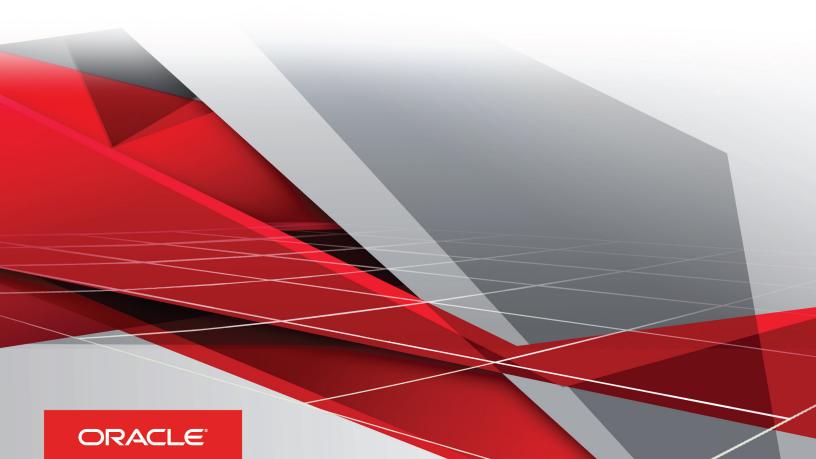
Oracle

Applications Cloud Customizing the Applications for Functional Administrators

Release 12

This guide also applies to on-premises implementations



Oracle® Applications Cloud Customizing the Applications for Functional Administrators

Part Number E73426-02

Copyright © 2011-2017, Oracle and/or its affiliates. All rights reserved.

Authors: Essan Ni Jirman, Barnali Roy, P.S.G.V. Sekhar

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

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, then 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.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

The business names used in this documentation are fictitious, and are not intended to identify any real companies currently or previously in existence.

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.

Oracle customers that have purchased support 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.

Contents

	Preface	
1	Overview	-
_	Customizing and Extending Applications: Overview	
	Run Time Customizations and Extensions	
	Design Time Customizations and Extensions: Highlights	(
	Personalization	(
	Customization Layers	
	Business Process Models: Explained	1
2	Customization Life Cycle	13
	Customization Life Cycle: Explained	1;
	Run Time Customizations	14
	Sandbox Manager	1
	Moving Customizations	23
	Viewing and Deleting Customizations: Procedure	29
	Advanced Customization Life Cycle Tasks: Highlights	30
3	Page Customization	3-
	Customizing Pages: Overview	3
	Page Composer Overview	3
	Page Content and Template Customization	3
	User Interface Text Customization	5
	Theme Customization	6
	Infolet Customization	6
	Design Time Page Customization	7



4 Flexfiel	ds for Custom Attributes	73
Flexfields: (Overview	73
Configuring	Flexfields: Overview	73
Flexfield Co	emponents: Explained	75
Flexfields a	t Run Time: Explained	76
Customizin	g Flexfields Using Page Composer: Explained	76
Flexfields a	nd Oracle Applications Cloud Architecture: How They Work Together	77
Flexfields a	nd Value Sets: Highlights	79
Flexfield Ma	anagement	80
Flexfield De	ployment	94
Manage Va	lue Sets	103
Manage De	escriptive Flexfields	118
Manage Ex	tensible Flexfields	126
Manage Ke	ey Flexfields	141
	tor and Springboard Customization g the Navigator and Springboard: Overview	159
	g the Navigator and Springboard	160
	ategories and Page Entries for Navigation: Procedure	160
•	Links for Page Entries: Procedure	162
Managing (Categories and Page Entries for the Navigator and Springboard: Procedure	164
Defining Se	ttings for Home and Navigation: Explained	165
FAQs for N	avigator and Springboard Customization	166
6 Help Cu	ustomization	167
Customizin	g Help: Overview	167
	dd and manage custom help?	167
	ustomization	168
•	ion of Getting Started Pages	180
	Help Customization	181



Preface

This preface introduces information sources that can help you use the application.

Oracle Applications Help

Use the help icon (?) to access Oracle Applications Help in the application. If you don't see any help icons on your page, click the Show Help icon (?) in the global header. Not all pages have help icons. You can also access Oracle Applications Help at https://fusionhelp.oracle.com.

Using Applications Help

Watch: This video tutorial shows you how to find help and use help features.

Additional Resources

- Community: Use Oracle Applications Customer Connect to get information from experts at Oracle, the partner community, and other users.
- Guides and Videos: Go to the Oracle Help Center to find guides and videos.
- Training: Take courses on Oracle Cloud from Oracle University.

Documentation Accessibility

For information about Oracle's commitment to accessibility, see the Oracle Accessibility Program .

Comments and Suggestions

Please give us feedback about Oracle Applications Help and guides! You can send e-mail to: oracle_fusion_applications_help_ww_grp@oracle.com.





1 Overview

Customizing and Extending Applications: Overview

While your application provides robust functionality as delivered, you can make changes to it, if necessary.

You can:

- **Customize:** Change a standard (existing) artifact. For example, you can add an attribute to an existing business object or change what's displayed on a standard page.
- Extend: Create a new artifact, such as a new object.

The three basic scenarios for customizations and extensions are:

- Run time customizations and extensions
- Design time customizations and extensions
- Personalization

What You Can Change

You can customize and extend many aspects of the application, for example the user interface, business intelligence, and data model.

The application is built using a common data model. So, 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
- Associated mobile page
- Associated reports

Generally, you use the same tools and processes to customize all applications. For more information on customizing business intelligence, see the Creating and Editing Analytics and Reports guides relevant to your products.

Related Topics

Customization Life Cycle: Explained

Run Time Customizations and Extensions

Run Time Customizations and Extensions: Examples

Run time customizations and extensions include changes that you can make to the application at run time using browser-based composers and other tools. All users or a subset of users can view and use these customizations and extensions. If



your role has an administrative privilege, you can access most run time customization tools to customize the user interface (UI), create and customize objects, and so on. Some customization tools, such as Application Composer, are available only for specific product families.

Customizing the UI

To customize the UI, use:

- The Customize User Interface Text page to edit text that appears on multiple pages. For example, you can change
 the term, supplier to vendor if that is your preferred term, and the change affects all pages where the term is
 displayed.
- The Settings work area to change the:
 - Look and feel of simplified UI
 - Announcements on the home page
- Page Composer to customize simplified and desktop pages for other users. For example, you can:
 - Add fields
 - Add validation
 - Change defaults
 - Rearrange regions
 - Add external content
 - Save queries
- Tip: In Page Composer, you can make changes using the WYSIWYG view. However, in some cases, you can also use the Source view.

Customizing Navigation

Use the Structure page to customize the Navigator and springboard. From the Navigator menu, select **Structure** under **Tools**.

Adding Custom Attributes to Business Components Using Flexfields

Most business components, except those in Oracle Sales Cloud products, support using flexfields to extend the object with custom attributes. Use a flexfield to create custom attributes without programming. The flexfield captures data that is related to a specific purpose, such as information about job positions or inventory items. Each attribute is a segment of the flexfield, and corresponds to a reserved column in the application database.

Customizing Reports and Analytics

Predefined analyses, dashboards, and reports help in meeting business intelligence requirements. But you can customize them (for example, change the layout) to fit specific business needs.

For more information, see the Creating Analytics and Reports guides relevant to your products.

Customizing Help

Use the Manage Custom Help page to:

- Customize the help files in the application Help
- · Determine which help files to show in specific help windows



You can open the Manage Custom Help page from any help window, or from the help site.

Note: You must have the appropriate job roles to customize help.

Related Topics

Help File Customization: Overview

• Flexfields: Overview

Tools for Customizations and Extensions: Critical Choices

You can customize and extend your application to suit your business needs.

Choose an appropriate tool based on the types of customizations and extensions to make, such as:

- Page customization
- Branding customization
- Object customization
- Security customization
- Business intelligence customization
- Help customization

Note:

- The following tables present only the top customization tasks, not all tasks.
- The tables don't list design time customizations and extensions that usually developers perform. These customizations and extensions aren't available in Oracle Cloud implementations.

Page Customization

This table shows some types of customizations that you can make to pages, and the corresponding tools to use. You can customize only certain pages in Page Composer. If the customization that you want to make isn't available in Page Composer, then developers can customize the pages using Oracle JDeveloper (not available in Oracle Cloud implementations).

Customization	Tool
Add, move, delete, show, or hide components on a page	Page Composer
Change a page layout	Page Composer
Create a site-level search for all users	Page Composer
Customize a page title	Page Composer
Customize a task list menu	Page Composer



Customization	Tool
Customize dialog box content	Page Composer
Customize attributes for a flexfield on a page	Page Composer
Customize properties for user interface (UI) components on a standard page	Page Composer
Customize the UI Shell template	Page Composer
Customize a text string wherever it appears across all pages	User Interface Text
Customize the look and feel of application pages	Appearance page
Change the announcements on the home page	Announcements page

Branding Customization

This table shows some types of customizations that you can make to use your own branding logo, and the corresponding tools to use.

Customization	Tool
Customize the UI Shell template	Page Composer
Change the logo and application name in the UI	Appearance page
Customize report layouts	Layout editor in the BI application or external applications such as Microsoft Word

Object Customization

This table shows some types of customizations that you can make to objects, and the corresponding tools to use.

Customization	Tool
Add an attribute to a business object using flexfields (not Oracle Sales Cloud)	Setup and Maintenance work area
Add a business object page to the Navigator menu	Setup and Maintenance work area



Security Customization

This table shows a security customization that you can make to objects, and the corresponding tool to use.

Customization	Tool
Add data security to a custom object.	Setup and Maintenance work area

Business Intelligence Customization

This table shows some types of customizations that you can make to business intelligence (BI) analytics and reports, and the corresponding tools to use. For more information, see the Creating and Editing Analytics and Reports guides relevant to your products.

Customization	Tool
Create report layout	Layout editor in the BI application or external applications such as Microsoft Word
Customize report layouts	Layout editor in the BI application or external applications such as Microsoft Word
Create a report	The BI application
Customize analyses	Reports and Analytics pane or the BI application
Customize dashboards	The BI application

Help Customization

This table shows some types of customizations that you can make to help, and the corresponding tools to use.

Customization	Tool
Customize text that is displayed when the user hovers over a button, link, icon button, or tab title	Page Composer
Customize help files and determine the help links to show on help windows	Oracle Fusion Applications Help
Simultaneously replace multiple occurrences of a word or phrase that appear in the embedded help	User Interface Text

Oracle Sales Cloud has additional tools available for customizations. For example, Application Composer and Business Process Composer.

Related Topics

• Flexfields: Overview



• Help File Customization: Overview

Design Time Customizations and Extensions: Highlights

Developers can customize and extend applications using design time tools, for example, Oracle JDeveloper (a comprehensive integrated development environment). Design timecustomizations and extensions include complex changes that developers need to deploy into the run time environment.

Note: These customizations and extensions aren't available in Oracle Cloud implementations.

Design Time Tasks

• To do design time customizations, use Oracle JDeveloper or other tools such as Oracle SOA Composer. Refer to the Oracle Fusion Applications Extensibility Guide for Developers.

