-000	560.00	18-Jan-2010		Processed
19	Item	01-530-7410-0000-000	560.00	18-Jan-2010		Processed
20	Item	01-540-7410-0000-000	560.00	18-Jan-2010		Processed
21	Item	01-550-7410-0000-000	560.00	18-Jan-2010		Processed
22	Item	01-560-7410-0000-000	560.00	18-Jan-2010		Processed
24	Item	01-750-7410-0000-000	560.00	18-Jan-2010		Processed
25	Item	01-420-7410-0000-000	560.00	18-Jan-2010		Processed
26	Item	01-720-7410-0000-000	560.00	18-Jan-2010		Processed
27	Item	01-740-7410-0000-000	560.00	18-Jan-2010		Processed
28	Item	01-810-7410-0000-000	560.00	18-Jan-2010		Processed

Summary		
	Original Amount	Remaining

Example 7–2 Customizing an RTF Template Using an Existing Data Model

This example demonstrates the general steps for customizing an RTF template using an existing data model. In this example, the Payables Invoice Register data model is used to create a new layout to display a summary for each currency. This example demonstrates general report layout concepts and features of the Oracle BI Publisher Template Builder for Microsoft Word. Follow the steps in this topic and the guidelines in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.

The final report layout is shown in the following figure:



Invoice Register Summary Report : Vision Operations 7/17/2011

Currency: CAD

Supplier	Invoice	Date	Invoice Amount	Amount Remaining
Company 789	INV00012398756	1/14/2010	1,000.00	1,000.00
Company 123	CO123_INVSp_125	1/18/2010	14,000.00	14,000.00
Company 123	CO123_Inv_1	1/22/2010	5,000.00	5,000.00
CV_SuppA04	CO123_INVSp_084	1/8/2010	16,000.00	16,000.00

Currency Total Remaining: 36,000.00

Currency: EUR

Supplier	Invoice	Date	Invoice Amount	Amount Remaining
Company 789	INV00012398756	1/14/2010	1,000.00	1,000.00
Paper Supplier, Inc	CO123_INVS_096	1/17/2010	6,400.00	6,400.00

Currency Total Remaining: 7,400.00

Currency: GBP

Supplier	Invoice	Date	Invoice Amount	Amount Remaining
Company L	KK Approval Test 2901 01	1/28/2010	5,000.00	5,000.00
Company L	KK Approval Test 2901 02	1/28/2010	2,500.00	2,500.00
Amenity Supplier	100000015121640	12/11/2009	1.00	1.00
Supplier C320	56812023 SuppC	1/14/2010	1,000.00	1,000.00
Office Supplies, Inc.	CO123_INVS_063	1/17/2010	12,000.00	12,000.00

Currency Total Remaining: 20,500.00

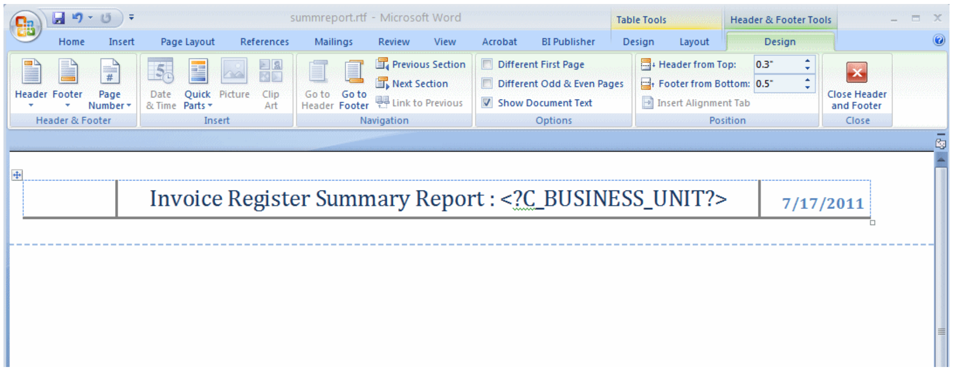
To create this layout:

1. Download to your local client sample data from the Payables Invoice Register data model (InvoiceRegisterDM).

2. In Microsoft Word, on the **BI Publisher** tab, in the **Load Data** group, click **Sample Data**. Select your downloaded file and click OK.

3. Using Microsoft Word functionality, insert the page header. Type the text for the header, and insert the field for the Business Unit as shown in Figure 7–13.

Figure 7–13 Inserting the Header to the RTF Template



Header from Top: 0.3"
Footer from Bottom: 0.5"

Invoice Register Summary Report : <?C_BUSINESS_UNIT?>

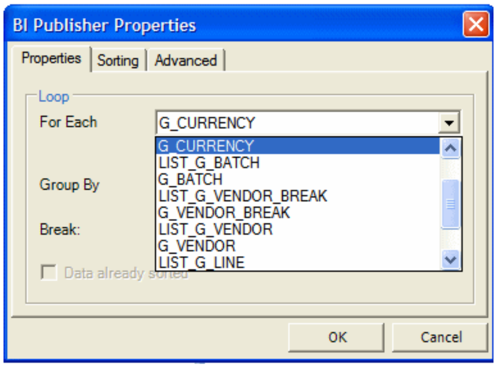
7/17/2011

4. In this example, the report will repeat the table of invoices for each occurrence of the currency code. To create this behavior, insert a repeating group based on the element C_CURRENCY_CODE. To insert the repeating group:

   a. On the **BI Publisher** tab, in the **Insert** group, click **Repeating Group**.

   b. In the **BI Publisher Properties** dialog, select the G_CURRENCY group as shown in Figure 7–14.

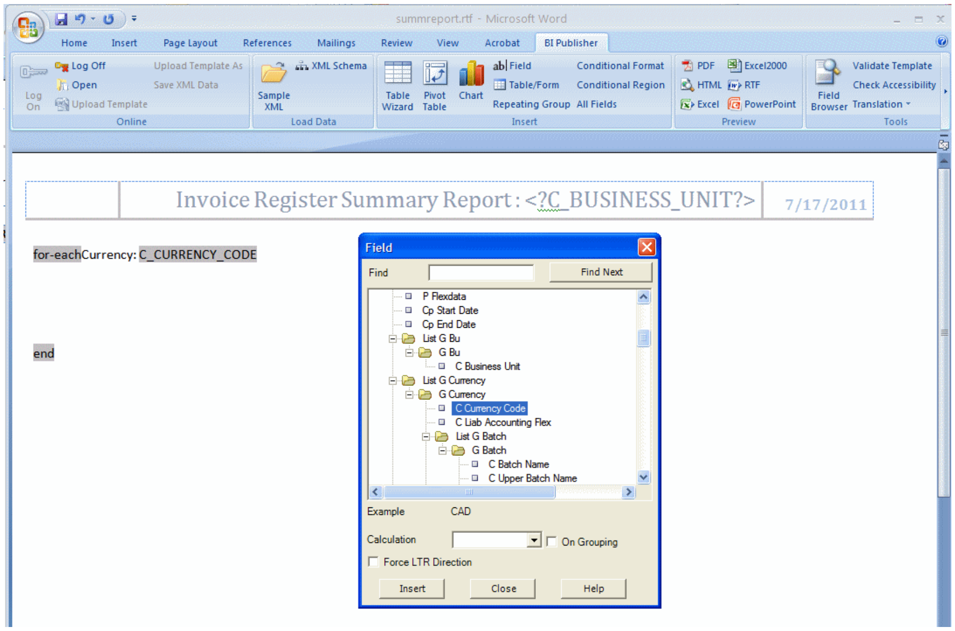
Figure 7–14 Selecting a Repeating Group



For Each: G_CURRENCY

   c. To display the field value, type the text "Currency:" after the for-each tag. To insert the element from the data: On the **BI Publisher** tab, in the **Insert** group, click **Field**. The **Field** dialog will display. Select the C_CURRENCY_CODE element, as shown in Figure 7–15.

Figure 7–15 Inserting the Currency Code Field



summreport.rtf - Microsoft Word

Invoice Register Summary Report : <?C_BUSINESS_UNIT?> 7/17/2011

for-eachCurrency: C_CURRENCY_CODE

end

Field

Find

Example CAD

Calculation

5. Use the table wizard to insert the table: On the **BI Publisher** tab in the **Insert** group, click **Table Wizard**. Make the following selections:

   - For the report format, select **Table**.
   - For the **Data Set** select the VENDOR group.
   - Select the fields to display in your table: C Vendor Name, C Invoice Num, C Invoice Date, C Invoice Amount, and C Amount Rem, as shown in Figure 7–16.

Figure 7–16 Selecting Fields for the Table

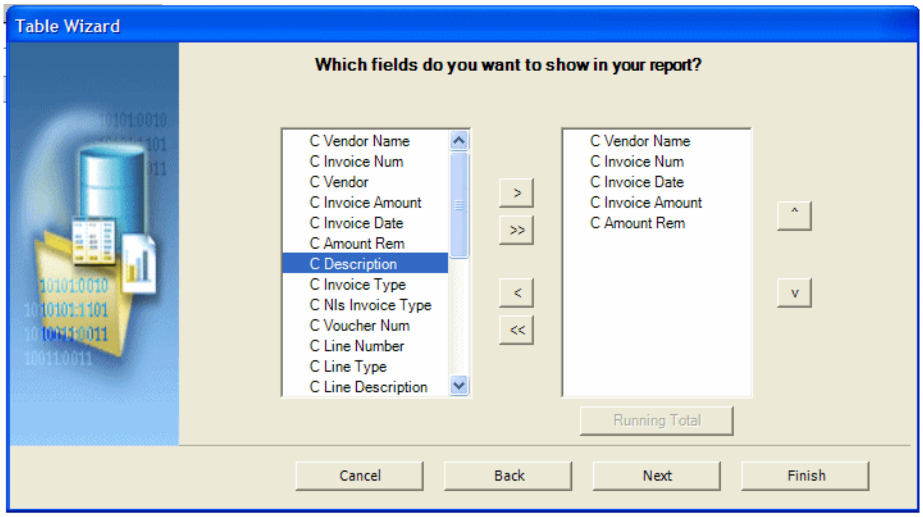


Table Wizard

Which fields do you want to show in your report?

C Vendor Name
C Invoice Num
C Invoice Date
C Invoice Amount
C Amount Rem

   - Click **Finish**. The inserted table will display with the column names from the data; also, if you preview the report, you will notice that no formatting is applied to number and date fields.

6. Edit the column names by simply editing the text in the column header row.

7. Apply formatting to the date and number fields. To apply formatting to the date field:

> **Note:** This example shows simple number formatting. If your report requires locale-driven number and date formatting, see the topic "Formatting Numbers, Dates, and Currencies" in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.

   a. Right-click the date field in the table and select **BI Publisher** then **Properties** from the menu.

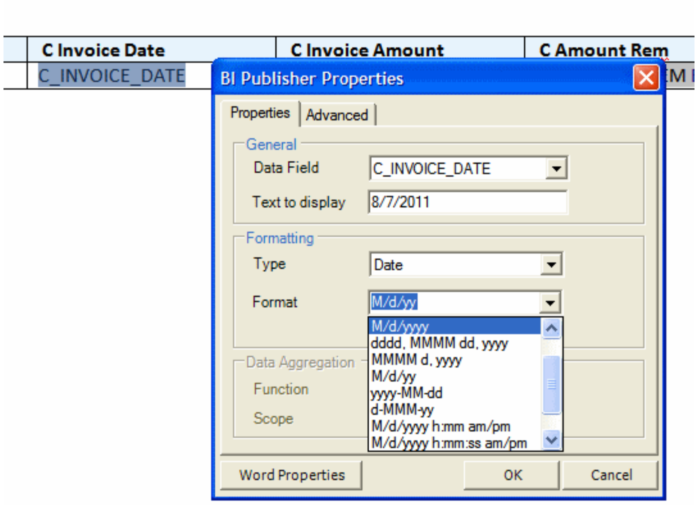
   b. In the **BI Publisher Properties** dialog, update the following (shown in Figure 7–17):

   Set the **Type** to Date.

   Select the date **Format** from the list.

   Enter **Text to display**, for example: 8/7/2011.

Figure 7–17 Formatting the Date Field



Data Field: C_INVOICE_DATE

Text to display: 8/7/2011

Type: Date

Format: M/d/yy

To apply formatting to a number field:

   a. Right-click the amount field in the table and select **BI Publisher** then **Properties** from the menu.