See: Design Time Customizations and Extensions

You can determine what users can personalize.

See: Configuring End-User Personalization

You can also create a complete Java EE application and integrate it with existing applications.

See: Oracle Fusion Applications Developer's Guide

Personalization

Personalization: Explained

Personalization refers to the changes that every user of the application can make to certain artifacts in the user interface (UI) at run time. To personalize the UI, click your user name in the global area and use the **Personalization** menu items.

Note: Personalization changes remain for a user each time the user signs in to the application.

Personalization includes:

- Changes based on how you use the UI, such as changing the width of a column in a table
- Changes that you select to save, such as search parameters
- Composer-based personalizations, where you can redesign aspects of a page. For example, you can use Page Composer to rearrange regions on a page and add or remove content



FAQs for Personalization

How can I retain minor personalizations after I sign out?

Whether minor personalizations (such as changing table layout and hiding regions) persist after you sign out, depends on the page you're on. However, to retain all minor personalizations, make changes by clicking your user name in the global area and selecting **Edit Current Page** under **Personalization** (if available).

Related Topics

Saving Searches for Searches with Multiple Criteria: Procedure

How can I restore a page to the default content and layout?

Click your user name in the global area and select **Reset to Default Content and Layout** (if available for the page you're on) under **Personalization**.

What happens if I restore a page to the default content and layout?

You remove all changes that you had previously made to the page using **Edit Current Page** (under **Personalization**, which you open by clicking your user name in the global area). The reset affects objects at the page level, for example, dashboard regions that you added or moved around, and dashboard layout changes. Restoring doesn't reset personalization changes made within the components of a page, for example, within a dashboard region.

Customization Layers

Customization Layers: Explained

Use the built-in customization layers in your application to make customizations that affect only certain instances or users of an application. Before you create customizations, select the layer in which you want to customize. Most of the customization tools provide a dialog box for selecting the layer for your customizations. If the dialog box to select a layer doesn't appear before you customize, then by default your customizations are made to the site layer.

Available Layers

The customization layers available to an application depend on its application family. However, all applications have the following customization layers:

- Site layer: Customizations made in the site layer affect all users.
- User layer: All personalizations are made in the user layer. Users don't have to explicitly select this layer as it's automatically applied while personalizing the application.

Layer Hierarchy

The layers are applied in a hierarchy, and the highest layer in that hierarchy in the current context is considered the tip layer. With the default customization layers, the user layer is the tip layer. An object may be customized more than once, but in different layers. At run time, the tip-layer customizations take precedence. For example, say you customize in the site layer. You use Page Composer to add a region on a page. A user personalizes the same page to hide the region. In such a case, the user-layer customization takes precedence for that user at run time.



Storage of Customizations and Layer Information

Customizations aren't saved to the base standard artifact. Instead, they're saved in Extensible Markup Language (XML) files for each layer. These files are stored in an Oracle Metadata Services repository. The XML file acts like a list of instructions that determines how the artifact looks or acts in the application, based on the customization layer. The customization engine in Oracle Metadata Services manages this process.

When you apply an application patch or upgrade, it updates the base artifacts, but it doesn't touch the customizations stored in XML files. The base artifact is replaced. Hence, when you run the application after the patch or upgrade, the XML files are layered on top of the new version. You don't have to redo your customizations.

Working With Customization Layers: Examples

The following scenarios illustrate working with customization layers to ensure that the correct customizations or personalizations are available at run time to appropriate users. 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 with a specific role, for example, a sales representative.

Customizing Pages for Users with Specific Roles

You want to remove the Quick Create panel from the Sales home page, and customize this page only for users with the Sales Representative role.

Following are the prerequisites:

- Activate a sandbox.
- When you customize a page for a specific job role, that role must be assigned to you for you to test the
 customization in the sandbox. Your security administrator can either assign the role to you directly, or make the role
 self requestable for you to add it from the resource directory.
- Select the layer in which you want to make your customization. In this case, select the role layer with the value, Sales Representative. While customizing, when you remove the panel from the page, an XML file is generated. This file contains instructions to remove the panel, but only for the role layer, and only when the value is Sales Representative.
- Note: The original page file remains untouched.

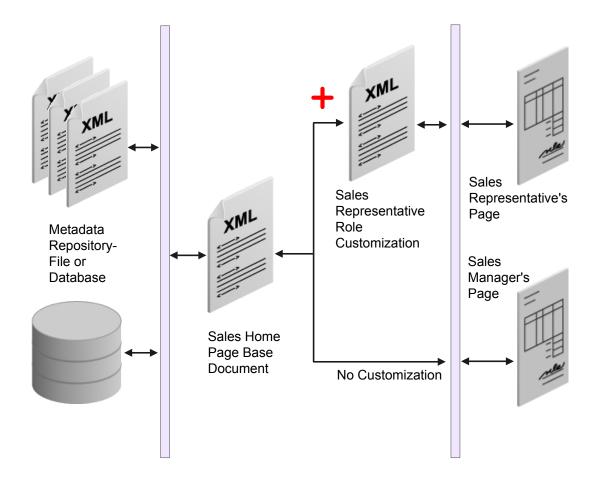
The customization engine in Oracle Metadata Services then stores the XML file in the Oracle Metadata Services repository. When someone signs in and requests an artifact, the customization engine in the metadata service checks the repository for XML files matching the artifact and the given context. On matching, the customization engine layers the instructions on top of the base artifact.

In this example, whenever someone:

- With the role of Sales Representative (the context) requests the Sales home page (the artifact), before the page is rendered, the customization engine in Oracle Metadata Services:
 - Pulls the corresponding XML file from the repository
 - Layers it on top of the standard Sales home page
 - Removes the panel
- Without the role of Sales Representative signs in, the customization engine doesn't layer the XML file on top of the standard Sales home page. So, the Quick Create panel is displayed on the page.



This figure shows how the customization XML file is applied to the base document and is displayed only for a sales representative.



Personalization

All users of the application can use the Personalization menu items to personalize certain pages. For example, you can:

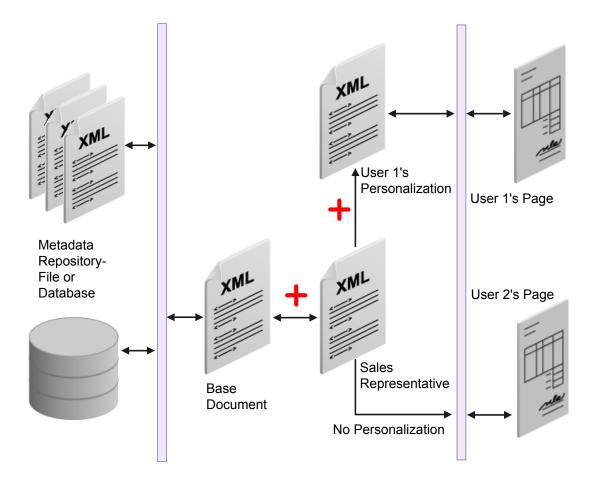
- Move elements around on a page
- Hide elements
- Add available elements to a page

While you personalize a page, the customization engine in Oracle Metadata Services creates an XML file specific to a user (in this case, you), for the user layer. For example, say User 1 (with the role of Sales Representative) personalizes the Sales home page. An XML file, noting the changes that the user made, is stored in the repository. When User 1 signs in, the customization engine in the metadata services:

- Pulls the XML file with the sales representative customizations from the repository, and layers the file on top of the standard Sales home page
- Pulls the XML file with the User 1 personalizations, thus enabling the user to see the personalization changes along with the Sales Representative changes.



When other sales representatives sign in, they don't see the User 1 personalization changes, as shown in this figure.



Related Topics

- Role Provisioning and Deprovisioning: Explained
- Autoprovisioning: Explained
- How do I provision roles to users?

Selecting Customization Layers to Include: Examples

While customizing, when you use the dialog box to select the customization layer, you can also include lower layers to view customizations from those layers.

The following scenarios explain what happens based on your selected layers. For these examples, the available layers are Site, Territory, and Job Role.

What You See While Customizing

Suppose you choose to:

Edit the Job Role layer and select Sales Representative as the value for that layer



- Include the Territory layer and select Southwest as the value
- Note: The Site layer is automatically included because it applies to everyone.

While customizing in Page Composer, you see customizations that apply to sales representatives in the Southwest territory, based on:

- What was defined for each layer
- Which is the highest layer with customization for a specific artifact

What Your Customizations Apply To

No matter what you see while customizing, the customizations you make apply only to the edit layer, Job Role. For example, say a field is hidden in the Site layer but displayed in the Territory layer for Southwest. No customization exists for the field in the Job Role layer for sales representative. Since Territory is higher than Site, you see the field displayed while customizing in Page Composer. However, suppose you choose to hide the field as part of your customization. So, that customization applies to the Job Role layer for sales representatives. Users with other job roles in the Southwest territory may still see the field. However, Job Role is higher than Territory. So, no sales representatives in any territory can see the field, unless a layer higher than Job Role applies to any of these users and has the field displayed.

Business Process Models: Explained

The application is based on business process models that map out business flows. When you customize and extend your application, for example to add new pages, you can use these models to help you plan. For diagrams of business process models, see Oracle Fusion Business Process Models (1542019.1) on My Oracle Support at https://support.oracle.com.

Business Process Modeling Levels

The business flows are presented in a five-level hierarchy: industry (L0), business process area (L1), business process (L2), activity (L3), and tasks (L4).

- The hierarchy goes from a high-level, conceptual view to a low-level, application-specific view.
- L1 through L3 are business-driven and don't depend on any specific implementation in the application.
- L4 aligns with specific features and functionality in the application.

In the Application

The application is organized around these hierarchy levels and flows, which puts focus on the activities and tasks that you must perform. Several aspects of the application are influenced by, if not directly based on, the business process modeling levels. For example, the navigation, user interface, and parts of security are influenced by the business process modeling levels.





2 Customization Life Cycle

Customization Life Cycle: Explained

All customizations and extensions to your application must be done in a full test environment. Typically, this environment contains one or more applications that will be moved to a production environment after you complete and test all customizations and extensions.

While using tools such as Page Composer, you can make application customizations in a sandbox. Sandboxes store customizations in Extensible Markup Language files in a separate Oracle Metadata Services repository that's available only when you work in that particular sandbox. You can make the changes in either of the following:

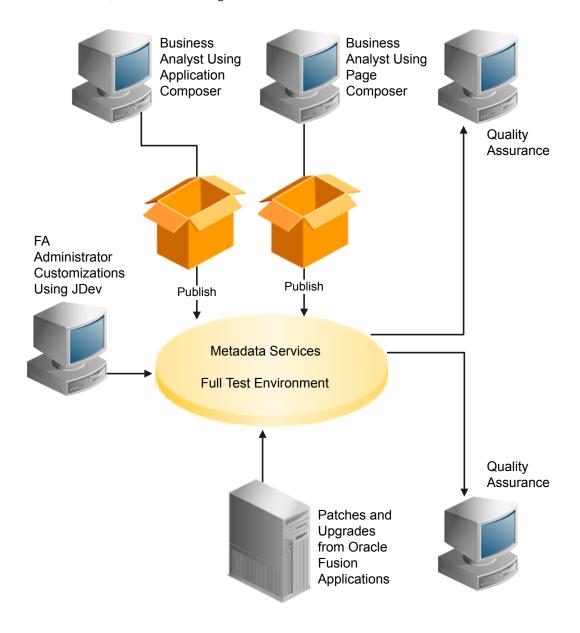
- A test-only sandbox (that is, the code in the sandbox is for testing only, and is never deployed)
- A sandbox that's then published to the full test environment

Developers using design time tools, such as Oracle JDeveloper, can deploy their customizations directly to that environment, or they can publish to a sandbox. For more information on design time customization workflow, see the Oracle Fusion Applications Extensibility Guide for Developers.