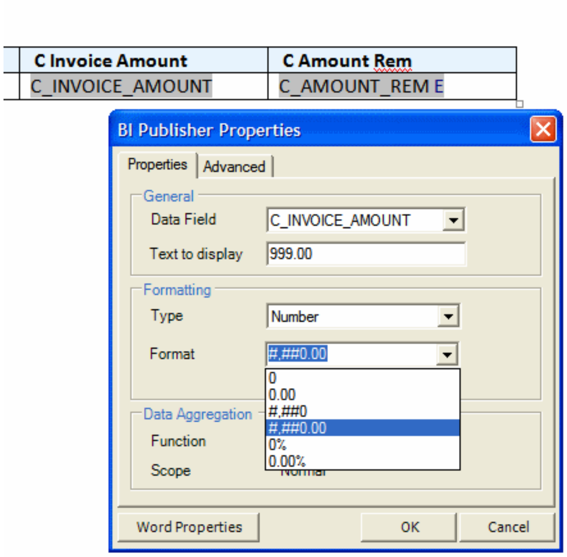
   b. In the **BI Publisher Properties** dialog, update the following (shown in Figure 7–18):

   Set the **Type** to Number

   Select the number **Format** from the list.

   Enter **Text to display**, for example: 999.00.

Figure 7–18 Formatting the Number Field



C Invoice Amount	C Amount Rem
C_INVOICE_AMOUNT	C_AMOUNT_REM E

Data Field: C_INVOICE_AMOUNT

Text to display: 999.00

Type: Number

Format: #,##0.00

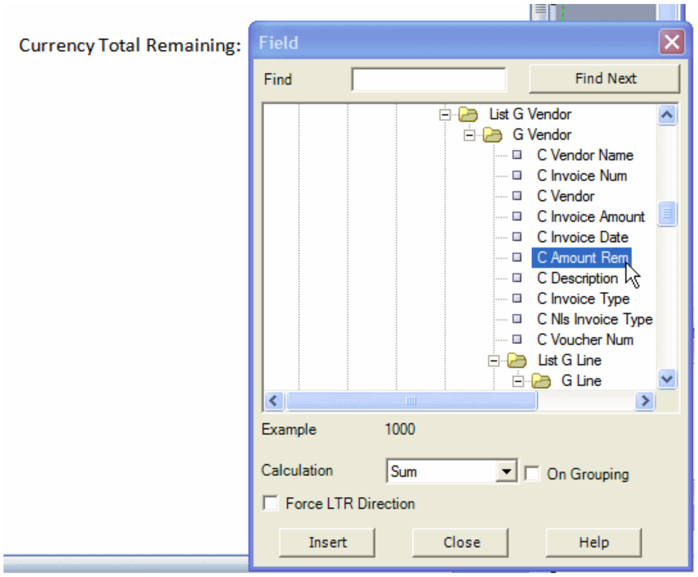
8. To display the total for each currency, enter the text: "Currency Total Remaining:" beneath the table, but inside the for-each / end tags. Insert the field as follows:

   a. On the **BI Publisher** tab, in the **Insert** group, click **Field**.

   b. In the **Insert Field** dialog, click the "C Amount Rem" field (under the G Vendor group).

   c. In the **Calculation** field, select **Sum** from the list as shown in Figure 7–19.

Figure 7–19 Inserting and Summing the Total Field



Currency Total Remaining:

Example 1000

Calculation: Sum

☐ On Grouping

☐ Force LTR Direction

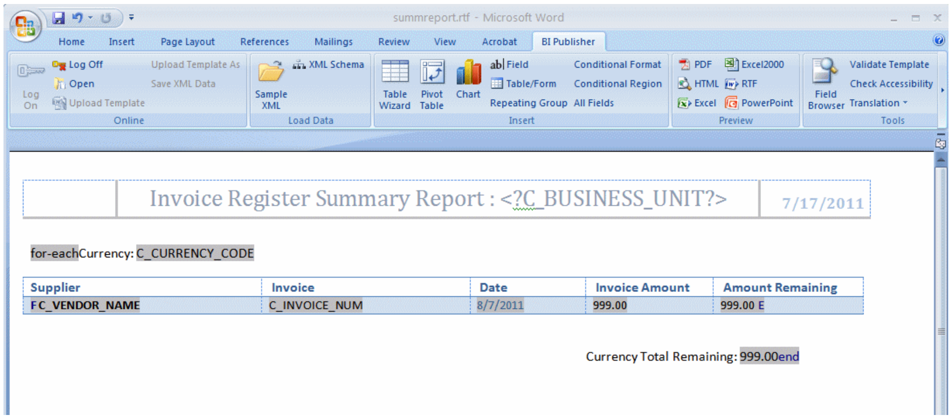
   d. Click **Insert**.

   e. Format the field as a number, as described in Step 7.

9. Perform any additional formatting of colors, fonts, borders, and text strings using Microsoft Word functionality.

The completed template will appear as shown in Figure 7–20. To preview the template, click the **PDF** button in the Oracle BI Publisher **Preview** group.

Figure 7–20 Finished RTF Template



Invoice Register Summary Report : <?C_BUSINESS_UNIT?> 7/17/2011

for-eachCurrency: C_CURRENCY_CODE

Supplier	Invoice	Date	Invoice Amount	Amount Remaining
F C_VENDOR_NAME	C_INVOICE_NUM	8/7/2011	999.00	999.00 E

Currency Total Remaining: 999.00end

7.2.2.2 Customizing BI Publisher Templates

BI Publisher templates are created using the BI Publisher Layout Editor - a design tool that provides an intuitive, drag-and-drop interface for creating pixel perfect reports in PDF, RTF, Excel, PowerPoint, and HTML. It also provides dynamic HTML output that supports lightweight interaction through a browser.

BI Publisher layouts are best suited for reports of simple to medium complexity. The interactive view is only available for BI Publisher layouts, therefore choose this layout type when you want your report consumers to interact with the report (change sorting, apply filters, and so on).

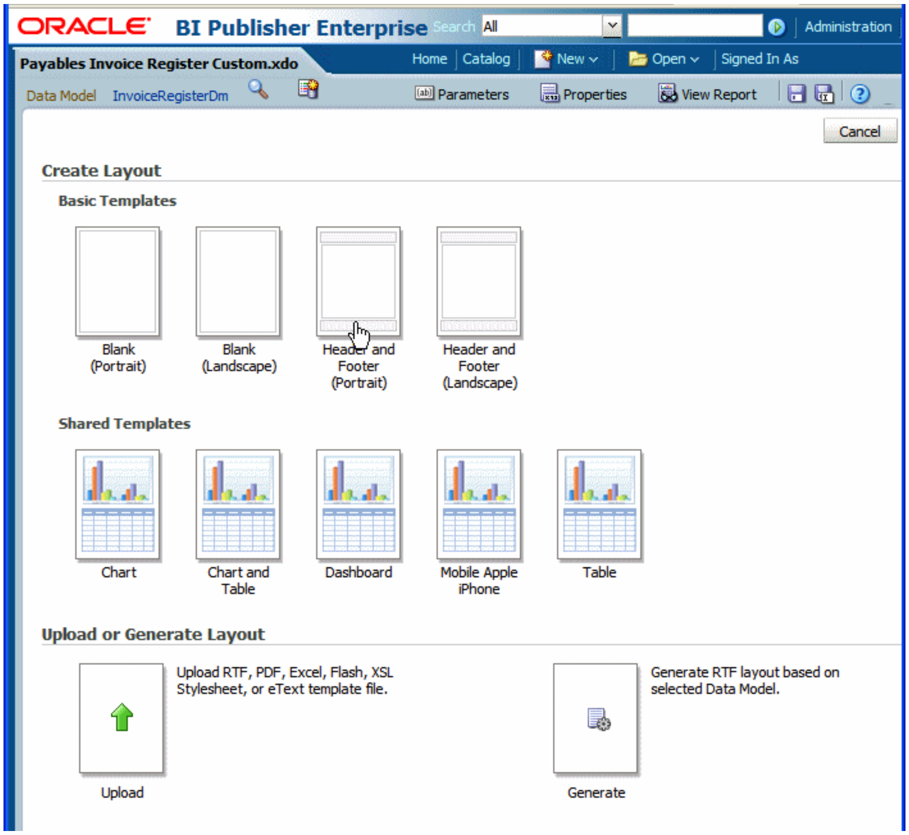
Before you begin:

The BI Publisher layout editor requires the data model to include sample data. To save sample data to the data model, you must generate data according to the first option described in "Task: Generate Sample Data from the Report".

Task: Launch the Layout Editor from the Report Definition

Navigate to the report and click **Customize**. Click **Add New Layout**. Under the **Create Layout** region, click one of the Basic Templates or Shared Templates to launch the layout editor, as shown in Figure 7–21.

Figure 7–21 Selecting a Boilerplate to Launch the Layout Editor


ORACLE BI Publisher Enterprise
Search All
Administration
Payables Invoice Register Custom.xdo
Home | Catalog | New | Open | Signed In As
Data Model InvoiceRegisterDm
Parameters
Properties
View Report
Cancel
Create Layout
Basic Templates
Blank (Portrait)
Blank (Landscape)
Header and Footer (Portrait)
Header and Footer (Landscape)
Shared Templates
Chart
Chart and Table
Dashboard
Mobile Apple iPhone
Table
Upload or Generate Layout
Upload RTF, PDF, Excel, Flash, XSL Stylesheet, or eText template file.
Upload
Generate RTF layout based on selected Data Model.
Generate

Task: Create and Save the Layout

Create the layout using the guidelines in the online help or in the "Creating BI Publisher Layout Templates" chapter in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*. Click **Save** to save the layout to the report definition.

Task: Add Translations for the Layout

If this report requires translations, see Section 7.2.5, "Adding Translations."

7.2.3 Customizing Data Models

Note: To create or edit data models you must be granted a role that includes the BI Publisher Data Model Developer role. To assign this role for editing data models, follow the guidelines in the "Configuring Roles" section in the *Oracle Fusion Applications Administrator's Guide*. For Oracle Cloud implementations, request this role by logging a service request using My Oracle Support at `https://support.oracle.com`.

A data model defines the source and structure of the data for a report. At runtime Oracle BI Publisher executes the data model to supply the XML data for a report. Create a custom data model when the data models supplied by your application do not provide the data required in your report. If you need to customize the data that is captured by the report data model, you can either copy and edit an existing data model or create a new data model.

In BI Publisher the Oracle Fusion Applications tables are provisioned as follows:

- ApplicationDB_FSCM - includes the Oracle Fusion Financials applications data
- ApplicationDB_CRM - includes the Oracle Fusion Customer Relationship Management applications data
- ApplicationDB_HCM - includes the Oracle Fusion Human Capital Management applications data

Before You Begin: Understand the Use of Parameter View Objects with Oracle Enterprise Scheduler

Editing Oracle Enterprise Scheduler jobs is not supported in Oracle Cloud implementations.

If the report requires user input for parameter values and the report is submitted by Oracle Enterprise Scheduler to Oracle BI Publisher, Oracle Enterprise Scheduler uses a parameter view object to present and validate parameter values in the Oracle Fusion application. The values are then sent by Oracle Enterprise Scheduler to BI Publisher for execution of the data model.

In the parameter view object the parameters are defined as attributes and must be named incrementally as ATTRIBUTE1, ATTRIBUTE2, and so on. In the BI Publisher data model, you must define the parameters in the same order as they are defined in the parameter view object. The mapping of the value passed by Oracle Enterprise Scheduler to the BI Publisher data model is by order alone.

For example, in a BI Publisher data model assume you have defined the following parameters in this order:

- P_BUSINESS_UNIT
- P_VENDOR_ID
- P_INVOICE_TYPE

In the parameter view object you must define ATTRIBUTE1 to retrieve the value for P_BUSINESS_UNIT; ATTRIBUTE2 to retrieve the values for P_VENDOR_ID; and ATTRIBUTE3 to retrieve the values for P_INVOICE_TYPE.

Ensure that any edits you make to the ordering of parameters in the parameter view object, the BI Publisher data model, or the Oracle Enterprise Scheduler job definition are reflected in all places. See Section 7.2.6.2, "Customizing Parameters for Reports Submitted Through Oracle Enterprise Scheduler."

7.2.3.1 Editing Existing Data Models

Do not directly edit a data model delivered with an Oracle Fusion application. Instead, copy the data model into the Custom folder and edit the copy.

Task: Copy the Existing Data Model

Navigate to the data model in the Oracle BI Presentation Catalog. To make a copy:

1. Click anywhere in the object's row to select the data model.
2. On the catalog toolbar, click the **Copy** toolbar button; then navigate to the destination location in the Custom folder and click the **Paste** toolbar button to paste the copy into the folder.

3. To rename the copy, click the **More** link, and then click **Rename**. Enter the new name, for example: InvoiceRegisterDM Custom.

Task: Customize the Data Model in the Data Model Editor

Click the **Edit** link in the catalog to open the data model in the data model editor. See the "Using the Data Model Editor" chapter in the *Oracle Fusion Middleware Data Modeling Guide for Oracle Business Intelligence Publisher (Oracle Fusion Applications Edition)* for general information about using the data model editor and the specific chapter for the component you wish to edit, for example:

- To add or delete a field from the SQL query, see the topic "Editing an Existing Data Set" in the chapter "Creating Data Sets."
- To add or delete a bursting definition, see the chapter "Adding Bursting Definitions."
- To edit parameters, see "Adding Parameters and Lists of Values."