Note: Design time customizations aren't available in Oracle Cloud implementations.



Project managers can monitor, import, and export customizations. Then, users can test the entire environment with all customizations, as shown in the figure below.



Tip: For on-premise implementations, developers may allow users, having access to the Setup and Maintenance work area, to configure the customizations and extensions made to the application.

Run Time Customizations



Run Time Customization Workflow: Explained

While using Application Composer and Page Composer to make run time customizations to your application, use sandboxes to save your changes in a segregated environment. For example, before making customizations, suppose you create a sandbox named Mysandbox, and then make your customizations in that sandbox. Now, if others want to see your customizations, they can use Mysandbox.

Note: If you have multiple users working on the same sandbox, then conflicts may arise within a sandbox. Hence, users must adhere to the prescribed guidelines to avoid such conflicts.

You can also use a sandbox while defining 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 approve your customizations, and then publish to the full test environment.

Note that a flexfield sandbox is for testing only and can't be published. Instead, you can 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. The changes that you deploy to a sandbox are isolated from the full test environment. Users who make the flexfield sandbox active in their session can only see these changes. After you're satisfied with the changes in the sandbox, you can deploy the changes to the full test environment.

You can also use the Manage Customizations dialog box to:

- View others' customization metadata files
- Download others' customization metadata files for manually moving them to another environment or diagnosing any issues

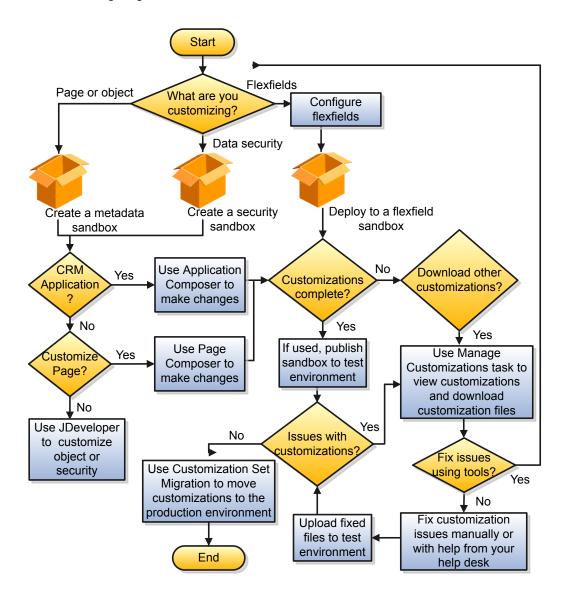
You can also upload others' customization metadata files to your environment.

This figure illustrates the use of sandboxes while:

Customizing pages, objects, and security using Page Composer and Application Composer



Configuring flexfields



Viewing and Diagnosing Run Time Customizations: Points to Consider

Use the Manage Customizations dialog box to view and diagnose run time customizations made to application pages. Customizations are role-dependent and by default, the Manage Customizations dialog box displays the customizations that the signed-in user had performed.

Before you begin viewing customizations, ensure that you have administrative privileges to access the Manage Customizations dialog box. If you're unable to display the page that contains the customizations:

1. Click your user name in the global area, and select **Manage Customizations** from the **Administration** menu.



Use the Search text field on the Manage Customizations dialog box to search for the page, page fragment, or task flow.

You can view the customizations for a user under the Current Context column on the Manage Customizations dialog box. On this dialog box, you can change the page, page fragment, or task flow for which you're viewing customizations using the **Search** field.

Developers too may be assigned to specific roles and can view only those customizations that are permitted for the specific roles. However, administrators can view all customizations made at the site level, and for any user, under the All Layers column on the Manage Customizations page. To view customizations made by more than one user, administrators can select multiple users.

Sometimes, an administrator may need to view a personalization that was made by another end user. For example, suppose a user made an error while personalizing a page and that page is no longer displayed for the user. Because the user can't open the page, the user can't 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.

Page-Level Customizations

To diagnose customization issues, determine whether customizations have been applied to a page. Use the Manage Customizations dialog box to determine if page-level customizations exist. If a page customization causes problems, such as a user interface component disappears from a page, you can export the customizations and examine the document file.

Related Topics

Customization Layers: Explained

Sandbox Manager

Managing Customizations Using Sandboxes: Explained

You can apply different types of customizations to an application. For example, you can apply changes to an application's metadata stored in the metadata services repository or changes related to data security of the application. All such customizations are stored in sandboxes and are validated before applying them to an application.

Types of Customizations in Sandboxes

Sandboxes can contain the following types of customizations:

- Metadata changes These changes (such as non-flexfield user interface (UI) page customizations) are captured in a metadata sandbox.
- Data security changes These changes are additionally captured in a data security enabled sandbox.
- Changes in the generated flexfields business components These changes are captured in a flexfield that's deployed as a single flexfield sandbox.

Once you're ready to make sandbox changes available in the mainline metadata, either publish the metadata or data security sandbox, or deploy the flexfield. You can download only metadata and data security sandboxes as a sandbox file for import to another application instance.

The following table lists the differences among the types of sandboxes.



Type of Changes	Type of Sandbox	Method for Making Changes Available in Mainline Metadata	Downloadable?
Metadata	Sandbox	Publish sandbox	Yes
Data security	Sandbox enabled for data security changes	Publish sandbox	Yes
Flexfield	Flexfield deployed as a flexfield- enabled sandbox	Deploy flexfield	No

Only one sandbox can be active at a time. All changes made in an active sandbox are captured in that sandbox.

Environment

To customize your application in run time, you must first create a sandbox and then use tools, such as Page Composer to make the customizations. These changes remain within the sandbox and don't affect the mainline metadata. You can test and validate the changes by publishing the sandbox to a full test environment. After testing the application, you can move to the production environment. The customizations created in the sandbox will be migrated to the production environment and will be available to the users.

Don't make customizations directly in the mainline metadata. Make all customizations first in the sandbox. When you make changes to an application at run time in a sandbox, you isolate the changes from the mainline metadata. After completing the changes in the sandbox, verify them. When you're ready to save the changes, publish the metadata or security-enabled sandbox to the mainline metadata.

When you create a sandbox, you can only see the customization information of the existing customizations in the current mainline metadata. For example, suppose you do customization in a sandbox and publish it. Then, on creating another sandbox for the next customization, you will see customization1 in the new sandbox because customization1 exists in the current mainline metadata.

Flexfield sandboxes are for testing only and can't be published. Make flexfield configurations that are stored in a database. Then, deploy those configurations to a sandbox to see the resulting deployment artifacts in a sandbox environment. Flexfields are deployed directly to the mainline metadata using the flexfield user interface.

Tools

You can use several run time tools to customize the application. For example, you can customize objects and pages using Page Composer, which uses sandbox manager. Oracle Business Process Composer and Oracle SOA Composer are also run time customization tools, but they don't use sandbox manager. They have their own mechanisms for handling customization changes.

Metadata sandboxes that you create using sandbox manager are available in Oracle JDeveloper while creating and deploying customizations for a deployed application in Oracle WebLogic Server. During deployment, the available sandboxes (except security sandboxes) appear in a selection list in Oracle JDeveloper. Oracle JDeveloper is not available in Oracle Cloud implementations. You can save, download, and import the metadata sandbox sessions as files into other applications.

Managing a Flexfield Sandbox

To create a flexfield-enabled sandbox, deploy one flexfield to a sandbox using the Manage Flexfield task flow. The flexfield sandbox gets its name from the flexfield you deploy. You can't test two flexfields in the same sandbox. After deploying a flexfield as a sandbox, sign out and sign in again to view how the sandbox run time reflects the flexfield changes, such as new segments. You can redeploy the same flexfield to the same sandbox repeatedly as you make incremental changes to the flexfield setup. A flexfield sandbox can't be published. So, when the flexfield is deployed to the mainline metadata, any page



customizations or data security in the flexfield sandbox can't reach the mainline metadata. If you're entitled to do so, manage flexfield-enabled sandboxes in the Sandbox Manager.

Related Topics

Deploying a Flexfield to a Sandbox: Points to Consider

Using Sandboxes: Points to Consider

In the customization run time workflow, use sandboxes to isolate the changes from the mainline metadata for testing and validating. After you're satisfied with the changes, you can publish the changes back to the mainline metadata.

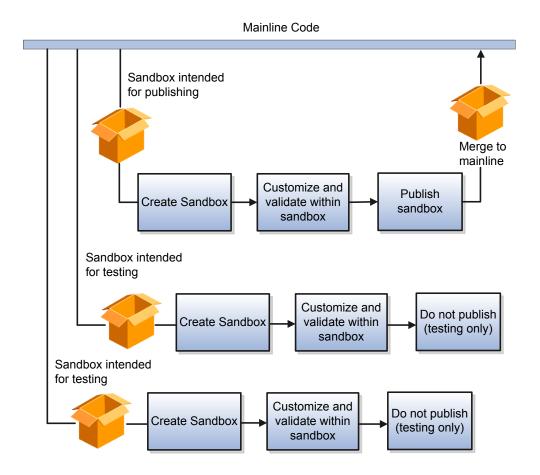
You can create two types of sandboxes:

- Sandboxes that are intended for testing purposes only
- Sandboxes that are intended to be published

The testing sandboxes are never published and therefore produce no concurrency conflicts between sandboxes. You can have several testing sandboxes at the same time. But if you have multiple users working on the same testing sandbox, then they must adhere to the prescribed guidelines.



Customizations in the sandboxes that are published are merged back to the mainline metadata. The following figure illustrates the two types of sandboxes and their relationship to the mainline metadata.



Working with a Single Sandbox

When multiple users are customizing an application using the same sandbox at the same time, conflicts within a sandbox may arise. This conflict may arise because multiple users attempt to customize the same artifact or perform customization tasks that indirectly affect other shared files. For example:

- A direct conflict arises when different users attempt to customize the same page, fragment, or metadata file within the same layer.
- An indirect conflict arises when two users, each creating a different object, save their changes at the same time. This conflict occurs in the metadata file that tracks which new objects both users created while saving their changes.

Conflicts may also arise when users edit a shared artifact, such as when a user performs an operation that adds or edits a translatable string. For example, say:

- One user edits a field's display label or help text, or a validation rule's error message. Whereas, another user performs an operation at the same time that similarly affects translatable strings.
- Two or more users are working in navigator menus that are shared across applications. Whenever a customization
 conflict arises among users, the application displays concurrency warning messages.



Whether the sandbox is meant for testing or production, if multiple users work with a single sandbox, follow these guidelines to avoid conflicts:

- 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 must not update object1. Suppose both modifications involve changes to translatable strings, and the users save changes to separate objects at the same time. Even then, a conflict can occur in the resource bundle that stores the translatable strings.
- If multiple users update the same artifact (same object or same underlying frequently modified file) concurrently, then they'll get a concurrent update error. The second user's updates won't be saved (the Save button will be disabled), and one of the users will have to cancel and try again.
- All users using the same sandbox should have the same application role. Users with different roles might not be able to see all content created by other users within the sandbox.

Working with Multiple Sandboxes

Multiple sandboxes are used when customizations are stored in testing as well as production sandboxes. Say, after you create a sandbox, a concurrent change is made in the mainline metadata. Now, when you attempt to publish that sandbox, the application detects such conflicts at publication time, and you get error messages.

Note: When you publish your sandbox, you may get a message showing a conflict on oracle/apps/menu/fnd/applcore/dataSecurity/dataSecurityService/mds/DSMO.xml. This message indicates that the security changes that you made in your sandbox conflict with other security changes in the mainline metadata. Delete the sandbox and recreate your changes in a new sandbox.

If multiple users are permitted to work in multiple sandboxes at the same time, follow these guidelines to avoid conflicts:

- Any number of test-only sandboxes can operate 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 aren't published, cause no conflicts with each other. Be aware, however, that all modifications will be lost when
 the sandboxes are destroyed.
- For sandboxes that aren't for test-only and will be published, users can use multiple concurrent sandboxes only if they operate on mutually exclusive artifacts. For example, you can have:
 - One sandbox that contains a page that a user is customizing to add a task flow
 - o Another sandbox that contains a different page from a different application

However, some objects might still share underlying artifacts, and thus it's not always obvious if two objects are truly mutually exclusive. Thus, proceed with caution when using multiple concurrent sandboxes that will be published. It's still possible that a conflict could occur, which would require the deletion of one or more sandboxes.

- Suppose the users update an artifact in both, the mainline metadata and in one sandbox, or in two different sandboxes. Now, when you publish the sandbox, the application detects such conflicts and you get an error message. At this point, cancel publishing the sandbox to avoid overwriting previous changes.
- Note: For a sandbox that contains ADF Business Components customizations, sign out and sign in again after switching in or out of this sandbox. This process ensures avoiding any inconsistencies between the run time caches and the ADF Business Components definitions.



Setting Up Sandboxes: Procedure

To make customizations to the application artifacts, you need to first store them in an active sandbox. You can either create a sandbox or select an existing sandbox, and designate it as an active sandbox. The active sandbox holds the context for all the changes. The sandbox uses a database to store the actual changes. After testing your changes, you can publish the sandbox, or deploy the flexfield, and the changes are merged into the mainline metadata. Eventually, the sandbox is archived.

The following procedure is for setting up non-flexfield sandboxes. For flexfields, use the Manage Descriptive Flexfields task or the Manage Extensible Flexfields task.

▲ Caution: Do not import or delete metadata files. These operations modify sandbox contents and could cause issues with the sandbox functionality.

To create a sandbox and set it up:

- 1. Click your user name in the global area, and select Manage Sandboxes from the Administration menu.
- 2. Use the Manage Sandboxes dialog box to create a sandbox.
- 3. Click Save and Close.
- **4.** On the Manage Sandboxes dialog box, select the new sandbox or an existing one, and click **Set as Active**. The sandbox is designated as the active sandbox.
- 5. Close the Manage Sandboxes dialog box.

Related Topics

• Hierarchy in Profile Levels: Explained

Publishing Sandboxes: Procedure

After completing the customizations in the sandbox, publish them to make them available in the application.

Prerequisites

Before publishing the customizations, test or validate the changes at run time using test systems and any combination of the validation setups.

If there are changes to the mainline metadata from another source and you publish your sandbox data, then the mainline metadata isn't overwritten. However, if you get error messages notifying about conflicts, then you must fix the conflicts before publishing.

Publishing Sandboxes

To publish a sandbox:

- Click your user name in the global area, and select Manage Sandboxes from the Administration menu.
- 2. On the Manage Sandboxes dialog box, select the sandbox and click **Publish**. The Publish confirmation message box appears.
- 3. Click **Yes**. The sandbox is published to the mainline metadata.
- 4. Close the Manage Sandboxes dialog box.



Moving Customizations

Using Customization Migration to Move Customizations: Points to Consider

Use the Customization Migration page to create a set of all customizations and extensions made to an application environment. Then, download the customization set and upload it into another environment. The customization set includes customizations across all product families. To open the Customization Migration page, select **Tools - Migration** from the Navigator menu.

Contents of the Customization Set

The customization set includes:

- Customizations done using Application Composer, except 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
 - User names and passwords for secured SOAP web service connections
 - o The enabled attachment feature for custom objects
- Customizations done using:
 - Page Composer
 - Appearance page
 - Structure page
 - User Interface Text page
 - Page Integration Wizard page
- Changes in the following artifacts of the Applications Core Setup application:
 - Messages
 - Lookups
 - Data security
 - o Descriptive, extensible, and key flexfields, and value sets
 - Attachment categories and entities
- Changes in scheduled processes
- Customizations done using the Manage Oracle Social Network Objects task
- Changes in security settings made in Application Composer
 - Note: Enterprise roles, new duty roles, and role hierarchy changes, which are made directly in Security Console aren't migrated.



- · Customizations done using Oracle Business Intelligence Enterprise Edition, including but not limited to:
 - Oracle Business Intelligence Answers
 - Oracle Business Intelligence Delivers
 - Business Intelligence Composer
 - Dashboard Builder
 - o Oracle Business Intelligence Publisher
 - Note: You can move the customizations done using the business intelligence tools only if the Disable BI for Customization Set Migration profile option is set to **No**.

The customization set doesn't include personalizations.

While an upload or restore activity processes Presentation Services customizations, the following can occur:

- Reports that were submitted by Oracle Enterprise Scheduler to Oracle Business Intelligence Publisher and were scheduled to execute during the process, will fail.
- The Reports and Analytics pane doesn't display.
- Oracle Business Intelligence Publisher reports may not display on Oracle Business Intelligence Presentation Services analyses or dashboard pages.
- Users may not be able to access Oracle Business Intelligence Enterprise Edition features, such as:
 - Oracle Business Intelligence Answers
 - Oracle Business Intelligence Delivers
 - Business Intelligence Composer
 - Oracle Business Intelligence Interactive Dashboards

Some important points to consider are:

- The customization migration doesn't include code extensions, such as managed beans, that you implement in Oracle JDeveloper using the applications developer role. These code extensions are stored in the app-inf/lib and web-inf/lib directories and you must manually move the extensions.
 - Note: Oracle JDeveloper isn't available in Oracle Cloud implementations.
- The type of customizations across all applications that will be added to the customization set are selected by default on the Customization Migration page. You can't change this selection.
- To prevent including in-progress customizations in the customization set, make your customizations in a sandbox. The customization set doesn't include customizations from a sandbox until the sandbox is published.

You can use the Customization Migration page to move customizations and extensions from any source environment to any target environment. However, you must always perform your customizations and extensions in a full test environment. Then use the Customization Migration page to move these changes to a production environment. As customization set migration doesn't provide a merge capability, never customize or extend a production environment. 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.

The customization set doesn't include all deletions. For example, the set does not include the removal of a customization document using the Manage Customizations dialog box. After you import a customization set into the target environment,



you must examine the environment for any deletions that you must make manually. Similarly, the customization set does not include roles or role hierarchy changes. Changes made to Security Console have to be manually updated in the target environment.

Related Topics

Importing and Exporting Custom Objects: Explained

Creating and Applying Customizations Using a Customization Set: Procedure

Create a customization set to move customizations across all the product families of the application from one environment to another. You can export all customizations, such as those stored in Oracle Metadata Services repository, JEDI, CRM, and BI using the Outgoing tab of the Customization Migration page. You can then import them to the target environment using the Incoming tab. You can use a customization set to move customizations in a batch instead of moving them one by one.

Prerequisites

Before creating a customization set, ensure that:

- The source and target application environments are of the same release and the same standard and one-off patches are applied to both environments.
- All Page Composer and Oracle JDeveloper (not available in Oracle Cloud implementations) customizations made in sandboxes are complete before they're published. Before starting the export process, you must publish all complete customizations. All customizations and extensions made using the Structure page, the Manage Standard Lookups task, and Security Console, are complete.
- To move content created using Oracle Business Intelligence Enterprise Edition features, the Disable BI for Customization Set Migration profile option is set to No in source and target environments. To view this profile option, open the Manage Profile Options task in the Setup and Maintenance work area.
- You have been granted the following privileges, which enable you to access the Customization Migration page:
 - FND_MANAGE_OUTGOING_CUSTOMIZATION_SET_PRIV
 - FND_MANAGE_INCOMING_CUSTOMIZATION_SET_PRIV

Contact your security administrator for details.

- You never make customizations in the target or production environment while applying customizations.
- Note: If you make customizations to the production environment in emergency circumstances, you must make the same customizations to the test environment. Making the customizations to the test environment ensures that the customizations are included in the next customization migration.
- You don't perform customizations in the source environment during the export process.

Creating Customizations

To create customizations:

- 1. In the source environment, select **Tools Migration** from the Navigator menu.
- 2. From the Outgoing tab of the Customization Migration page, click Create Customization Set.



- ▼ Tip: If the Delete button appears for an existing customization set, click the button. This action removes the temporary files that are on the server from the previous customization set creation. You can't create a customization set until the previous set has been deleted.
- 3. Provide a name for the customization set.
- **4.** Optionally, type a description of the set.
- 5. Click Save and Close.
- Periodically, click Refresh to view the current status of the set creation. Eventually, the status changes to Ready for Download.

To see the detailed status of each customization type, expand Customization Details.

The process runs asynchronously. You can close the dialog box and return to it later.

- 7. Click **Download**, and specify the name and location for the file, and click **Save**. Ensure that the downloaded file is a JAR file.
- 8. After you download the file on your local file system, click **Delete** to remove the temporary files that were created on the server.

Applying Customizations

After you apply customizations, end users must sign out and sign in again to see the changes. Hence, apply customizations when a few people are signed into the environment. To apply customization to the target environment, follow these steps:

- 1. Open the Customization Migration page in the target environment.
- 2. From the Incoming tab, click **Browse**, specify the name and location of the customization set file, and click **Open**.

If the Browse button appears disabled, click **Delete** to remove the previously uploaded customization set from the environment and enable the Browse button.

- 3. When the status for the customization set is Ready to Apply Customizations, click Apply.
- 4. Periodically, click **Refresh** to view the current status of the Apply action.

The process runs asynchronously. You can close the dialog box and return to it later.

For Oracle Cloud implementations, if problems occur during an Apply action, log a service request using My Oracle Support at https://support.oracle.com.