> **Important:** If the report will use Oracle Enterprise Scheduler to collect parameter values from end users, then the parameter logic, list of values and display attributes that are presented in the application interface are defined in a parameter view object and the Oracle Enterprise Scheduler job definition. The parameter values are then passed to BI Publisher when the job is submitted.
>
> In this case, edit the parameters in the parameter view object then ensure that the parameters in the BI Publisher data model are in the same order as they are defined in the parameter view object. Do not create the list of values in the BI Publisher data model for reports submitted via Oracle Enterprise Scheduler.

7.2.3.2 Creating a New Data Model

> **Note:** To create or edit data models you must be granted a role with the BI Author role and the developDataModel permission. To create the custom role for editing data models, follow the guidelines in the "Configuring Roles" section in the *Oracle Fusion Applications Administrator's Guide*. For Oracle Cloud implementations, request this role by logging a service request using My Oracle Support at `https://support.oracle.com`.

To create a data model:

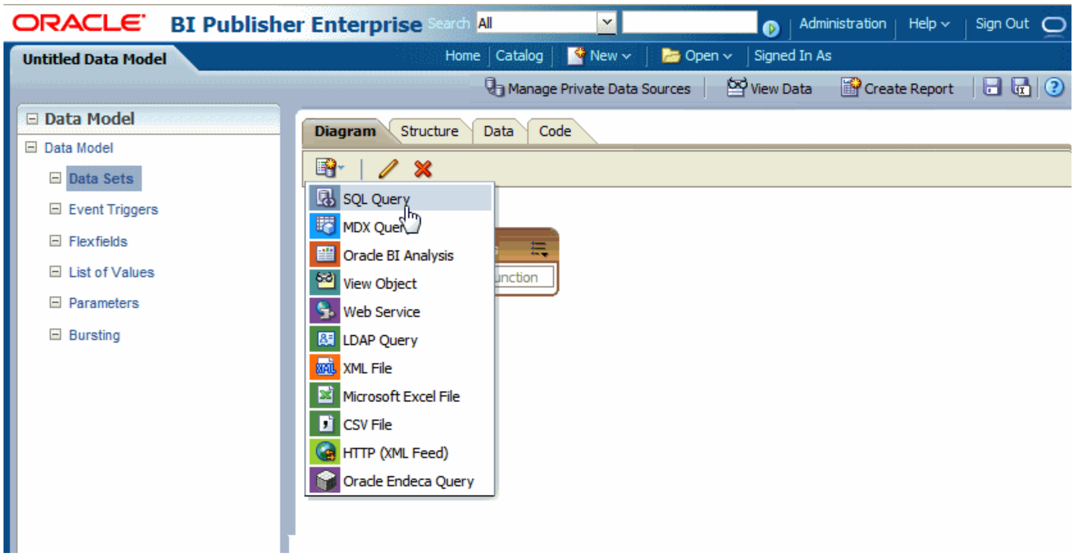
1. Open the Data Model editor.

   On the global header, click **New**, then click **Data Model** to open the data model editor.

2. Configure the data model properties. For more information see the "Setting Data Model Properties" section in the *Oracle Fusion Middleware Data Modeling Guide for Oracle Business Intelligence Publisher (Oracle Fusion Applications Edition)*.
3. If your data model will include event triggers you must enter the **Oracle DB Default Package**.
4. Create the data set.

Typically, you create data sets from SQL queries against your Oracle Fusion application data tables. The data model editor also supports using an Oracle BI analysis as its source of data as well as entity view objects created in Oracle JDeveloper. See the "Creating Data Sets" chapter in the *Oracle Fusion Middleware Data Modeling Guide for Oracle Business Intelligence Publisher (Oracle Fusion Applications Edition)* for more information about all supported data set types and how to create them. Figure 7–22 shows the menu of data set types available.

Figure 7–22 Creating a SQL Data Set



5. Create the optional components.

   A data model can include the following components:

   - Event triggers
   - **Flexfields**
   - Lists of values
   - Parameters

   Important: See "Before You Begin: Understand the Use of Parameter View Objects with Oracle Enterprise Scheduler" for information about how to define parameters in the BI Publisher data model when the report is submitted by Oracle Enterprise Scheduler. This functionality is not available in Oracle Cloud implementations.

   - Bursting definition

   See the *Oracle Fusion Middleware Data Modeling Guide for Oracle Business Intelligence Publisher (Oracle Fusion Applications Edition)* for details on creating these components.

6. Add sample data to your data model. See "Task: Generate Sample Data from the Report" for the steps for adding sample data to a data model.

7. Save your data model to the Custom folder.

7.2.4 Creating Custom Reports

Create a custom report when the reports delivered with your Oracle Fusion application do not provide the data you need; or, if you want to use a predefined data model but change other properties of the report. Save your custom report to the Custom folder.

Task: Create the Data Model

If you are using a predefined data model, skip this step.

To create a data model, see Section 7.2.3.2, "Creating a New Data Model."

Task: Create the Report Definition in the Catalog and Choose the Data Model

On the global header, click **New**, click **Report**., then click Use **Existing Data Model**. Select the data model to use for the data source for this report.

Task: Create and Add the Layout

Follow the tasks described in the section "Creating Custom Layouts" beginning with "Task: Edit or Create the Layout".

Task: Configure Report Properties

You can configure a variety of properties to set specific formatting, caching, and processing options for your report. To access the **Properties** dialog, click **Properties** in the report editor. For information on report properties see the "Configuring Report Properties" section in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.

Task: (Conditional) Create an Oracle Enterprise Scheduler Job to Run the Custom Report

This task does not apply to Oracle Cloud implementations.

To schedule this report from an Oracle Fusion application, you must create an Oracle Enterprise Scheduler job and job definition. This may also require creating a parameter view object. See Section 7.2.6, "Tasks Required to Run Custom Reports with Oracle Enterprise Scheduler."

Task: (Conditional) Enable Access to the Report Through the Reports and Analytics Pane

See Section 7.2.8, "Making Reports Available to Users in the Reports and Analytics Pane."

7.2.5 Adding Translations

Template translation is a feature of BI Publisher that enables the extraction of translatable strings from a single RTF-based template or a single BI Publisher layout template. Use this option when you need multiple translations of the final report document; for example, you need to generate invoices for both German and French customers.

For information on adding translations for your custom report layouts, see the "Translating Individual Templates" chapter in the *Oracle Fusion Middleware Report Designer's Guide for Oracle Business Intelligence Publisher*.

7.2.6 Tasks Required to Run Custom Reports with Oracle Enterprise Scheduler

Note: When you use the Customize feature to add a custom layout to an existing report, you do not need to edit the Oracle Enterprise Scheduler job.

Note: Oracle Enterprise Scheduler jobs are not customizable for Oracle Cloud implementations. Instead, schedule custom reports using the Oracle BI Publisher scheduler. For reports appropriate to run online, you can also run them on demand through the Reports and Analytics pane or from the Oracle Business Intelligence catalog.

Many Oracle Fusion applications use Oracle Enterprise Scheduler to submit report requests to Oracle BI Publisher. For reports that require parameter value input from users, Oracle Enterprise Scheduler uses a parameter view object to collect and validate parameter values to send to BI Publisher.

When you create a new report in Oracle BI Publisher, additional tasks are required to set up the Oracle Enterprise Scheduler job definition to run your report from Oracle Fusion applications. If your report customization includes customizing parameters that are passed to Oracle BI Publisher, you will likely need to customize the parameter view object.

This section highlights the additional tasks required for these related components. You must also reference the following documents for full descriptions of how to customize them and how to configure the Oracle Enterprise Scheduler job definition to integrate the parameter view object:

- For customizing Oracle Enterprise Scheduler job definitions, see the "Customizing and Extending Oracle Enterprise Scheduler Jobs" chapter in the *Oracle Fusion Applications Extensibility Guide for Developers*.
- For information on how to customize view objects in Oracle Fusion Applications, see the "Customizing and Extending Oracle ADF Application Artifacts" chapter in the Oracle Fusion Applications Extensibility Guide for Developers.
- For the full description of view objects and creating them in JDeveloper, see the "Defining SQL Queries Using View Objects" chapter in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*.

7.2.6.1 Creating a New Oracle Enterprise Scheduler Job Definition

To create an Oracle Enterprise Scheduler Job Definition:

Follow the instructions for creating a job definition in the "Customizing and Extending Oracle Enterprise Scheduler Jobs" chapter in the *Oracle Fusion Applications Extensibility Guide for Developers*. Note the following for Oracle BI Publisher jobs:

- **Job Type**: Select BIPJobType
- **ReportID**: Enter the path to the report in the Oracle BI Presentation Catalog, starting with the folder beneath Shared Folders; for example: Custom/Financials/Payables/Payables InvoiceRegisterCustom.xdo

Tip: Ensure that you include the .xdo extension for the report object.

- **Default Output**: Select a default output format.
- **Bursting Job**: If the output from this job is to be burst, select this box. The BI Publisher report must have a bursting definition to use this option. When the report is executed, the output and delivery options are determined by the bursting definition. For information on setting up a bursting definition, see the "Adding Bursting Definitions" chapter in the *Oracle Fusion Middleware Data Modeling Guide for Oracle Business Intelligence Publisher (Oracle Fusion Applications Edition)*.
- Define the property `parametersVO` to point to the parameter view object you defined, if your custom job requires parameter input.

7.2.6.2 Customizing Parameters for Reports Submitted Through Oracle Enterprise Scheduler

The parameter view object is a view object used by Oracle Enterprise Scheduler to collect user input for report parameters that Oracle Enterprise Scheduler then sends to BI Publisher. The parameter view object defines the display of the parameters in the Oracle Enterprise Scheduler interface and performs validation of the input. Use JDeveloper to edit the parameter view object.

Task: Create or Customize the Parameter View Object

To customize a parameter view object, see the following information:

- The "Customizing and Extending Oracle ADF Application Artifacts" chapter in the *Oracle Fusion Applications Extensibility Guide for Developers* for information about how to customize view objects in Oracle Fusion Applications.
- "Defining SQL Queries Using View Objects" chapter in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework* for the full description view objects and creating them in JDeveloper.

Task: Align Parameter Order in the BI Publisher Data Model

Ensure that the attributes defined for the parameters in the view object are in the same order as the parameters in the BI Publisher data model. See "Before You Begin: Understand the Use of Parameter View Objects with Oracle Enterprise Scheduler" for more information.

Task: Integrate the Parameter View Object in the Oracle Enterprise Scheduler Job Definition

Follow the instructions for creating a job definition in the "Extending Custom Oracle Enterprise Scheduler Jobs Using Existing Oracle Fusion Applications" section in the *Oracle Fusion Applications Extensibility Guide for Developers*.

7.2.7 Securing Custom Reports and Related Components

When you create a custom report you must secure it so that only users with appropriate privileges can run it. You have two options for securing it:

- Use an existing role
- Create a new custom role

Use an existing role when a custom report is functionally aligned with an Oracle-delivered report. In this case simply use the roles and entitlements that secure the associated Oracle-delivered report. For example, if a functional user who runs the Account Analysis report should also run or have access to the custom report, then you can use the same role and entitlements from the Account Analysis report rather than creating a custom role.

If you choose to enable the custom report for an existing role, the only actions required are to grant this role permissions on the objects in the Oracle BI Presentation Catalog and on the Oracle Enterprise Scheduler job definition. Perform the tasks: Task: Configure Permissions in the Oracle BI Presentation Catalog and Task: Grant Permissions to the Oracle Enterprise Scheduler Job Definition.

If you choose to create a custom role, keep the following in mind:

- Application roles and entitlements are administered through Oracle Authorization Policy Manager.
- You must add custom roles and entitlements to both the appropriate Fusion application stripe (FSCM, HCM, or CRM) and the Business Intelligence stripe (called "OBI") in Oracle Authorization Policy Manager.
- If you have also created an Oracle Enterprise Scheduler job to run the report, your users must also be assigned execution permissions for the job.
- All the tasks in this section are required when you create a custom role.

Task: Create the Custom Report Role in the OBI Stripe

Only a system administrator can create a new **application role**, and optionally include the role in an existing role hierarchy. For information about creating application roles, see the "Managing Policies and Policy Objects" chapter in the *Oracle Fusion Middleware Oracle Authorization Policy Manager Administrator's Guide (Oracle Fusion Applications Edition)*.

For Oracle Cloud implementations, request a new role by logging a service request using My Oracle Support at `https://support.oracle.com`.