- 5. Access the target environment and examine the environment for any deletions that you must manually make.
- 6. Deploy all flexfields that display a Patched status.
- 7. Perform the following steps to send the new and updated social network definitions to the social network server:
 - a. In the Setup and Maintenance work area, open the Manage Oracle Social Network Objects task.
 - **b.** As part of the applying customizations process, some objects are created or updated. If the Enabled value of such an object is anything other than No, trigger the process of sending its definition to the social network server. You can do it by disabling the object and enabling it again with its original status. For example, if the Enabled value is Manual, then:
 - i. Disable the object.
 - ii. Enable the object, and select the value, Manual.
 - iii. Click **OK** and save the changes.
- 8. Manually migrate all business processes created in the source environment to the target environment.



- 9. After applying customizations, perform functional testing to verify the changes. Suppose testing exposes problems with the customizations, such as importing more than you intended, or the changes weren't what you expected. In such cases:
 - a. Open the customization set in the Incoming tab of the Customization Migration page.
 - **b.** Click **Restore** to revert to the state before the customization set was applied.

Skip the next step in such cases.

You can view the process log to monitor the progress of the download or the applying process. This process takes approximately 15 minutes. If it takes any longer and you don't see any progress, click **Refresh**. You can either let the server take its time and click **Continue**, or click **Restart** to restart the export process.

Following are some important points regarding customization import and export:

- After an environment upgrade, any previous imports which were performed in an earlier release can't be reverted. However if a new import is submitted in the upgraded instance, then the most recent import can be reversed.
- Lookup values for lookup fields that exist in both source and target aren't overwritten during the customization import. The lookup values from source are added to the target and all the lookup values coexist for the same field. For example, the Status field in its source environment has values, Open and Closed. In the target environment this field has values, Yes and No. After the import, the Status field in the target environment has values, Open, Closed, Yes, and No.
- After importing, perform the following steps in the target environment to send the new and updated social network definitions to the social network server.
 - i. In the Setup and Maintenance work area, open the Oracle Social Network Objects task.
 - ii. On the Oracle Social Network Objects page, click Synchronize to synchronize a selected object, or click Synchronize All to synchronize all objects together.
- During customization import, the data security privileges aren't automatically revoked in the target environment. For example, say a specific privilege is granted in the target environment, but the corresponding privilege doesn't exist in the source environment. During import, the privilege in the target environment won't be automatically revoked. To address this issue manually, add such a privilege to the source environment and revoke it. The revoke action is picked up as a customization instance during the customization import process and applied to the target environment.
- You can create custom reports directly in the target environment. However, ensure that you create the custom reports and reference them to the existing custom subject areas. Don't create the custom subject areas directly in the target environment.
- You can initiate customization export and import tasks only from the mainline metadata. If you initiate customization from a sandbox, the process doesn't execute.
- ▲ Caution: All user personalizations that are performed after a customization set is applied are lost when you perform a restore action on that customization set.
- 10. Broadcast information to the users that they must sign out and sign in to view the most recent changes.

Related Topics

Importing and Exporting Custom Objects: Explained



Exporting and Moving Customizations: Points to Consider

Customizations are stored in XML files. You can use these XML files to export customizations for the following reasons:

- To move customizations and extensions to another environment, such as the production environment.
- To diagnose issues noticed in the test environment.
- To send files to your help desk for further diagnosing.
- To import a customization into another environment. For example, a customization developer using Oracle JDeveloper may see customizations that other users do.

The following table lists the tools to use to export and move customizations and extensions.

Tasks	Tools to Use
Move all customizations to another application environment.	Customization Migration.
Move only Oracle Metadata Services	Enterprise Manager Cloud Control.
customizations made to pages and the user interface to another application environment.	You can also use Enterprise Manager Cloud Control to download and upload a set of customizations.
Move only descriptive flexfield configurations to another application	Setup and Maintenance work area. Moves configurations for a specified module.
environment.	To move configurations for all modules, use Customization Migration.
Move only extensible flexfield configurations to another application	Setup and Maintenance work area. Moves configurations for a specified module.
environment.	To move configurations for all modules, use Customization Migration.
Move only value set configurations to	Setup and Maintenance work area. Moves configurations for a specified module.
another application environment.	To move configurations for all modules, use Customization Migration.
Move only lookups to application environment.	Setup and Maintenance work area. Move application standard lookups, application common lookups, or both.
Move only data security policies to	Setup and Maintenance work area.
another application environment.	It doesn't move Oracle Fusion Human Capital Management roles.
Export customizations to a file to help diagnose an issue.	Manage Customizations dialog box.
Export customizations to import them into an application workspace in Oracle JDeveloper.	Manage Customizations dialog box.



Note: Enterprise Manager Cloud Control isn't available in Oracle Cloud implementations. Therefore, in Oracle Cloud implementations, to perform tasks that require use of Enterprise Manager Cloud Control, log a service request using your help desk at https://support.oracle.com.

Downloading Customizations

Use the Manage Customizations dialog box to download customization files for a given page. You can also upload the files into the Oracle Metadata Services repository using the same dialog box. To open Manage Customizations dialog box, click your user name in the global area, and select **Manage Customizations** from the Administration menu. You can use these files for diagnosing customization issues.

You can also download all customizations of a page for all layers (AllCustomization.zip) using the **Download**Customizations for All Layers link. This link is located at the bottom of the Manage Customizations dialog box. The

AllCustomization.zip file contains all the customization XML files for the page. However, you can't upload this file anywhere.

Downloading Customization Set Reports

After exporting customizations, you can view and download a customization set report that contains a list of all customizations available in a customization set. To do so, click **Content Read Me** from the Outgoing tab of the Customization Migration page, select **Tools - Migration** from the Navigator menu.

The customizations in this report include all:

- Custom objects
- Custom fields
- Custom pages
- Business intelligence (BI) changes

Business logic changes such as Groovy scripts and triggers aren't included in the customization set report.

Related Topics

- Implementation Project Based Export and Import: Explained
- Moving Related Common Reference Objects: Points to Consider
- Customization Layers: Explained

Viewing and Deleting Customizations: Procedure

Use the Manage Customizations dialog box to view the customizations made to the application pages and to delete unwanted customizations. Click your user name in the global area, and select **Manage Customizations** from the **Administration** menu.

Deleting Customizations

To delete customizations:

 On the page that contains the customizations, select the page fragment or task flow, and then select Manage Customizations from the Administration menu.



- 2. In the Name list, select the correct layer, and find the page, task flow, or fragment that contains the customizations.
- 3. Click **Delete** for the customization document that you want to delete.

Advanced Customization Life Cycle Tasks: Highlights

You can perform design time tasks as part of the customization life cycle.

Note: These customization tasks aren't available in Oracle Cloud implementations.

Customization Tasks

• The logging functionality for customization set migration is different from the standard logging functionality for your application. You can adjust the amount of detail to log customization migration without requiring a server restart. For more information, refer to the Oracle Fusion Applications Administrator's Guide.

See: Configuring ApplSession Log Levels

 Developers can create customization metadata using Oracle JDeveloper, and package and deploy customizations to the source application environment. For more information, refer to the Oracle Fusion Applications Extensibility Guide for Developers.

See: Design Time Customization Workflow

See: Using Oracle JDeveloper for Customizations



3 Page Customization

Customizing Pages: Overview

Use Page Composer to customize page content, layout, and more. Using other tools, you can create new pages and customize UI text, themes, infolets, and so on.

For example, you can:

- Use tools such as User Interface Text Customization to customize UI text.
- Use the Appearance page to change the look and feel of the application.
- Open an infolet page in Page Composer to customize it.
- Use the Page Integration page to create and manage pages for hosting third party applications.

Page Composer Overview

Prerequisites for Customizing Existing Pages: Explained

Before customizing pages, do the following tasks:

- Understand the typical workflows for working with run time customizations.
- Verify that the page is customizable. To do so, check if either the Customize Pages or the Customize <Page Name>
 Pages menu item is available under the Administrator menu. If no, then that means the page can't be customized.
- Confirm that your privileges are sufficient for customizing the page.
- Activate a sandbox.

Related Topics

• Run Time Customization Workflow: Explained

Customization Layers: Explained

Setting Up Sandboxes: Procedure

Page Composer Views: Explained

You can use either Design view or Source view for viewing and customizing page content and layout in Page Composer. To open a view option, select it from the View menu at the top left corner of the page. Although both views share many common customization features, you can use some unique features in each view.



Tip: Use the **Dock** option at the bottom right corner of the page to specify whether the page source appears on top, bottom, left, or right of the Selection pane. This option is available only when you open the page in Source view.

Design View

In Design view, you see one region that shows a WYSIWYG rendering of the page and its content. Work with components directly and access their properties from the toolbar in the chrome of the individual component.

Source View

In Source view, you see two regions:

- Selection pane, showing a WYSIWYG rendering of the page and its content
- Source pane, showing a hierarchical ordering of the page components, including some components that otherwise don't appear on the page. You can select and configure such components in Source view.
- 💡 Tip: Controls on individual components are inactive in Source view, but you can click an individual component to select it.

In Source view, you can:

- Click a component in the Selection pane to highlight the component in the hierarchical list. The cursor turns to a magnifier and a blue outline appears around the component selection. You can also traverse the hierarchy and select components directly.
- Click Edit on the view header to work with components indirectly and access their properties. You can also right click the object in the hierarchy, and click Edit.

Page Component Properties: Explained

All components have configurable properties that control, or express their appearance and functionality. Many properties are common to all component types, while some properties are unique to one component type. Use the Component Properties dialog box to view the properties of a component. To open this dialog box, select the component that you want to edit, and click **Edit**. You can see properties of similar functions under tabs that name the category of the properties.

Note: Properties and tabs can vary from component to component.

Component Property Tabs

This table describes the tabs that you may see in a component properties dialog box.

Tab	Description	
Parameters	Settings that control component aspects that are specific, or often unique to the component.	
	For example, on a page containing a map, a component may have a parameter that provides a choice between units of measurement.	



Tab	Description
Display Options	Settings that affect the chrome of a component, including: Header, header text, and border Actions menu Edit, Remove, Expand, Collapse, and other icons Tooltips
	For example, display options on image layout components specify the image source URL and its optional link target.
Child Components	The list of all components contained within and under the control of the parent component, including controls for ordering the child components.
Style Content Style	Settings that affect the look and feel of the component chrome or the component contents. These settings override corresponding values from a parent object, such as a component, page, and application, providing an opportunity to fine-tune appearance.
	For example, font, color, and dimension. Some style properties may be disabled at the component level if other page or application elements (such as the skin) don't support modification to the property.
Events	Events and event handlers for all the components on the current page that the current component can consume.
	For example, an event can be selecting a check box within the current context. The code that runs and drives the result of an event, such as making another component visible, is an event handler.

Working With Components in Page Customizations: Procedure

Use the Resource Catalog to customize pages. This catalog provides a selection of task flows, portlets, and layout components. Open the Resource Catalog in Page Composer from either Design view or Source view.

Aspects of components pertinent to page customizations include:

- Opening the Resource Catalog
- Adding components
- · Hiding components manually
- Hiding components programmatically

Prerequisites

Activate a sandbox.

Opening the Resource Catalog

In Design view:

- 1. Open the page that you want to customize in Page Composer.
- 2. From the existing components, select the one that you want to be the parent component.