Task: Configure Permissions in the Oracle BI Presentation Catalog

For a user to run a report, Read permissions must be granted to that user's role on every object in the Oracle BI Presentation Catalog that is used in the report. This will always include at least two objects: the report and the data model. If your report also references a subtemplate or a style template, you must also grant Read permissions on those objects. The report object requires additional grants to run, schedule, and view output.

See "Managing Objects in the BI Presentation Catalog" in the *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition)* for more information about catalog permissions.

Configuring permissions in the BI Presentation catalog consists of the following subtasks:

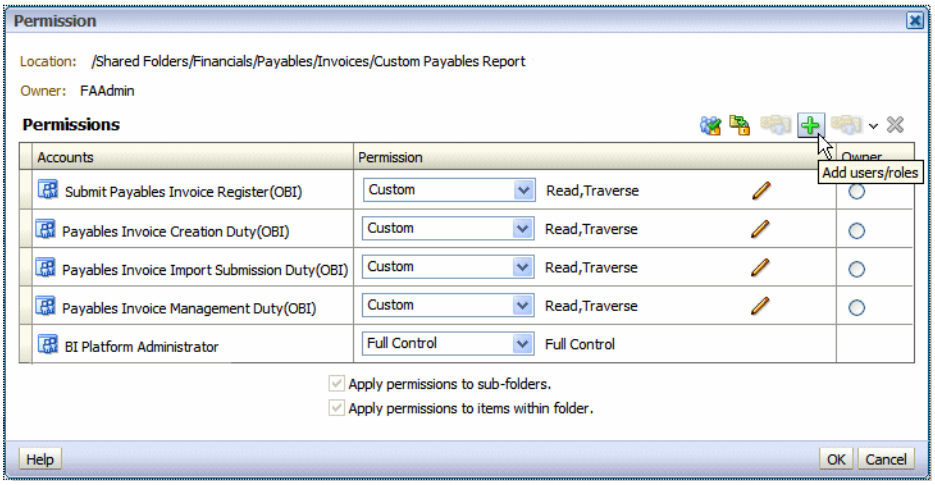
- "Grant the Role Permissions to the Report"
- "Grant the Role Permissions to the Data Model and Other Referenced Objects"
- "Delete Permissions"

Grant the Role Permissions to the Report

To grant permissions in the catalog:

1. Navigate to the report in the catalog and click **More** and then click **Permissions**. The **Permissions** dialog launches and the inherited permissions are shown.
2. In the **Permissions** dialog, click **Add users/roles** as shown in Figure 7–23.

Figure 7–23 Detail of Add Users/ Roles in the Permission Dialog



Permission

Location: /Shared Folders/Financials/Payables/Invoices/Custom Payables Report

Owner: FAAdmin

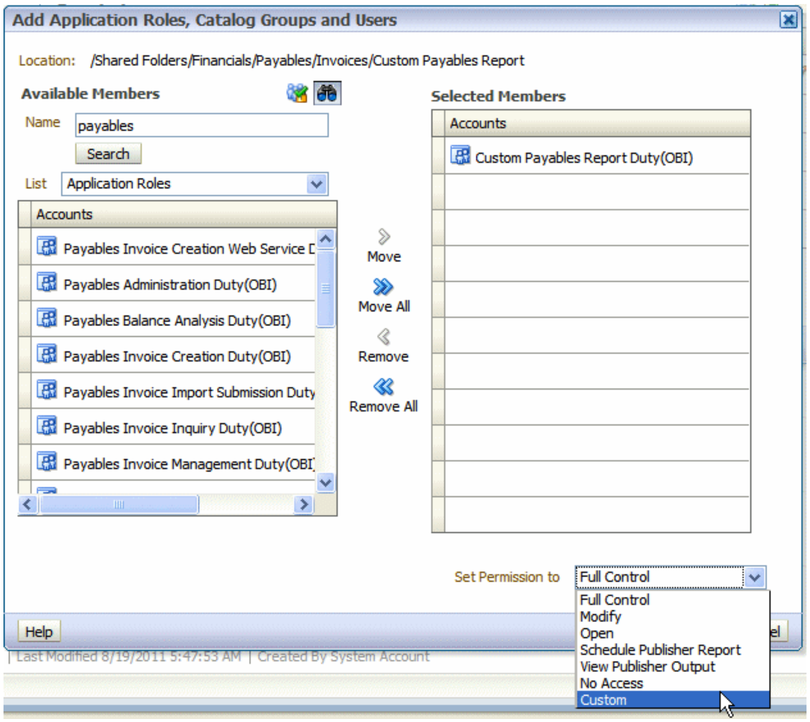
Permissions

Accounts	Permission		Owner
Submit Payables Invoice Register(OBI)	Custom	Read,Traverse	○
Payables Invoice Creation Duty(OBI)	Custom	Read,Traverse	○
Payables Invoice Import Submission Duty(OBI)	Custom	Read,Traverse	○
Payables Invoice Management Duty(OBI)	Custom	Read,Traverse	○
BI Platform Administrator	Full Control	Full Control	

☑ Apply permissions to sub-folders.
☑ Apply permissions to items within folder.

3. In the **Add Application Roles, Catalog Groups and Users** dialog, search for your custom role and use the shuttle buttons to move it to the **Selected Members** list. In the **Set Permission to** list, select **Custom**, as shown in Figure 7–24 and then click OK.

Figure 7–24 Adding a Role to the Selected Members List



Add Application Roles, Catalog Groups and Users

Location: /Shared Folders/Financials/Payables/Invoices/Custom Payables Report

Available Members

Name: payables

List: Application Roles

Accounts
Payables Invoice Creation Web Service D
Payables Administration Duty(OBI)
Payables Balance Analysis Duty(OBI)
Payables Invoice Creation Duty(OBI)
Payables Invoice Import Submission Duty
Payables Invoice Inquiry Duty(OBI)
Payables Invoice Management Duty(OBI)

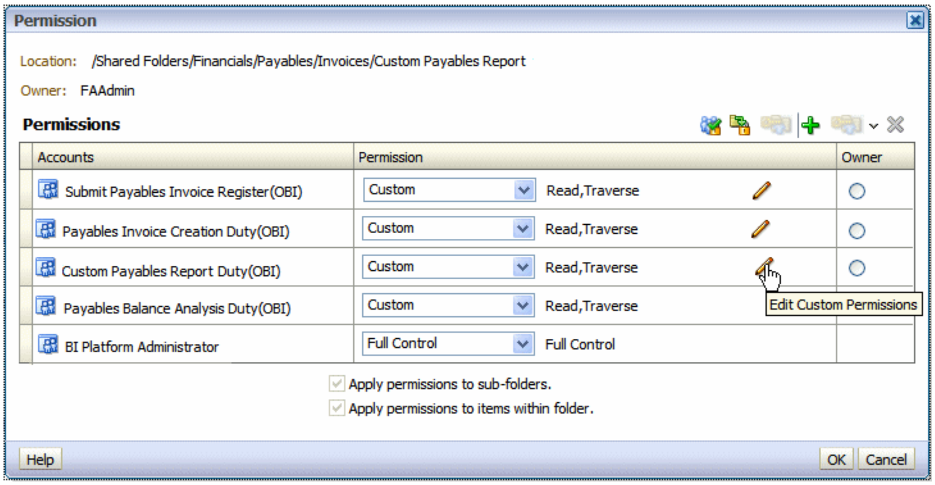
Selected Members

Accounts
Custom Payables Report Duty(OBI)

Set Permission to: Full Control

4. In the **Permissions** dialog, locate the role you added and click **Edit** as shown in Figure 7–25.

Figure 7–25 Editing the Permissions Assigned to a Role



Permission

Location: /Shared Folders/Financials/Payables/Invoices/Custom Payables Report

Owner: FAAdmin

Permissions

Accounts	Permission		Owner
Submit Payables Invoice Register(OBI)	Custom	Read,Traverse	○
Payables Invoice Creation Duty(OBI)	Custom	Read,Traverse	○
Custom Payables Report Duty(OBI)	Custom	Read,Traverse	○
Payables Balance Analysis Duty(OBI)	Custom	Read,Traverse	
BI Platform Administrator	Full Control	Full Control	

Edit Custom Permissions

☑ Apply permissions to sub-folders.

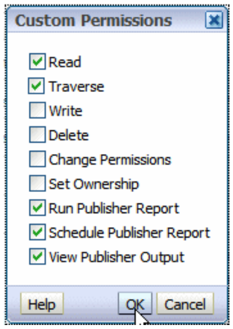
☑ Apply permissions to items within folder.

5. In the **Custom Permissions** dialog, select the permissions to enable. Typically, you will enable the following for a BI Publisher report:

   - Read - gives authority to access, but not modify, the report
   - Traverse - gives the authority to copy the report
   - Run Publisher Report - gives authority to read, traverse the folder that contains the report, and run the report.
   - Schedule Publisher Report - gives authority to read, traverse the folder that contains the report, and schedule the report
   - View Publisher Output - enables the user to view the report output generated by the scheduler

   The **Custom Permissions** dialog is shown in Figure 7–26.

Figure 7–26 Setting Custom Permissions on the Report



Custom Permissions

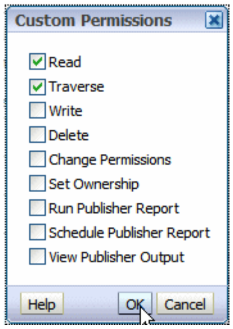
☑ Read
☑ Traverse
☐ Write
☐ Delete
☐ Change Permissions
☐ Set Ownership
☑ Run Publisher Report
☑ Schedule Publisher Report
☑ View Publisher Output

Grant the Role Permissions to the Data Model and Other Referenced Objects

1. Navigate to the data model in the catalog and click **More** and then click **Permissions**. The **Permissions** dialog launches and the inherited permissions are shown.

2. Follow the instructions in "Grant the Role Permissions to the Report" but in the **Custom Permissions** dialog, enable the **Read** and **Traverse** permissions, as shown in Figure 7–27.

Figure 7–27 Setting Custom Permissions on the Data Model



Custom Permissions

☑ Read
☑ Traverse
□ Write
□ Delete
□ Change Permissions
□ Set Ownership
□ Run Publisher Report
□ Schedule Publisher Report
□ View Publisher Output

3. Repeat this procedure for any other referenced objects (subtemplates or style templates).

Delete Permissions

If the custom report inherited permissions that you want to remove from the report, you can delete permissions as follows:

1. Navigate to the object in the catalog and click **More** and then click **Permissions**. The **Permissions** dialog launches and the inherited permissions are shown.
2. Select the permission to delete and click the **Delete** toolbar button.

Task: Grant Permissions to the Oracle Enterprise Scheduler Job Definition

This task does not apply to Oracle Cloud implementations.

If the custom report is to be scheduled through Oracle Enterprise Scheduler, execution permissions must also be granted to enable users belonging to the role to submit the job. See the "Extending Custom Oracle Enterprise Scheduler Jobs Using Existing Oracle Fusion Applications" section in the *Oracle Fusion Applications Extensibility Guide for Developers*

7.2.8 Making Reports Available to Users in the Reports and Analytics Pane

To make a report available to users through the Reports and Analytics pane, map the report to the work areas of the user roles that need access. For the procedure to map reports to work areas, see "Define Application Toolkit Configuration" in the *Oracle Fusion Applications Common Implementation Guide*.

7.2.9 Enabling Reports for Scheduling from the Reports and Analytics Pane

To enable scheduling through the Reports and Analytics pane, configure the report properties:

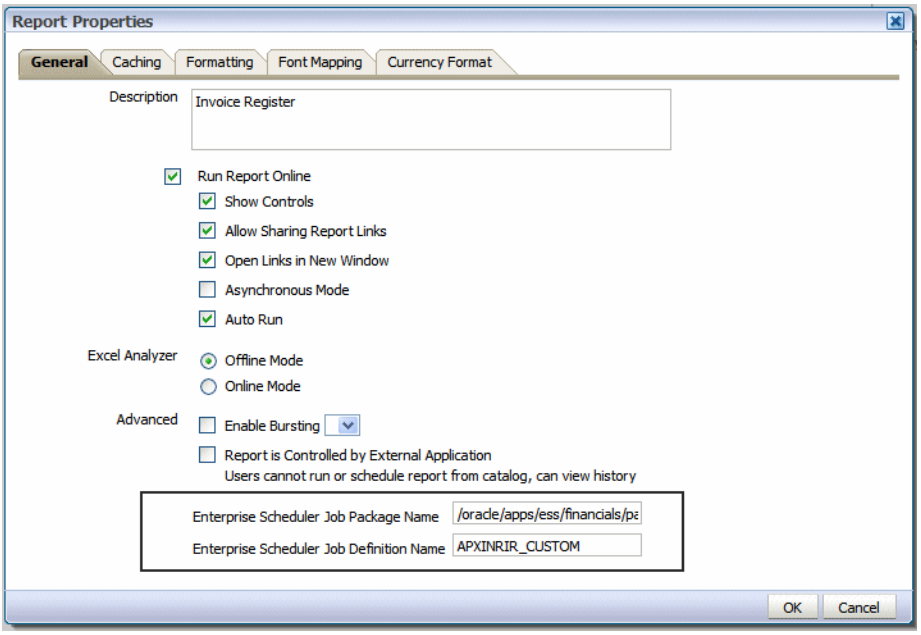
1. Navigate to the report in the Business Intelligence catalog and click **Edit.**
2. In the report editor, click **Properties**.
3. On the **Properties** dialog enter the following fields:
   - Enterprise Scheduler Job Package Name

     Enter the Path for the Job Definition. For example: /oracle/apps/ess/financials/payables/invoices/transactions/Jobs
   - Enterprise Scheduler Job Definition Name

     Enter the Name for the Job Definition. For example: APXINRIR

The **Report Properties** dialog is shown in Figure 7–28:

Figure 7–28 Report Properties to Enable Scheduling Through the Reporting Pane



Description: Invoice Register

☑ Run Report Online

☑ Show Controls

☑ Allow Sharing Report Links

☑ Open Links in New Window

☐ Asynchronous Mode

☑ Auto Run

Excel Analyzer: ● Offline Mode ○ Online Mode

Advanced: ☐ Enable Bursting

☐ Report is Controlled by External Application
Users cannot run or schedule report from catalog, can view history

Enterprise Scheduler Job Package Name: /oracle/apps/ess/financials/pa

Enterprise Scheduler Job Definition Name: APXINRIR_CUSTOM

7.3 Customizing Analytics

This section describes how to use Oracle Business Intelligence Enterprise Edition to customize and extend analytics for Oracle Fusion Applications.

7.3.1 About Customizing Analytics

Analytics are analyses and dashboards built with Oracle Business Intelligence Presentation Services, based on objects in the Oracle BI repository. Analyses are queries based on real-time, transactional or operational data that provide answers to business questions. Dashboards provide personalized views of corporate and external information. A dashboard consists of one or more pages that contain content, such as analyses, links to Web sites, BI Publisher reports, and so on.

You can customize analyses using the Oracle BI Composer interface from within Oracle Fusion Applications. You can customize dashboards using Oracle Business Intelligence Enterprise Edition.

You can also customize objects in the Oracle BI repository (RPD) using the Oracle BI Administration Tool in either online or offline mode. Use online mode only for small changes that do not require running consistency checks. Running consistency checks against the full online repository can take a long time. Instead, make more complex changes that require consistency checks in offline mode against a project extract of the repository.

Table 7–4 provides guidelines for when to perform online and offline edits. See the *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition)* for full information about how to use the Administration Tool to edit RPD files.

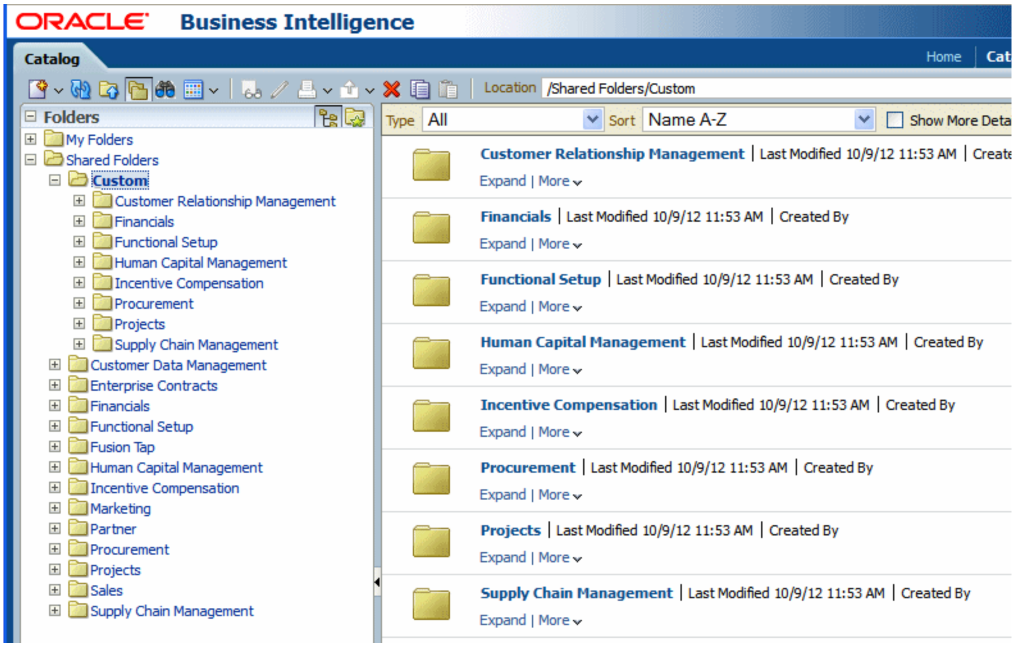
Table 7–4 Guidelines for Online and Offline RPD Edits

Mode	Use This Mode For:	Example Use Cases	Example Operations Information
Online	- Changes that do not require running a consistency check - Small changes that are required to fix things in a running system - Changes that need to be deployed quickly	- Renaming Presentation Layer metadata - Reorganizing Presentation Layer metadata	1. Connect to the RPD in online mode. 2. Check out, modify, then check in the appropriate objects. 3. In a clustered system, restart all Oracle BI Servers except for the master server to propagate the changes. You can use the Cluster Manager in the Administration Tool to identify the master Oracle BI Server. 4. Reload metadata in Oracle BI Presentation Services by clicking the **Reload Files and Metadata** link from the Administration page.
Offline	- Full-scale development or customization activities that require running consistency checks multiple times and iterating	- Configuring a **descriptive flexfield** or a **key flexfield** for Oracle Business Intelligence - Customizing existing fact or dimension tables - Adding new fact or dimension tables	1. Copy the RPD from the production computer to the Windows development computer. 2. Open the RPD in offline mode and make the appropriate changes. 3. Upload the repository using Fusion Applications Control and restart all Oracle Business Intelligence system components.

7.3.1.1 Maintaining Report Customizations in the Custom Folder

The Oracle BI Presentation catalog includes a Custom folder under Shared Folders. The Custom folder contains a subfolder for each product family, similar to the product family subfolder structure under Shared Folders. The structure of the Custom folder is shown in Figure 7–29.

Figure 7–29 Structure of the Custom Folder



Location /Shared Folders/Custom

Type All | Sort Name A-Z | □ Show More Deta

Folders
- My Folders
- Shared Folders
  - Custom
    - Customer Relationship Management
    - Financials
    - Functional Setup
    - Human Capital Management
    - Incentive Compensation
    - Procurement
    - Projects
    - Supply Chain Management
  - Customer Data Management
  - Enterprise Contracts
  - Financials
  - Functional Setup
  - Fusion Tap
  - Human Capital Management
  - Incentive Compensation
  - Marketing
  - Partner
  - Procurement
  - Projects
  - Sales
  - Supply Chain Management

Customer Relationship Management | Last Modified 10/9/12 11:53 AM | Create
Expand | More

Financials | Last Modified 10/9/12 11:53 AM | Created By
Expand | More

Functional Setup | Last Modified 10/9/12 11:53 AM | Created By
Expand | More

Human Capital Management | Last Modified 10/9/12 11:53 AM | Created By
Expand | More

Incentive Compensation | Last Modified 10/9/12 11:53 AM | Created By
Expand | More

Procurement | Last Modified 10/9/12 11:53 AM | Created By
Expand | More

Projects | Last Modified 10/9/12 11:53 AM | Created By
Expand | More

Supply Chain Management | Last Modified 10/9/12 11:53 AM | Created By
Expand | More

By default, the BI Platform Author Role is granted Full Control permissions to the product family subfolders, enabling users with this role to edit and delete the report objects that are copied to or created in the Custom folders, as well as set permissions on those objects. Oracle recommends that you copy the report objects that you wish to customize into the corresponding product family folder under the Custom folder. Maintaining custom objects in the Custom folder has the following benefits:

- Oracle patches that update the original object will not overwrite customizations in the Custom folder.
- Customized objects are easily located and identified.
- Default security facilitates the editing of objects in the Custom folder without compromising security on the original objects.

Security Considerations When Copying Objects into the Custom Folder

When you copy an object, the copied object inherits the permission settings of the folder into which it is copied. You must manually reset the permissions on the object and product subfolder structure that you create under the Custom folder to reestablish the security settings of the original object. For information on setting permissions in the catalog see Section 7.2.7, "Securing Custom Reports and Related Components."

7.3.1.2 What You Can Customize in Analytics

You can customize analyses and dashboards, as well as objects in the Oracle BI repository (RPD). In Oracle Cloud implementations you can customize analyses and edit dashboards; you cannot create new dashboards or make changes to the Oracle BI repository (RPD).

Customizations to analyses and dashboards result in changes to the Oracle BI Presentation Catalog. Be aware that some patches include updates to the Oracle BI Presentation Catalog. All new objects are preserved during the patch process; in addition, changes to existing objects are preserved when the patch does not include a new version of that object.

If you change an existing presentation catalog object and subsequent patches do include a new version of the object, the patch process detects and logs conflicts, and patching will stop. The catalog administrator must resolve any conflicts manually using Catalog Manager and then rerun the patch.

In the Oracle BI repository, non-Cloud implementations can create new repository objects such as physical columns, logical table sources, logical columns, and presentation columns. Be aware that some patches include updates to the Oracle BI repository. New objects are preserved during the patch process; in addition, changes to existing objects are preserved when the patch does not include a new version of that object.

If you change an existing object and subsequent patches do include a new version of the object, the Merge Wizard in the Administration Tool provides a method to merge the changes. For most typical customizations, the merge process is straightforward. The exception is when presentation columns have been moved across presentation tables; in this situation, it is important to plan ahead and track the changes carefully to ensure your changes are preserved during the merge.

See the "Oracle BI Applications Patching" chapter in the *Oracle Fusion Middleware Reference Guide for Oracle Business Intelligence Applications* for more information about patching the Oracle BI Presentation Catalog and Oracle BI repository.

7.3.1.3 Understanding Catalog Security and Allowed Customizations

Before you customize analytics, ensure that you have proper permissions for editing and creating Oracle Business Intelligence Presentation Catalog objects and understand how to set permissions in the catalog. For more information about setting permissions in the catalog, see the "Managing Objects in the Oracle BI Presentation Catalog" chapter in the *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition)*.

Only users with the BI Platform Administrator role (not available in Oracle Cloud implementations) can directly edit objects delivered by Oracle in the BI Presentation catalog. Edit Oracle-delivered objects directly only when necessary to maintain the proper functioning of applications or processes that reference the original objects (such as embedded dashboards or targets of navigation actions). Ensure that you understand patching implications described in the previous section.

Users with the BI Author role have full control permissions to the Custom folder located under Shared Folders. Oracle provides the Custom folder with default permissions for BI Author to facilitate customizations in a centralized location while maintaining security on the original objects. Table 7–5 summarizes the actions allowed for BI Authors.

Table 7–5 Allowed Actions for BI Author

Customization	Action Allowed for BI Author
Edit analysis	Can edit analysis but cannot save to original folder. Must save the edited version to the Custom folder or My Folders.
Copy analysis	Can copy analysis and save to the Custom folder or My Folders. Cannot save the copy to original folder.
Create new analysis	Can create a new analysis and save to the Custom folder or My Folders.
Edit dashboard	Not allowed for Oracle predefined dashboards. Use Copy dashboard.

Table 7–5 (Cont.) Allowed Actions for BI Author

Customization	Action Allowed for BI Author
Copy dashboard	Can copy dashboard and save to the Custom folder or My Folders. In the Custom folder the copy can be edited. Cannot save the copy to original folder.
Create new dashboard	Not allowed.

7.3.2 Customizing Analytics

You can customize analytics from the Reports and Analytics pane in Oracle Fusion Applications.

Task: Customize Analyses

To work with analyses, use one of the following editors:

- The "BI Composer wizard" — A simple-to-use wizard that allows you to quickly and easily create, edit, or view analyses without the complexities of the Analysis editor. For more information, see the "Using BI Composer to Work with Analyses" chapter in the *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition)*.

  You have access to this editor only if you have been granted the Access to BI Composer privilege by the administrator.

- The "Analysis editor" — A more complex editor that allows you to explore and interact with information by visually presenting data in tables, graphs, pivot tables, and so on. For more information, see the chapter "Creating Analyses" in the *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition)*

  You have access to this editor only if you have been granted the Access to Analysis privilege by the administrator.