- Tip: Alternatively, use the Add Box icons (Add Box Above, Add Box Below, Add Box Left, and Add Box Right) to insert a box component. Then select the component as the parent component.
- 3. Click the Add Content button associated with the parent component.

The Resource Catalog appears.

Adding Components

In Design view:

- 1. Open the Resource Catalog.
- 2. In the Resource Catalog, find the component that you want to add.
- 3. Click the **Add** icon associated with the component.
- **4.** Cut and paste, or drag and drop the component to place it.

In Source View:

- 1. Select the container component in the selection pane.
- 2. In the Source view toolbar, click Add Content.
- 3. In the Resource Catalog, find the component you want to add.
- 4. Click the **Add** icon associated with the component.

Hiding Components Manually

Use the **Show Component** property to specify whether the component appears to users at run time. By default, all components are visible. To manually hide a component, deselect **Show Component** on the Component Properties dialog box.

If the component is a:

- Child component, then deselecting the **Show Component** property hides only the child component.
- Parent component, then deselecting the **Show Component** property of the parent component hides the parent and all child components it contains. So, when you hide a parent component, you automatically hide all child components.

You can do any of the following:

- Hide a child component directly
- Hide a child component from within the parent component
- Hide a parent component and all child components

To hide a child component directly:

- 1. Click the **Edit** icon in the header of the child component. This opens the Component Properties dialog box.
- 2. Click the **Display Options** tab.
- 3. Deselect Show Component.
- 4. Click OK.

To hide a child component from within the parent component:

- 1. Click the **Edit** icon on the containing box's toolbar.
- 2. Click the Child Components tab.
- 3. Deselect the box next to the component you want to hide.
- 4. Click OK.



To hide a parent component and all child components:

- 1. Click the **Edit** icon in the box header.
- 2. Click the **Display Options** tab.
- 3. Deselect Show Component.
- 4. Click OK.

Hiding Components Programmatically

You can add an Expression Language (EL) expression to a component that enables you to set a condition for hiding the component. For example, suppose you have two check boxes (1 and 2) on a page. You also have a button (B) that you want to be visible only if check box 2 is selected. To step through the logic, ask yourself questions such as the ones in the following table.

Question	Answer	Purpose of Question
What's the condition?	Check box 2 is selected	Determines what the occurrence, or event, is.
What action or event must happen?		Determines the component that triggers the event.
		Determines what expression to write.
What happens when the condition is met?	Button "B" appears.	Determines the effect of the action.
What happens when the event happens?	Show the component: Button B	
	(The implication is that button B is hidden until the event occurs.)	
What property determines whether a component is visible?	The Show Component property	Determines the property the code affects.

So the logic is: If 2 is checked, then the Show Component property of B is activated.

You place the expression on the component that receives the action.

Here's a sample code that you may add to the component.

#{if checkbox2.selected = true}

After you think through the logic and find the correct expression, add it to the property. You can add an expression using the expression builder for the Show Component property only on dashboard pages; not on work area pages. Also, only administrators can perform this task.

To open the EL Editor and add an expression to a property for a dashboard page:

- 1. Click the **Edit** icon in the component header.
- 2. Click the **Display Options** tab.
- 3. Click the Edit icon next to the Show Component property, and select Expression Builder....
- **4.** Add an expression to check for an event or condition, and set the property. Based on the result, turn the property on or off.

To hide a parent component and all child components programmatically for a dashboard page:

- 1. Click the **Edit** icon in the box header.
- 2. Click the **Display Options** tab.



- 3. Click the Edit icon next to the Show Component property, and select Expression Builder....
- Add an expression to check for an event or condition, and set the property. Based on the result, turn the property on or off.

Related Topics

• Setting Up Sandboxes: Procedure

Customizing Tabs on Application Pages Using Page Composer: Worked Example

This example demonstrates how to customize tabs on pages using Page Composer.

The following table summarizes the key decisions for this scenario.

Decisions to Consider	In this Example
Who do you want to make the customization for?	All users
What customizations are you going to do?	Hide the Incentive Management tab and the Projects tab from the Worklist: Notifications and Approvals page

Prerequisites

Activate a sandbox.

Customizing Tabs

Do the following:

- 1. From the Navigator menu, select Tools Worklist.
- 2. Click your user name in the global area, and select **Customize Pages** under Administration.
- 3. Select **Site layer** as the customization layer, which affects all users.
- **4.** In the View menu at the top of the page, make sure that the Design view is selected, which lets you navigate to the component you want to customize.
- 5. Click the **Select** tab.
- 6. Hover over any tab on the page, and click when the cursor turns to a magnifier and a blue outline appears around the tab.
- 7. Select Edit Parent Component.
- 8. In the Component Properties dialog box, click the **Children** tab.
- 9. Deselect Incentive Compensation and Projects to hide these tabs from the page.
- **10.** Click **OK**.
- 11. Close Page Composer.

Related Topics

Setting Up Sandboxes: Procedure



Making Customizations Visible Based on User Roles: Worked Example

This example demonstrates how you can make customizations visible to a specific user role.

To control page components conditionally based on user role:

- Create security privileges
- Add an Expression Language expression to the component property that you want to control

The following table summarizes the key decisions for this scenario.

Decisions to Consider	In This Example		
What's the page object you're securing?	Reports link		
To which user role you want to make the customization visible?	Hiring managers, Sherry Callaway and Terrance Whitaker		
	▼ Tip: Create a role just for testing customizations. Call it DEVCUST_TEST_ROLE. When you're sure that the customization works, change the security to the appropriate role.		
Which expression to add for verifying whether a user has the appropriate privilege?	#{securityContext. userGrantedPermission['MANAGERREPORTS LINK_PRIV']}		

As a prerequisite, activate a sandbox.

Making Customizations Visible to a User Role

Follow these steps:

- 1. Create a privilege.
- 2. Add the 'manager reports link priv' object to 'devoust test role'.
- 3. Assign **DEVCUST TEST ROLE** to Terrance and Sherry.
 - Note: Make sure that Terrance and Sherry have access to the page before you make any changes.
- **4.** Open the page, having the Reports link that you want to customize, in Page Composer, and select the Reports link component.
- 5. Click the **Edit** icon. This opens the Component Properties dialog box.
- 6. Click the **Display Options** tab.
- 7. Click the Edit icon next to the Show Component property, and select Expression Builder....
- 8. Add an expression to verify whether the user has the appropriate privilege. Use this sample code:
 - #{securityContext.userGrantedPermission['MANAGER REPORTS LINK PRIV']}
- 9. Save the property changes, and close Page Composer.

After you change the show component property, whenever users open this page, the application evaluates the expression. Since Sherry and Terrance have the privilege, the Show Component property evaluates to be selected. Hence, Sherry and Terrance can see the Reports link while all other users can't.



Tip: Use Oracle JDeveloper to define custom permissions that render UI components based on the user's access rights. (Oracle JDeveloper is not available in Oracle Cloud implementations.)

Related Topics

Setting Up Sandboxes: Procedure

Customizing Objects That Appear on Multiple Pages: Points to Consider

Use Page Composer to customize objects that appear on multiple pages. Whether your customizations affect one or more pages depends on the way you include the object on the page and the applications that you use.

Consider these points when you customize objects that appear on multiple pages.

Customization Appearance Based on Shared and Non Shared Task Flows

The following table briefly describes the task flow scenarios when object customizations appear on one or more pages.

If the object is	Then the customizations
Not part of a shared task flow	Don't appear on other pages
Part of a shared task flow	Do appear on all pages that include the shared task flow



🛕 Caution: When shared task flows include embedded logic that uses data from the page, the logic can override the customizations that you make in Page Composer.

FAQs for Page Composer Overview

How can I move page components?

To move page components using Page Composer:

- In Design view, drag and drop the component
- In Source view, do any of the following:
 - Cut and paste the component
 - Drag and drop the component
 - Open the Component Properties for the container component and rearrange the components on the Child Components tab



How can I delete components from a page?

Click the **Delete** icon in the component header in Page Composer.

Caution:

- Delete a component only if you're certain that no other components or processes depend on that component. If you're unsure whether any dependencies exist, then hide the component instead of deleting it.
- If you delete a parent component, you delete all the child components automatically.

How can I work on customizations, but prevent users from viewing them until the customizations are complete?

Create or select an appropriate sandbox, and set it as active to capture your customizations using Page Composer. When you're ready, publish the sandbox to make your changes available to users.

Related Topics

- Managing Customizations Using Sandboxes: Explained
- Setting Up Sandboxes: Procedure
- Publishing Sandboxes: Procedure

What happens if my customizations or personalizations make the page inaccessible?

You must contact an administrator, and the administrator can use the Manage Customizations task to view and, if necessary, delete your changes.

How can I reset a page or task flow to a previously saved version?

To reset a page to a previously saved version or the original out-of-the-box state, click **Reset Page** in Page Composer (Design or Source view).

To reset a task flow to a previously saved version or the original out-of-the-box state, click **Reset Task Flow** in Page Composer (Source view only) while you have the task flow open.

How can I use metadata to perform customization-related tasks that I can't do in Page Composer?

In Page Composer, open the Manage Customizations dialog box, and download the metadata to inspect the existing customizations in the metadata.

Page Content and Template Customization

Changing Page Layout Using Page Composer: Procedure

Page layout defines the number, placement, and orientation of content regions on a page. Although you set the layout style while creating a page, for some layouts, you can change the layout style even after adding content to the page.

Note: You can't change the page layout for all pages.



Prerequisites

Activate a sandbox.

Changing Page Layout

To change the layout of a page:

- 1. Open the page in Page Composer.
- 2. From the View menu, select **Design**.
- 3. Click Change Layout.
- 4. Select the new layout.

Related Topics

Setting Up Sandboxes: Procedure

Customizing the Global Page Template: Explained

The global page template provides a common header area and the footer panel for all pages in your application. You can use Page Composer to customize the global page template. To open the global page template in Page Composer, click your user name in the global area, and select **Customize Global Page Template** from the Administration menu.

You can make the following customizations to the global page template:

- Add components
- Edit components

Example: Add expression language to hide the tags link

Delete components

Example: Remove the tags link

Tip: When you move your cursor over the global page template, the areas that you can edit display a blue outline.

You have two separate global page templates - one for the desktop user interface (UI) and the other for the simplified UI. If you click **Customize Global Page Template** from a desktop page, you will customize the template for the desktop UI. Similarly, if you click **Customize Global Page Template** from a simplified page, you will customize the template for the simplified UI. Hence, to ensure a consistent look and feel for all pages, you must customize the global page template twice, once for each UI.

Adding Components to the Global Page Template

To add components to the global page template:

- 1. Open the global page template in Page Composer.
- 2. Select the portion of the global area to which you want to add a component, and click Add Content.
- 3. In the component catalog, select **Components** to display the list of available components.
- 4. Click the Add button associated with the component you want to add.

The component appears in the global area.



- 5. Change component properties, as appropriate. For example, if you added the Text component, enter the text that you want to display.
- 6. After completing your changes, click Close. When prompted, click Save to save your changes.

Editing Components in the Global Page Template

To edit components in the global page template:

- 1. Open the global page template in Page Composer.
- 2. Select the component that you want to edit.
- 3. Click Edit.
- 4. Edit the component properties, and click **OK** to save your changes.
- 5. After completing your changes, click **Close**. When prompted, click **Save** to save your changes.

Deleting Components from the Global Page Template

To delete components from the global page template:

- 1. Open the global page template in Page Composer.
- 2. Select the component that you want to delete.
 - When you move your cursor over the global page template, the areas that you can edit display a blue outline.
- 3. Click **Delete**. When prompted, click **Delete** to delete the component.
- 4. After completing your changes, click Close. When prompted, click Save to save your changes.

Editing Footers in the Global Page Template

To edit footers in the global page template:

- Note: For simplified pages, you can't customize footers in the global page template.
- 1. Open the global page template in Page Composer.
- 2. Select the left region of the footer area. In the Source region, make sure the spacer: 40px node is selected after you select the left region of the footer area.
- 3. In the Source region, select the panelGroupLayout: horizontal node that appears above the spacer, and click Add Content
- 4. In the component catalog, select **Components** to display the list of available components.
- 5. Click the **Add** button associated with the component you want to add. The component appears in the footer.
- **6.** Change component properties, as appropriate.
- 7. After completing your changes, click Close. When prompted, click Save to save your changes.

For information about making skin customizations, such as selecting a different color palette, see the Oracle Fusion Applications Extensibility Guide for Developers.

Customizing Page Titles: Procedure

Page titles appear in multiple places in the application: the browser header, page headings, tab titles, and as items in task lists and the Navigator.

Three separate title and label properties control the different places where the title appears. The properties are:

- Task list Page Title property
- Task list task Label property



Navigator item node Label property

To customize a page title, you must enter the same value for all three properties. You can change the two task list properties using Page Composer, and the navigator item node property using the Structure page.

Tip: You can also make text changes in page titles, task labels, and navigator item nodes using the User Interface Text tool.

If your application requires language localization, you must provide the translated custom text. For more information about translating custom text, see the Oracle Fusion Applications Extensibility Guide for Developers.

Prerequisites

Activate a sandbox.

Task List Page Title Property

Use this property to control the text that appears in the browser header.

To change this property:

- 1. Open the page you want to customize, and then open Page Composer.
- 2. From the View menu, select **Source**.
- 3. In the Selection pane, click a task list link. An edit warning confirmation dialog box appears.
- 4. Select **Do not ask next time**.
- 5. Click Edit.
- In the Source pane, right-click the panelFormLayout node, and select Edit. The panelFormLayout Component Properties dialog box appears.
- 7. Select the **Tasks List Properties** tab.
- 8. Enter the new value in the Page Title field.
- 9. Click **Apply** to save the changes, if you are changing the next property now. If you keep the dialog box open, skip the next step and then start at step 6 in the next section.
- 10. Click **OK** to save the changes and close the Component Properties dialog box.

Task List Task Label Property

Use this property to control the text that appears:

- In page headings and tab titles
- As menu items in task lists

To change this property:

- 1. Open the page you want to customize, and then open Page Composer in Source view.
- 2. In the Selection pane, select the task list. A confirmation dialog box appears.
- 3. Click **Edit** to edit the task flow.
- 4. In the Source pane, click the **Edit Task Flow** link next to the first subordinate region node.
- 5. Click the Edit icon to edit the task flow. The Component Properties dialog box appears.
- 6. Click the Tasks List Task Properties tab.
- 7. Expand the tree to display the child items in the task list hierarchy.
- 8. Right-click the child item that you want to customize and click Edit in the toolbar.
- 9. Enter the new value in the Label field.
- 10. Click **OK** to save the changes and close the Component Properties dialog box.



Navigator Item Node Label Property

Use the Structure page to control the names of categories and page entries that appear as navigator menu items.

To change this property:

- 1. From the Navigator menu, select **Tools Structure**.
- 2. On the Structure page, click the name link for the category or page entry.
 - Tip: You can use the search panel on the Structure page to find the category or page entry that you want to rename.
- 3. On the Edit Page Entry or Edit Category page, enter the new value in the Name field.
- 4. Click Save and Close.

Related Topics

- Managing Categories and Page Entries for the Navigator and Springboard: Procedure
- Setting Up Sandboxes: Procedure

Customizing Task Lists: Procedures

Use task lists to organize direct links to task flows in your application and web pages outside your application. You link task flows and web pages to tasks through the Tasks List Task Properties in the Component Properties dialog box. Use the Source view of Page Composer to customize task lists.

To customize task lists, you should know how to:

- Insert tasks into the task list.
- Remove tasks from the task list.
- Reorder tasks in the task list.
- Change properties associated with tasks and task lists.

To perform these customizations:

- 1. Navigate to the task list that you want to change and open Page Composer.
- 2. If you are in Design view, change to Source view. You customize task lists in Source view only.
- 3. In the Selection region, click the task list.
- 4. Click **Edit** to confirm your intent and automatically select the task list entry in the Source region.
- 5. In the Source region, click the **Edit Task Flow** link next to the task list region.
- 6. Click Edit to confirm your intent and automatically select the task list entry in the Source region.
- 7. In the Source region, right-click the panelFormLayout node, and select Edit.
 - The Component Properties: panelFormLayout dialog box appears.
- 8. Click the Tasks List Task Properties tab.
- 9. Expand the tree to view the hierarchy of items in the task list.
- 10. Select an item in the task list hierarchy.
- 11. The toolbar provides multiple customization options. You can edit the selected item or insert a new task in the task list.
- **12.** If you insert a new task in the task list, you must enter values in several properties.
 - The following table describes the properties that you use to create a task in your task list.



Property	Description or Value
Web Application	Select the target web application from the list of web applications that you defined in the deployments tables.
	Oracle Cloud customers must contact the help desk at https://supportoracle.com to log a service request and obtain a list of valid values.
Focus View Id	Enter the focusViewId value of the target page, for example, / ServiceRequest
	Focus View ID and Web Application are mutually inclusive properties.
Action	Enter the action that occurs when the user selects this item in the task menu.
	Pages with actions appear in the adfc-config. xml file. 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 a name for the new task.
	This name appears in the task list menu and on the task tab when opened if the value of the page attribute isDynamicTabNavigation is true.
Rendered	Select this to display the item in the task list. Deselect to hide the item.
	To help you identify hidden items quickly, deselected items appear in italics on the customization dialog box.
Destination	Enter the full URL for the item, such as http://www. example. com. Use to open a new window that takes you out of the Oracle Fusion Middleware UI Shell pages.
	Destination takes precedence over any specified Web Application value.
Task Type	Select the task type for newly created items from the list of these values:
	_o dynamicMain
	o defaultMain
	o defaultRegional
	o taskCategory

13. Save any changes that you made to the properties, and then save the page in Page Composer.

Customizing Dialog Box Content Using Page Composer: Procedure

Use Page Composer and work in source view to customize the content in your dialog boxes.

Prerequisites

Activate a sandbox.



Customizing Dialog Boxes

To customize dialog box content:

- 1. Open the page where the dialog box appears, and then open Page Composer.
- 2. From the View menu, select Source. You must be in Source view to customize dialog box content.
- 3. Select the button that opens the dialog box.
- 4. Click Edit to open the Component Properties dialog box.
- 5. Click the Child Components tab.
- 6. Edit the dialog box content.
- 7. Click **Apply** to save your changes, then **OK** to save your changes and close the Component Properties dialog box.

Related Topics

• Setting Up Sandboxes: Procedure

Saved Search Customization

Making Saved Searches Available to All Users: Procedure

Use Page Composer to customize saved searches, and make them available for all users. Create and edit saved searches using Page Composer at the site layer. Users can run these saved searches again later to use the same search criteria and other settings. You must create or edit saved searches only at the site layer to make them available for all users.

Customizing Saved Searches for All Users

Follow these steps:

- Activate a sandbox.
- On the search page that has a Save button, click your user name in the global area, and select Customize <Page Name> Pages under Administration.
- 3. If prompted to select a customization layer, select the site layer to open the search page in customization mode.
- 4. From the View menu, select **Design**.
- 5. Create and edit saved searches.
 - Note: The steps for creating and editing saved searches are the same regardless of whether you're working on saved searches for yourself or for all users.
- 6. Save your changes and close Page Composer.
- 7. After testing your changes, publish the sandbox to make your changes available to all users.

Related Topics

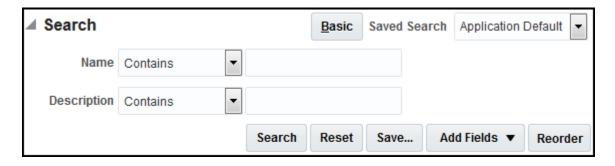
- Setting Up Sandboxes: Procedure
- What gets saved when I create a saved search for searches with multiple criteria?

Saving Searches for Searches with Multiple Criteria: Procedure

On many pages, you can run a search with multiple search criteria to find specific business objects. Some of these searches have a Saved Search list, as well as a Save button below the search criteria. A saved search captures search criteria and other settings so that you can easily run the same search again later. Aside from using any predefined saved searches, you



can create and edit them for your own use. If you have the right roles, you can also customize saved searches for other users using Page Composer.



Creating Saved Searches

Follow these steps:

- 1. Go to a search that has a **Save** button.
- 2. Optionally add or reorder fields, if available.
- 3. Enter your search criteria values, and, click the **Save** button.
- **4.** Name your saved search and define its settings:
 - Set as Default: The saved search is automatically selected whenever you open this page.
 - Run Automatically (if available): The saved search runs on this page as soon as you select it from the list of saved searches.

If you select both options, then the saved search automatically runs whenever you open this page.

5. Close the dialog box.

Your saved search can be limited to the current page, or in some cases available in other searches for the same object.

Changing the Search Criteria in Saved Searches

Follow these steps:

- 1. Select the saved search if it's not selected already.
- 2. Set your search criteria, including any additional fields.
- 3. Click the Save button.
- 4. If the saved search is one that you created, save without changing the name.

If it's predefined, then you can't overwrite it, so you create a new saved search with a unique name.

Changing Settings or Renaming and Deleting Saved Searches

Follow these steps:

- 1. Go to a search that has a **Save** button.
- 2. Select Personalize from the Saved Search list.
- 3. In the Personalize Saved Searches dialog box, select a saved search.
- 4. Change any of the settings, where available.
 - Set as Default: The saved search is automatically selected whenever you open this page.
 - Run Automatically: The saved search runs on this page as soon as you select it from the list of saved searches.



- o Show in Search List: The saved search is available for you to select and run on this page.
 - You can still find hidden saved searches in the Personalize Saved Searches dialog box.
 - You can't change this setting if the saved search is currently selected on your page.
- Note: Some settings can't be changed for predefined saved searches. What you do change applies only to you, unless you're customizing the saved search using Page Composer.
- 5. If you selected a saved search that you created, then you can rename or delete it. You can't do so for predefined saved searches.
- **6.** Save your changes and close the dialog box.