To customize existing analyses using BI Composer:

1. From the Navigator, go to the Reports and Analytics pane and locate the analysis you want to customize.
2. Click the object and then select **Edit** to use the Oracle BI Composer to edit the object.
3. Make the changes to the analysis by navigating the steps of the wizard using the buttons in the BI Composer train at the top of the wizard.
4. When finished editing, click the **Save** button in the BI Composer train.
5. Select the desired location to save the customized analysis and click **Submit**. Oracle recommends saving customizations to the appropriate application folder under Shared Folders/Custom. You can only save to folders to which you have Write permissions.

To create new analyses using BI Composer:

1. From the Navigator, go to the Reports and Analytics pane.
2. Click **Create** and then select **Analysis**.
3. Select a subject area. The BI Composer wizard displays.
4. Create the analysis by navigating the steps of the wizard using the buttons in the BI Composer train at the top of the wizard.

5. When finished, click the **Save** button in the BI Composer train.
6. Select the desired location to save the new analysis and click **Submit**. Oracle recommends saving customizations to the appropriate application folder under Shared Folders/Custom. You can only save to folders to which you have Write permissions.

Task: Customize Dashboards

To customize dashboards:

> **Note:** Only BI Administrators can create new dashboards or edit an Oracle predefined dashboard in place. The BI Administrator role is not available in Oracle Cloud implementations.

1. From the Navigator, go to the Reports and Analytics pane and locate the dashboard you want to customize.
2. Click the dashboard name and then select **More** to go to the catalog page in Oracle Business Intelligence Enterprise Edition.
3. Copy the dashboard to a folder in the catalog to which you have Write permissions as follows:

   In the catalog, click the dashboard's **More** link and then click **Copy**. Navigate to the destination folder and click the **Paste** catalog toolbar button. Oracle recommends saving customizations to the appropriate application folder under Shared Folders/Custom. You can only save to folders to which you have Write permissions.
4. Select the dashboard copy in the saved location and click **Edit** (or if the **Edit** link is not visible, first click **More**, then **Edit**). Edit the dashboard copy and click **Save**.

For information about customizing dashboards in the catalog page, see the "Building and Using Dashboards" chapter in the *Oracle Fusion Middleware User's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition)*.

7.3.3 Customizing the Oracle BI Repository (RPD)

You can customize and extend the Oracle BI repository (RPD file).

> **Note:** Customizing the Oracle BI repository is not available in Oracle Cloud implementations.

Task: Create BI View Objects for Custom Fact and Dimension Tables

Whenever you create a custom fact or dimension table, you must create a BI view object for that table and incorporate it into the Oracle Fusion application before you can import it into the Oracle BI repository. To do this, follow these steps:

1. From a JDeveloper application workspace in the developer role, define the custom view object for the custom table. You must follow the view object guidelines for Oracle Transactional Business Intelligence as described in "Designing and Securing View Objects for Oracle Business Intelligence Applications" in the *Oracle Fusion Applications Developer's Guide*. For information about creating view objects from an application workspace, see the "Creating Custom Business Components" section in the *Oracle Fusion Applications Extensibility Guide for Developers*.

> **Tip:** When you create the custom table, you must grant the necessary privileges (such as `SELECT`) to the FUSION_BI schema user in addition to the FUSION_RUNTIME schema user. Otherwise, queries against the new table will fail.
>
> See also the "Customizing and Extending the Oracle Fusion Applications Schemas" section in the *Oracle Fusion Applications Extensibility Guide for Developers* for guidelines on creating custom tables.

2. Create an application module (AM), as described in the "Creating Custom Business Components" section in the *Oracle Fusion Applications Extensibility Guide for Developers*, and add the custom BI view object instance to the application module.
3. Create an ADF Library JAR for the custom artifacts as described in the "Deploying Oracle ADF Customizations and Extensions" section in the *Oracle Fusion Applications Extensibility Guide for Developers*
4. From a customization workspace, import the ADF Library JAR for the custom artifacts into the Oracle Fusion application and restart the host server for the application so that the customizations are picked up. For more information, see the "Customizing Oracle ADF Artifacts with JDeveloper" section in the *Oracle Fusion Applications Extensibility Guide for Developers*.
5. Continuing in the customization role, nest the BI application module in the root application module as described in the "Defining Nested Application Modules" section in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*.
6. In the customization role, create a MAR file and load the MAR file using WLST commands or the Oracle WebLogic Server Administration Console as described in the "Deploying Oracle ADF Customizations and Extensions" section in the *Oracle Fusion Applications Extensibility Guide for Developers* and restart the host server for the application so that the customizations are picked up.

Task: Modifying Existing Fact or Dimension Tables

In some cases, you might want to modify existing fact or dimension tables in the Oracle BI repository. For example, assume you want to deploy Oracle Fusion Project Portfolio Management, but use the PeopleSoft Procurement application as a source. In this situation, you would set up a custom table in Oracle Fusion Applications that populates Commitments data from PeopleSoft. Then, you would need to change the Commitments fact table in the Oracle BI repository (RPD file) to point to the new custom table.

To accomplish the task described in this example:

1. Create a custom BI view object for the custom table and incorporate it into the application as described in Task: Create BI View Objects for Custom Fact and Dimension Tables.
2. Use the Import Metadata Wizard in the Administration Tool to import the new view object into the Physical layer of the RPD under the appropriate database object. Then, join the new view object to the existing dimension view objects. You must connect as the FUSION_APPS_BI_APPID user in the Select Data Source screen of the Import Metadata Wizard.

   See the "Importing Metadata from ADF Data Sources" and "Working with Physical Foreign Keys and Joins" sections in the *Oracle Fusion Middleware Metadata*

Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition) for more information.

3. Create a new logical table source under the existing Commitment logical fact table, and map all metrics to the physical columns from the new view object. Then, deactivate the existing Commitments logical table source.

   See the "Managing Logical Table Sources (Mappings)" chapter in the *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition)* for more information.

Using this approach, all Presentation layer metadata, analyses, and dashboards will continue to work with data coming from the new physical columns.

In other cases, you might want to extend existing fact or dimension tables using existing view objects that have new attributes. For example, you might want to incorporate fields on standard Oracle Fusion Applications pages that are not currently being used for analysis into the Oracle BI repository.

To extend existing fact or dimension tables for existing view objects that have new attributes:

1. Use the Import Metadata Wizard in the Oracle BI Administration Tool to import the view objects that correspond to the Oracle Fusion Applications fields into the Physical layer of the RPD. You must connect as the FUSION_APPS_BI_APPID user in the Select Data Source screen of the Import Metadata Wizard.

   See the "Importing Metadata from ADF Data Sources" section in the *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition)* for more information.

2. Drag and drop the new physical columns into the Business Model and Mapping layer.

   See the "Working with Logical Tables, Joins, and Columns" chapter in the *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition)* for more information.

3. Drag and drop the new logical columns into the Presentation layer.

   See the "Creating and Maintaining the Presentation Layer" chapter in the *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition)* for more information.

Task: Adding New Fact or Dimension Tables

In some cases, you might want to add new fact or dimension tables to your Oracle BI repository. Possible sources include custom tables in Oracle Fusion Applications, additional tables in the data warehouse, or new physical data sources.

To add new fact or dimension tables to your RPD:

1. For Oracle Fusion Applications sources, create a custom BI view object for the custom table and incorporate it into the application as described in Task: Create BI View Objects for Custom Fact and Dimension Tables.

2. Use the Import Metadata Wizard in the Oracle BI Administration Tool to import the new view object (for Oracle Fusion Applications) or physical table (for warehouse or other physical sources) into the Physical layer of the RPD. For non-warehouse physical sources, you must create a new connection pool as part of the import process. You must connect as the FUSION_APPS_BI_APPID user in the Select Data Source screen of the Import Metadata Wizard.

For more information, see the following resources in the *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition)*:

- **Oracle Fusion Applications sources:** The "Importing Metadata from ADF Data Sources" section
- **Other physical sources:** The "Importing Metadata and Working with Data Sources" chapter

3. Define new logical dimensions and measures to extend the semantic model, and add physical and logical joins.

   See the "Working with Logical Tables, Joins, and Columns" and "Working with Physical Foreign Keys and Joins" sections in the *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition)* for more information.

4. Add corresponding Presentation layer metadata.

   See the "Creating and Maintaining the Presentation Layer" chapter in the *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition)* for more information.

Task: Changing How Metadata Is Displayed in Answers Reports

In some cases, you might want to change how the names of facts and dimensions in the Presentation layer appear in Answers reports, to comply with naming standards or for other reasons. Table 7–6 summarizes considerations for different use cases.

Table 7–6 Use Cases for Changing How Metadata Is Displayed

Use Case	For More Information
For warehouse sources, display names are typically externalized into a database table. To customize the names, you can change them in the externalized tables with no impact to the metadata itself. Note that for situations where display names are externalized into a database table, changing the names of Presentation layer objects in the RPD has no impact on the names displayed in Answers reports.	See the "Localizing Metadata Names in the Repository" section in the *Oracle Fusion Middleware System Administrator's Guide for Oracle Business Intelligence Enterprise Edition* for more information about externalizing display names.
For Oracle Transactional Business Intelligence sources, display names are typically customized using UI hints (labels and tooltips) within Oracle Fusion Applications. Changing the UI hint name does not impact metadata. Note that for situations where display names are customized using UI hints, changing the names of Presentation layer objects in the RPD has no impact on the names displayed in Answers reports.	See the "Propagating Labels and Tooltips from ADF Data Sources" section in the *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition)* for more information about how UI hints are propagated in the RPD.
For situations where Presentation layer names are not externalized or tied to UI hints, display names must be modified directly in the RPD. Existing reports will continue to work because the old names are stored as aliases.	See the "Renaming Presentation Columns to User-Friendly Names" section in the *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition)* for more information.

Task: Reorganizing Presentation Layer Metadata

Note the following about reorganizing Presentation layer metadata in the RPD:

- Reordering presentation columns within a presentation table will not cause existing reports to break. When subsequent patches are applied, the new custom order is preserved when the patch does not include changes to the column order for that table.
- Moving presentation columns across different presentation tables can cause existing reports to break and is not recommended. If you do move presentation columns across tables, it is important to plan ahead and track the changes carefully.

Task: Configuring Descriptive Flexfields and Key Flexfields for Oracle Business Intelligence

You can use the Import Metadata Wizard in the Administration Tool to incrementally import flexfield changes to the Physical layer of the Oracle BI repository (RPD).

> **Tip:** See Chapter 4, "Using Flexfields for Custom Attributes" for information about changing flexfields. In particular, see "Task: Incorporate Custom Attributes into Oracle Business Intelligence" for information about enabling flexfields for BI.

In addition, you can use the Map to Logical Model screen of the Import Metadata Wizard to automatically propagate the flexfield changes to the Business Model and Mapping layer.

Finally, for Oracle BI Applications customers, you can configure and enable the BI Extender functionality to propagate flexfield changes to the data warehouse.

See the following sections in the *Oracle Fusion Middleware Metadata Repository Builder's Guide for Oracle Business Intelligence Enterprise Edition (Oracle Fusion Applications Edition)* for more information about these topics:

- "Using Incremental Import to Propagate Flex Object Changes"
- "Automatically Mapping Flex Object Changes to the Logical Model"
- "Using the BI Extender to Propagate Flex Object Changes"

Task: Moving RPD Changes to Production Systems

Typically, data source connection pool settings are different in production repositories. You can use the Oracle BI Server XML API to programmatically update these settings in the repository when moving changes to production systems. See the "Moving from Test to Production Environments" section in the *Oracle Fusion Middleware XML Schema Reference for Oracle Business Intelligence Enterprise Edition* for more information.

Part III

Appendixes

This part contains information about troubleshooting Oracle Fusion Applications extensions and customizations. It contains the following appendix:

A

Troubleshooting Customizations

This appendix describes common problems that you might encounter when extending and customizing Oracle Fusion Applications and explains how to solve them.

> **Note:** Some features and functionality described in this chapter are not available in Oracle Cloud implementations.

This appendix contains the following topics:

- Section A.1, "Introduction to Troubleshooting Customizations"
- Section A.2, "Getting Started with Troubleshooting and Logging Basics for Customizations"
- Section A.3, "Resolving Common Problems"
- Section A.4, "Using My Oracle Support for Additional Troubleshooting Information"

In addition to this appendix, review *Oracle Fusion Middleware Error Messages Reference* for information about the error messages you may encounter.

A.1 Introduction to Troubleshooting Customizations

Use the following guidelines and process within this appendix to help focus and minimize the time you spend resolving problems.

Guidelines

When using the information in this appendix, consider the following guidelines:

- After performing any of the solution procedures in this appendix, immediately retry the failed task that led you to this troubleshooting information. If the task still fails when you retry it, perform a different solution procedure in this appendix and then try the failed task again. Repeat this process until you resolve the problem.
- Make notes about the solution procedures you perform, symptoms you see, and data you collect while troubleshooting. If you cannot resolve the problem using the information in this appendix and you must log a service request, the notes will expedite the process of solving the problem.

Process

Follow the process outlined in Table A–1 when using the information in this appendix. If the information in a particular section does not resolve your problem, proceed to the next step in this process.

Table A–1 Process for Using the Information in this Appendix

Step	Section to Use	Purpose
1	Section A.2	Get started troubleshooting customizations. The procedures in this section quickly address a wide variety of problems.
2	Section A.3	Perform problem-specific troubleshooting procedures for customizations. This section describes: ■ Common problems ■ Solution procedures corresponding to each of the possible problems
3	Section A.4	Use My Oracle Support to get additional troubleshooting information about Oracle Fusion Applications or Oracle Business Intelligence. My Oracle Support provides access to several useful troubleshooting resources, including Knowledge Base articles and Community Forums and Discussions.
4	Section A.4	Log a service request if the information in this appendix and My Oracle Support does not resolve your problem. You can log a service request using My Oracle Support at `https://support.oracle.com`.

A.2 Getting Started with Troubleshooting and Logging Basics for Customizations

This section provides the following general approaches for managing and diagnosing customization issues:

- Diagnosing Customization Issues Using the Manage Customizations Dialog
- Exporting Customizations
- Troubleshooting Customization Set Migration Issues
- Deleting Customizations
- Backing Up and Restoring Customizations
- Choosing the Right Customization Layer
- Determining the Full Path for a Customizations Document
- Logging Customizations that Are Applied to a Page
- Determining Whether a Page Has Customizations
- Using Sandboxes for Page Customizations
- Using Sandboxes for Flexfield Configurations
- Troubleshooting Flexfield Deployment
- Validating Flexfield Metadata

A.2.1 Diagnosing Customization Issues Using the Manage Customizations Dialog

The Manage Customizations dialog displays the customizations of the task flows in a page. You can access the Manage Customizations dialog from Page Composer and from the **Administration** menu in the global area of Oracle Fusion Applications. You can also use this dialog to delete page customizations, and to upload and download page customization files. For more information, see Section 2.1.1.1, "Viewing and Diagnosing Runtime Customizations."

A.2.2 Exporting Customizations

Customizations are stored in XML files. You can export the customizations for diagnosing issues in a number of ways, as described in Section 2.3, "Exporting and Moving Customizations."

A.2.3 Troubleshooting Customization Set Migration Issues

> **Note:** The functionality described in this section is not available in Oracle Cloud implementations.

The logging functionality for customization set migration is different from the standard logging functionality for Oracle Fusion Applications. The Customization Set Migration dialog uses Oracle Fusion Middleware logging functionality, which lets you adjust the amount of detail to log for customization set migration without requiring a server restart. You can use Oracle Enterprise Manager Fusion Applications Control to view the log messages. For more information about Oracle Fusion Middleware logging functionality, see the "Managing Log Files and Diagnostic Data" chapter in the *Oracle Fusion Middleware Administrator's Guide*.

To generate diagnostic information, select the desired level for the oracle.adf.share.customizationset logger as described in the "Adjusting ApplSession Log Levels for Troubleshooting" section in the *Oracle Fusion Applications Administrator's Guide* and rerun.

When requesting assistance from My Oracle Support, provide the log files and the customization set JAR file.

A.2.4 Deleting Customizations

You can use the Manage Customizations UI to delete customizations. In the page that contains the customizations, select the page fragment or task flow and choose **Manage Customizations** from the **Administration** menu. In the **Name** list, select the correct layer, find the page, task flow, or fragment that contains the customizations, and click **Delete** for the customization document that you want to delete.

> **Tip:** If you are unable to display the page that contains the customizations, choose **Manage Customizations** from the **Administration** menu, and then, in the Manage Customizations dialog, type the page, page fragment, or task flow in the **Search** text field, and click the **Search** icon.

If the customizations were made by someone other than the logged-in user, and you have administrator privileges, you can display that person's customizations by selecting **Select User** from the **Layer Name** dropdown list.

For more information about the Manage Customizations dialog, see Section 2.1.1.1, "Viewing and Diagnosing Runtime Customizations."

In non-Cloud implementations, you can also use Oracle WebLogic Scripting Tool (WLST) commands to delete customizations, as shown in Example A–1.

Example A–1 WLST Command to Delete a Customization Document

```
deleteMetadata (application='application name',
server='server name',
docs=
'/oracle/apps/hcm/dashboard/hrSpecialist/publicUi/page/mdssys/Site/SITE/VisaWorkPe
rmitExpirationRegion.jsff.xml');
```

For more information about the `deleteMetadata` command, see the "Application Metadata Management Commands" section of *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

Alternatively, in non-Cloud implementations you can use Fusion Applications Control to delete an application's metadata. For more information, see the "Managing the Metadata Repository" chapter in the *Oracle Fusion Middleware Administrator's Guide*.

A.2.5 Backing Up and Restoring Customizations

Note: The functionality described in this section is not available in Oracle Cloud implementations.

Before you make customizations, you can create a backup of a known good state by creating a label. If an issue occurs after creating the label, you can revert back to that label by promoting it to the tip as described in the "Creating Metadata Labels" and "Promoting Metadata Labels" sections of the *Oracle Fusion Middleware Administrator's Guide*.

Another way to back up and restore customizations is by exporting and importing customization files, as described in Section A.2.2, "Exporting Customizations."

A.2.6 Choosing the Right Customization Layer

When you make customizations, be careful to choose the correct layer:

- Use the site layer for customizations that affect all end users.
- Use the global layer for ADF Business Components customizations.
- Use product-specific layers appropriately as documented.

A.2.7 Determining the Full Path for a Customizations Document

The following string shows the structure of the full document path for a customization document:

```
/package/mdssys/cust/layer-name/layer-value/document-name.suffix.xml
```

For example, the full document path for the Visa Work Permit Expiration region is `/oracle/apps/hcm/dashboard/hrSpecialist/publicUI/page/mdssys/Site/SITE/VisaWorkPermitExpirationRegion.jsf.xml`.

You can obtain the full document path of a customized region on a page by completing the following steps:

1. Go to the page that contains the customized region and choose **Customize** *page_name* **Pages** from the **Administration** menu in the global area of Oracle Fusion Applications to open Page Composer.

2. If you have more than one layer available for customization, the Layer Picker dialog is displayed. In the **Edit** column, select the desired layer.
3. Choose **Source** from the **View** menu.
4. In the hierarchical list, drill down to and hover over the customized region to display the full document path of the JSF fragment that contains the customization, such as `/oracle/apps/hcm/dashboard/hrSpecialist/publicUI/page/mdssys/Site/SITE/VisaWorkPermitExpirationRegion.jsf.xml`. Make a note of this path.

For descriptive flexfield configurations, you can use the Register Descriptive Flexfields task to find the name of the flexfield's package.

A.2.8 Logging Customizations that Are Applied to a Page

To turn on runtime logging for customizations that are applied to a page, set the log level for the `oracle.mds.custmerge` module to `FINEST`. You can set the application's log level by choosing **Troubleshooting** from the **Help** menu. You might need to ask your administrator to give you privilege to set the log level.

If you have administration privileges, you can also use Fusion Applications Control to set the log level in non-Cloud implementations.

A.2.9 Determining Whether a Page Has Customizations

You can use logging, as described in Section A.2.8, "Logging Customizations that Are Applied to a Page," or you can use the Manage Customizations dialog, as described in Section A.2.1, "Diagnosing Customization Issues Using the Manage Customizations Dialog," to determine whether customizations have been applied to a page.

If you suspect that a problem might have been caused by a customization on a page, such as a user interface component disappearing from a page, you can export the page's customizations as described in Section A.2.2, "Exporting Customizations" and examine the document file.

A.2.10 Using Sandboxes for Page Customizations

Perform your page customizations in sandboxes and publish only after full testing. You can use any number of sandboxes for testing, but you should use just one sandbox for publishing to prevent merge conflicts, as described in Section 2.2.1, "Sandboxes and Concurrent Usage." For information about analyzing merge conflicts, see Section A.3.2, "Conflict Errors Reported When You Publish a Sandbox."

A.2.11 Using Sandboxes for Flexfield Configurations

Before deploying flexfield configurations to the full test environment, first deploy the configurations to a sandbox so that you can test the configuration without affecting other end users or risking the inclusion of the untested configuration in customization set movement to a production environment. A flexfield that is successfully deployed to a sandbox will have a status of Deployed to Sandbox. After deploying to a sandbox, log out and log back in. The flexfield's sandbox will be active in your session for you to test. After testing, deploy to the full test environment so that the changes can be seen by other users. For more information, see Section 4.7, "Deploying Flexfield Configurations."

A.2.12 Troubleshooting Flexfield Deployment

After you deploy a flexfield using the Manage Descriptive Flexfields task or the Manage Extensible Flexfields task, look at the value in the Deployment Status column to ensure that its status is Deployed (or Deployed to Sandbox if you are testing in a sandbox). If not, review the message in the Deployment Error Message column. For more information, see Section 4.7, "Deploying Flexfield Configurations."

If you successfully deployed a flexfield, but the custom attributes do not appear on the user interface page, ensure that you signed out and signed back in after deploying the flexfield. The changes do not appear until you sign back in.

In the case of an extensible flexfield, if a context does not appear in the user interface, verify that the context is associated with one of the category's pages.

To examine a flexfield's configuration, export the deployed artifacts using the `exportMetadata` WLST command as described in Section A.2.2, "Exporting Customizations."

A.2.13 Validating Flexfield Metadata

When you deploy a flexfield from the Manage Descriptive Flexfields task or the Manage Extensible Flexfields task, the metadata is validated and, if errors are found, the flexfield is not deployed. If you want to check whether a flexfield will pass validation, choose **Validate** from the **Actions** menu.

A.3 Resolving Common Problems

The following are common problems and solutions:

- User Interface is not Displaying the Active Sandbox Customizations
- Conflict Errors Reported When You Publish a Sandbox
- Dashboard Title Change Does Not Appear in Browser Title Bar, Navigator Link, or Tab
- Cannot Open Page After Personalizations
- Missing Navigator Menu Item
- Navigator Menu Item Does Not Work
- Nothing Changes After Clicking Cancel in Set Preferences Page

A.3.1 User Interface is not Displaying the Active Sandbox Customizations

Problem

The customizations that were made in the active sandbox are not appearing in the user interface.

Solution

Sign out and sign back in.

To ensure that the sandbox customization cache is cleared, log out and log back in before you enter a sandbox and after you perform any of the following sandbox-related actions:

- Exit a sandbox
- Publish a sandbox

- Delete a sandbox

A.3.2 Conflict Errors Reported When You Publish a Sandbox

Problem 1

You created customizations in a sandbox using the guidelines suggested in Section 2.2, "Using the Sandbox Manager." When you published the sandbox, merge conflicts were detected but not resolved.

Conflicts between sandboxes can arise when there is more than one sandbox that is intended for publishing in use. If two sandboxes contain customization changes to the same artifact and both are being published, the sandbox that is published last is given an option (by the sandbox manager) to overwrite the changes for that artifact from the sandbox that was published first. If the user working in the second sandbox decides to force-publish the second sandbox, the changes published by the first sandbox are overwritten. These types of conflicts can also occur with shared metadata files such as resource bundles that store translatable strings.

For more information, see Section 2.2.1, "Sandboxes and Concurrent Usage."

Solution 1

Before forcing the publishing of the sandbox, complete the following steps to analyze the conflicts:

1. Use the Manage Customizations dialog to download the customization document from the mainline code.
2. Make the sandbox active and use the Manage Customizations dialog to download the customization document from the sandbox.
3. Compare the two document files to analyze the conflicts.

Problem 2

When you publish a sandbox, you get a message that reports a conflict with `oracle/apps/menu/fnd/applcore/dataSecurity/dataSecurityService/mds/DSMO.xml`.

Solution 2

The security changes that you made in your sandbox conflict with other security changes in the mainline code. Cancel the sandbox and recreate the changes in a new sandbox.

Problem 3

When you publish a sandbox, you get a message that reports a conflict with `/oracle/apps/fnd/applcore/profiles/profileService/mds/ProfileMO.xml`.

Solution 3

Ignore the message, continue the process, and force the sandbox to be published.

A.3.3 Dashboard Title Change Does Not Appear in Browser Title Bar, Navigator Link, or Tab

Problem

After using Page Composer to change a dashboard page title, the old name is still displayed in the browser title bar, navigator link, or tab.

Solution

In addition to changing the page title property in the Task List Properties tab in Page Composer, which affects the browser title, you must also make the following changes:

- Task lists menu entry, page heading, and tab title: Change the label in the Task List Task Properties tab as described in "Task: Customize a Page Title" in Section 3.2, "Editing a Page in Page Composer."
- Navigator menu entry: Use the Manage Menu Customizations task to change the label for the menu item, as described in Chapter 5, "Customizing Menus."

A.3.4 Cannot Open Page After Personalizations

Problem

After making personalizations to a page, an end-user cannot open that page.

Solution

An administrator can use the Manage Customizations task to display and delete that user's personalizations for the page as described in Section A.2.4, "Deleting Customizations."

A.3.5 Missing Navigator Menu Item

Problem

An expected menu item is not appearing in the Navigator menu.

Solution

Verify whether the menu item has been hidden from view as described in Section 5.4, "Hiding and Showing Nodes."

A.3.6 Navigator Menu Item Does Not Work

Problem 1

A custom menu item was added and the browser is not displaying the page indicated by the URL.

Solution 1

Open the Manage Menu Customizations task and verify that the web application name is the same as the context root for the application, and that the view ID is the `id` attribute for the page's `<view>` tag in the product's `public_html/WEB-INF/adfc-config.xml` file. The URL should not contain the `.JSPX` suffix.

For more information, see Section 5.3, "Adding Menu Items."

Problem 2

When you choose an item in the **Navigator** menu, you get a "webApp *value* not defined" error message.

Solution 2

Verify that the application is in the topology tables, as described in the "Viewing the Routing Topology of an Oracle Fusion Applications Instance, Product Family, or Product" section in the *Oracle Fusion Applications Administrator's Guide*.

A.3.7 Nothing Changes After Clicking Cancel in Set Preferences Page

Problem

You click the Cancel button in the Set Preferences page and nothing changes.

Solution

The cancel action resets the page to the same values that appeared when you accessed the page. If you have not made any changes, the page appears the same. To navigate away from the page, select **Recent Items** or **Navigator** from the global area of Oracle Fusion Applications or use the text box to search for a different page.

A.4 Using My Oracle Support for Additional Troubleshooting Information

You can use My Oracle Support (formerly MetaLink) to help resolve Oracle Fusion Middleware problems. My Oracle Support contains several useful troubleshooting resources, such as:

- Knowledge base articles
- Community forums and discussions
- Patches and upgrades
- Certification information

Note: You can also use My Oracle Support to log a service request.

You can access My Oracle Support at `https://support.oracle.com`.

Glossary

application role

A **role** specific to applications and stored in the policy store.

application stripe

A collection of Oracle Application Server Java Authentication and Authorization Service (JAAS) Provider policies that are applicable to the application with which it is associated. Out of the box, an application stripe maps to a Java EE application. Oracle Platform Security Services also supports mapping multiple Java EE applications to one application stripe.

artifact (SAR file)

A file included in the **SAR file** of the **SOA composite application**. Examples of artifacts include **binding components** and **service components**, references to B2B agreements, Oracle Web Services Manager (Oracle WSM) policies, **human task** flows, and metadata such as WSDL and XSD files.

binding component

A component that establishes the connection between a **SOA composite application** and the external world. There are two types of binding components:

- services: provide the outside world with an entry point to the SOA composite application
- references: enable messages to be sent from the SOA composite application to the external services in the outside world.

BPEL

Business Process Execution Language. An XML-based markup language for composing a set of discrete web services into an end-to-end process flow.

BPEL process

A **service component** that integrates a series of business activities and services into an end-to-end business process flow. See also **BPEL**.

bucketset

A container for defining a list of values or a range of values of a specified type. After you create a bucketset, you can associate the bucketset with a fact property of matching type. **Business rules** use the bucketsets that you define to specify constraints on the values associated with fact properties in rules or in **decision tables**.

business event

A message sent as the result of an occurrence or situation, such as a new order or completion of an order. You can raise business events when a situation of interest occurs. When an event is published, other applications can subscribe to it. Definitions for business events are stored in an Oracle Metadata Services (MDS) repository, and then published in the Event Delivery Network (EDN).

business object

A resource in an enterprise database, such as an invoice or purchase order.

Business Process Execution Language (BPEL)

See **BPEL**.

business rule

A statement that describes business policies or describes key business decisions.

Child Components

The components contained within a parent **component**. For example, the task flows contained within a Box layout component are the child components of the Box.

component

An individual piece of an application, for example, a task flow, portlet, page, or layout element such as a box or image.

configuration plan

As you move projects from one environment to another (for example, from testing to production), you typically must modify several environment-specific values, such as JDBC connection strings, hostnames of various servers, and so on. Configuration plans enable you to modify these values using a single text (XML) file. During process deployment, the configuration plan searches the SOA project for values that must be replaced to adapt the project to the next target environment.

custom object

A high-level artifact, which manages data that resides in a database table, that you create using Oracle Fusion CRM Application Composer.

customize

To change a standard (existing) Oracle Fusion Applications artifact.

data dimension

A stripe of data accessed by a data **role**, such as the data controlled by a business unit.

data security

The control of access to data. Data security controls what action an end user can take against which data.

data stripe

A dimensional subset of the data granted by a **data security** policy and associated with a data **role**. The data dimension expresses stripes of data, such as territorial or geographic information, that you can use to partition enterprise data.

decision table

An alternative business rule format that is more compact and intuitive when many rules are needed to analyze many combinations of property values. You can use a decision table to create a set of rules that covers all combinations or when no two combinations conflict.

dependent value set

A list of values whose availability and meaning depend on the value that the end user provides for a prior **segment**, where the prior segment is associated with an **independent value set**.

descriptive flexfield

A type of **flexfield** used to give additional attributes to a data model. A descriptive flexfield can support only a set amount of **segments**.

design time customizations and extensions

Customizations and extensions that include more complex changes, such as creating Oracle SOA Suite composite applications or creating new batch jobs. Design time customizations are most often done by Java developers using Oracle JDeveloper (a comprehensive IDE), or may be done in other tools, such as Oracle SOA Composer. The customizations are then uploaded or deployed to a running instance of Oracle Fusion Applications.

domain value map

A set of value mappings that enables you to associate values from one application with values from another. For example, one value can represent a city with a long name (Boston), while another value can represent a city with a short name (BO). In such cases, you can directly map the values by using domain value maps.

desktop page

A page in a user interface that is optimized for extended periods of use with monitors.

entitlement

A set of grants of access to functions and data. This is an Oracle Fusion Middleware term for **privilege**.

extend

To create a completely new artifact, such as a custom **business object** or custom view page.

extensible flexfield

A type of **flexfield** that is similar to a **descriptive flexfield**, but does not have a fixed number of **segments**, allows grouping of segments into contexts, allows entities to inherit segments from their parents, and supports one-to-many relationships between an entity and its extended attribute rows.

flexfield

A set of placeholder fields (**segments**) that is associated with a business object. Oracle Fusion Applications provides three types of flexfields: descriptive, extensible, and key. Implementors use descriptive and extensible flexfields to add custom attributes to **business objects**. Implementors use key flexfields to define keys, such as part numbers.

flexfield sandbox

A **sandbox** to which you deploy **flexfield** configurations for testing purposes before deploying to the mainline code.

format-only value set

A **value set** that conforms to formatting rules. This is used when you want to allow end users to enter any value so long as that value conforms to formatting rules. For example, if you specify a maximum length of 3 and numeric-only, then users can enter 456 but not 4567 or 45A. Use a format-only value set only when no other types of validation are required.

function security

The mechanism by which user access to application functionality is controlled.

global layer

A customization layer in which customizations affect all end users of the application. This layer's XML files are added for everyone, whenever the artifact is requested. Customizations made to ADF Business Components in Oracle JDeveloper must be made in the global layer.

human task

A Business Catalog component that enables you to define how end users interact with your Oracle Business Process Management (Oracle BPM) processes. Human tasks are implemented in an Oracle BPM process using the user task. Human tasks are also used in **SOA composite applications**, where they are known as **service components**.

independent value set

A predefined list of values for a **flexfield segment**. These values can have an associated description. The meaning of a value in this **value set** does not depend on the value of any other segment.

key flexfield

A non-optional type of **flexfield**. This type of flexfield is used to define the parts of a key structure such as the parts of a product key or the parts of a customer key.

metadata sandbox

The type of **sandbox** that supports making changes to the application's metadata stored in an Oracle Metadata Services (MDS) repository.

multi-tenant environment

An environment where a single application instance serves multiple client organizations by partitioning the data and configurations into separate virtual application instances.

navigator menu

The global menu that is accessible from the Oracle Fusion Applications global area.

page fragment file

A particular type of file that represents only a fragment of a page, but that is necessary to properly display the entirety of that page.

partner link

Characterizes the conversational relationship between two services in a **BPEL process** by defining the roles played by each service in the conversation and specifying the port type provided by each service to receive messages within the conversation.

permission

A security artifact in the policy store that maps a specific application resource with an allowed action. For example, a permission may be granted to a **role** to confer access rights to users.

personalization

The changes that every end user of the Oracle Fusion Applications product suite can make to certain artifacts in the user interface (UI) at runtime. These changes remain for that user each time that user logs in to the application. Personalization includes changes based on user behavior (such as changing the width of a column in a table), changes the user elects to save, such as search parameters, or composer-based personalizations, where an end user can redesign aspects of a page.

privilege

A grant of access to functions and data. A privilege is a defined by a single, real world action on a single business object.

production-to-test movement

The process of completely refreshing the test environment by copying configuration and data from the production system.

resource bundle

A collection of locale-specific objects. When a program needs a locale-specific resource, a `String` for example, it can load the resource from the resource bundle appropriate for the current user's locale. In this way, the program code is largely independent of the user's locale, isolating most, if not all, of the locale-specific information in resource bundles.

role

An identity that determines permitted access to application functions and data.

rule dictionary

A **business rules** container for facts, functions, globals, **bucketsets**, links, decision functions, and **rulesets**. A dictionary is an XML file that stores the application's rulesets and the data model. Dictionaries can link to other dictionaries.

ruleset

A **business rules** container for rules and **decision tables**. A ruleset provides a namespace, similar to a Java package, for rules and decision tables.

runtime customizations and extensions

Customizations and extensions that can be made to Oracle Fusion Applications at runtime using browser-based components. These customizations and extensions are available to all or to a subset of Oracle Fusion Applications end users, and range from changing the look and feel of a page, to customizing standard **business objects**, adding a new business object and associated pages and application functionality, changing workflows, defining security for new objects, and customizing reports.

sandbox

A testing environment that separates sections of an application so that changes and modifications are kept within the sandbox and do not affect the mainline code or other sandboxes. Different users can create their own sandboxes to test their own sections. After the changes made in the sandbox have been tested, the sandbox user has the option to publish the changes to the mainline code.

SAR file

A SOA archive deployment unit. A SAR file is a special JAR file that requires a prefix of sca_. (For example, sca_OrderBookingComposite_rev1.0.jar). The SAR file packages **binding components** and **service components**, such as **BPEL processes**, **business rules**, **human tasks**, and mediator routing services, into a **SOA composite application**.

security reference implementation

Provides **role**-based access control in Oracle Fusion Applications, and is composed of predefined security policies that protect functions, data, and segregation of rules. The reference implementation supports identity management, access provisioning, and security enforcement across the tools, data transformations, access methods, and the information lifecycle of an enterprise.

security sandbox

The type of **sandbox** that supports making **data security** changes.

segment

A subdivision of a **flexfield**. A segment captures a single atomic value, which is represented in the application database as a single column. In the application UI, segments can be presented as individual table columns, as separate fields, or as a concatenated string of values.

service component

A component that implements the business logic or processing rules of a **SOA composite application**. Service components include **Oracle Business Process Execution Language processes**, **business rules**, **human tasks**, and mediator routing services.

simplified page

A page in a user interface that is optimized to provide quick access to high-volume, self-service tasks from any device.

site layer

A customization layer in which customizations affect end users at a particular location.

skin

A style sheet based on the CSS 3.0 syntax and that is specified in one place for an entire application. Instead of providing a style sheet for each component, or inserting a style sheet on each page, you can create one skin for the entire application.

SOA composite application

An assembly of service **binding components**, **service components**, and reference binding components designed and deployed in a single application. Wiring between the components enables message communication. The details for a composite are stored in the composite.xml file.

subset value set

A set of values taken from an existing **independent value set**. For example, if you have an independent **value set** for the days of the week, a weekend subset can be composed of its entries for Saturday and Sunday.

table value set

A **value set** with the Table validation type. The valid values in the value set are obtained from a specified column in a database table.

UI Shell template

The template that is use for the base UI for all Oracle Fusion Applications pages.

user layer

The customization layer in which all **personalizations** are made. The user layer is automatically selected when you use the Personalization menu.

value set

A list of values used to specify the validation rules for a **flexfield segment**.

wire

Wires connect service **binding components**, **service components**, and reference binding components into a complete **SOA composite application**.