Related Topics

- What gets saved when I create a saved search for searches with multiple criteria?
- Creating Watchlist Items: Procedure

Saving Searches for Keyword Searches with Filters: Procedure

One type of search you might find on the page is a keyword search with filters. Some of these searches have predefined saved searches, and you can also create and edit saved searches for your own use. A saved search captures entered search terms, filters, and other settings so that you can easily run the same search again later. If you have the right roles, you can also customize saved searches for other users using Page Composer.

Creating and Editing Saved Searches

Follow these steps:

- 1. Enter search terms in the search field, and click **Search**, or select a saved search in the autosuggest.
- 2. Click the **Show Filters** link if filters are currently hidden
- 3. Optionally add or reorder filters, if available.
- 4. Set filter values, and click the **Save** button.
- 5. Name your saved search.
 - o To create a saved search, enter a new, unique name.
 - o To update an existing saved search that you created, save with the original name.

You can't overwrite predefined saved searches by using their names, but you can create a copy with a unique name.

- 6. Click Set as Default if you want the saved search to be automatically selected whenever you open this page.
- 7. Click OK.

Your saved search can be limited to the current page, or in some cases available in other searches for the same object.

Changing Settings or Renaming and Deleting Saved Searches

Follow these steps:

- 1. Click the **Show Filters** link if filters are currently hidden.
- 2. Select Manage from the Saved Search list.
- 3. Define settings for any saved search, predefined or custom, in the Manage Saved Searches dialog box:
 - Default: The saved search is automatically selected whenever you open this page.
 - **Show in Saved Search List:** The saved search is available for you to select and run on this page. You can still find hidden saved searches in the Manage Saved Searches dialog box.



- Note: Changes you make to predefined saved searches apply only to you, unless you're customizing the saved search using Page Composer.
- **4.** The **Active** setting identifies the saved search that's currently selected on the page. You can designate a different active saved search to have that saved search automatically selected as soon as you click **OK** in this dialog box.
- 5. For custom saved searches only, you can also rename or delete the saved search.
- 6. Click OK.

Simplified Pages Layout and Content

Customizing Simplified Pages Using Page Composer: Procedure

On a simplified page, you can customize user interface (UI) components by updating their properties, for example to change field labels, hide the component, or make a check box required.

When you start customizing simplified pages, by default, you can use the Design view. In the Design view, you can add content and make layout changes only in some pages. For other pages, you must use the Source view to make such customizations.

- Note: Any changes you make apply:
- Only to the page you're on.
- To all or specific groups of users, depending on the customization layer you select before making changes.

Prerequisites

Activate a sandbox.

Customizing a UI Component

To update component properties:

- 1. Click your user name in the global area and select Customize Pages.
- 2. Select a customization layer, for example to make changes only for users with a specific job role.
 - Note: When you customize a UI component for a specific job role, that job role must be assigned to you for you to test the customization in the sandbox. Your security administrator can either assign the job role to you directly, or make the job role self-requestable for you to add it yourself from the resource directory.
- 3. By default, you start in the Design view, which lets you navigate to the component you want to customize.
 - You can tell you're in this view when the Design button above the page is highlighted. To use the Source view, you must select **Source** from the View menu. This menu isn't displayed by default. To display the View menu, and then select Source, set the **Source View for Page Composer Enabled** profile option (FND PAGE COMPOSER SOURCE VIEW) to Yes.
- 4. When you have found your UI component, click the **Select** button above the page.
- 5. Hover over the UI component until a border appears around the component, and click.
- 6. Select Edit Component.
- 7. Update the component's properties to make the customization you want.
 - Each component has its own set of properties, which may include some of the properties in this table. In Design view, you get the main properties only; but in Source view, you get all properties.



Property	Description
Description	Text used by screen readers, for information in addition to what is provided in the Short Desc property.
Label	Display text for the component, for example the field prompt or the single prompt for a group of check boxes.
Read only	Whether users can edit the component, for example if a check box can be selected or not.
Rendered	Whether the component is visible or hidden to users on the page.
Required	Whether users must enter something for the component before saving the page.
Short Desc	Text that appears when users hover or focus on the component, for example hover over a field label or click in the text box.
Show Component	Whether the component is visible or hidden to users.
Show Required	Whether an asterisk is displayed to indicate that the component is required.

- 8. To customize more components, click Add Content to return to the Design mode and repeat steps 4 to 7.
- 9. Click Close to save.

If available, click **Save and Label** instead to also label your changes so that you can later revert to the customizations you're saving. Labels are saved with a prefix of **composer_**. For example, if you enter **myLabel**, then the label is **composer_myLabel**.

Related Topics

- Working With Customization Layers: Examples
- Setting Up Sandboxes: Procedure
- Customizing the Navigator and Springboard: Overview
- Role Provisioning and Deprovisioning: Explained

How can I post announcements on the home page?

Use the Announcements page to create, edit, and delete announcements. From the Navigator menu, select **Tools - Announcements**.

- Only the announcement's content (not subject) appears on the home page.
- If social networking features appear instead of announcements, then on the Set System Options page, change
 the home panel settings to display announcements. To open this page from the Navigator menu, select **Tools -**Structure, and then click **Set System Options**.
- What you do on the Announcements page applies immediately to all users, even if you saved your changes while a sandbox is active and not yet published.



Related Topics

- Setting Up Sandboxes: Procedure
- Defining Settings for Home and Navigation: Explained

New Page Creation

Creating Pages for Hosting Third Party Applications: Procedure

Use the Page Integration pages to create pages for hosting third party applications to address needs specific to your organization. All these new pages are grouped under a single category on the Navigator menu and springboard.

Prerequisites

Activate a sandbox.

Creating the First Page

Follow these steps:

- 1. From the Navigator menu, select **Tools Page Integration**.
- 2. Click New Page.
- 3. On the Create Page page, enter a page name.
- 4. In the Category Name field, enter a name for the category to place your page under.
 - Note: You can move your new pages to other categories later using the Structure page.
- 5. Search and select an icon for the page.
- 6. Select the role to whom you want to grant access to the page.
- 7. In the Web Page field:
 - Enter the application URL that you want to host on this page.
 - Alternatively, you can create a secure web page URL:
 - i. Click the Create Secure Web Page URL icon.
 - ii. Select the name of the web application.
 - iii. Enter the destination for the web application.
 - iv. Enter a secure token name.
 - v. Save and close the Create Secure Web Page URL dialog box.

In a secure web page, the application validates the secure token and uses it to authenticate web services within the end-user context. Using this mode of customized access, a partner can directly perform an action or display information to the specific user without any additional authentication.

- 8. To create a partner application for exporting to Oracle Marketplace, do the following:
 - a. Select Partner Application for Oracle Marketplace. Once you create the first page without selecting this check box, you can't create partner applications any more.
 - **b.** Enter the extension name and the extension short name for the partner application. The extension short name must start with an alphabet letter or underscore, and contain only alphanumeric characters and underscores.



▲ Caution: If the extension short name is invalid, you may destabilize the environment.

9. Click Save and Close.

New pages are secure. Your security administrator must assign the privileges shown on the Page Integration Overview page to users such that they can access these pages.

If you have only one page under a category, then that page icon appears at the top level (not under any category) on the springboard. However, such page icons appear under their respective categories on the Navigator menu.

Creating Subsequent Pages

After creating the first page, follow these steps to create more pages:

- 1. On the Page Integration Wizard: New Pages page, click New Page.
- 2. Enter a page name.
- **3.** Follow steps 5 to 7 in the procedure for creating the first page.
- 4. Click Save and Close.

After you have created the first page, all subsequent pages are added in the same category as that of the first page, by default.

Related Topics

- Setting Up Sandboxes: Procedure
- Managing Categories and Page Entries for the Navigator and Springboard: Procedure

Managing Pages Hosting Third Party Applications: Procedure

After creating pages for hosting third party applications, manage them using the options available on the Page Integration Wizard: New Pages page and the Page Integration Wizard: Existing Pages page.

You can do the following actions:

- Edit page settings.
- Add tabs to new and existing pages.
- Edit page tabs.
- Rename Categories.
- Navigate to pages.

Prerequisites

Activate a sandbox.

Editing Page Settings

Follow these steps:

- 1. On the Page Integration Wizard: New Pages page, click the name link for the page that you want to edit.
- 2. On the Edit Page page, make the required changes.
- 3. Click Save and Close.

You can make the following changes to a page:

• Change the icon for the page.



- Change the web page URL that you want to host on this page.
- Add tabs to the page.
- Delete the page.
 - Note: If a page has additional tabs, apart from the one created by default with the page, then you can delete the page only after deleting its tabs.
- Edit the page tabs.

Adding Tabs to New and Existing Pages

When you create a page, a page tab is created by default. You can then add more tabs to your new and existing pages, as required.

To add tabs to new pages, follow these steps:

- 1. On the Page Integration Wizard: New Pages page, click the name link for the page containing the tab that you want to edit.
- 2. On the Edit Page page, click Add Tab.
- 3. On the Create Tab page, enter a tab name.
- **4.** Search and select an icon for the page.
- 5. Select the application role to whom you want to grant access to the page.
- 6. In the Web Page field:
 - Enter the application URL that you want to host on this page.
 - Alternatively, you can create a secure web page URL:
 - i. Click the Create Secure Web Page URL icon.
 - ii. Select the name of the web application.
 - iii. Enter the destination for the web application.
 - iv. Enter a secure token name.
 - v. Save and close the Create Secure Web Page URL dialog box.
- 7. Click Save and Close.

To add tabs to existing pages, follow these steps:

- 1. Click the Page Integration Wizard: Existing Pages icon on the left region of the Page Integration Wizard: New Pages page.
- 2. Click Add Tab to Existing Page.
- 3. In the Select Page dialog box, select a page to add a new tab to.
- **4.** Follow steps 2 to 7 in the procedure for adding tabs to new pages.

Editing Page Tabs

To edit page tabs, follow these steps:

- 1. On the Page Integration Wizard: New Pages page, click the name link for the page containing the tab that you want to edit.
- 2. On the Edit Page page, click the name link for the page tab that you want to edit.
- 3. On the Edit Tab page, make the required changes.
- 4. Click Save and Close.

You can make the following changes to a page tab:

Change the icon for the page.



- Change the web page URL that you want to host on this page tab.
- Delete the page tab.

Renaming Categories

All pages that you create using the Page Integration Wizard: New Pages page are grouped under a single category. To rename the category for all pages, click **Rename Category** on the Page Integration Wizard: New Pages page.

Navigating to Pages

On the Page Integration Wizard: New Pages page, click the icon for the page that you want to navigate to, and view its content.

Related Topics

- Setting Up Sandboxes: Procedure
- Configuring Links for Page Entries: Procedure

User Interface Text Customization

Selecting a Text Customization Tool: Points to Consider

You can modify and replace words or phrases that appear on pages, in messages, and other areas of user interface using several tools or tasks.

Text customization tools include:

- Application Composer
- User Interface Text
- Page Composer

Multiple factors influence the option you select. For example:

- The offering you use
- The extent and scope of your customization
- The components that you customize

This table presents the navigation and offering availability options associated with the tools you can use to customize user interface text.

Tool or Task	Navigation	Offering Availability
Application Composer	In the Navigator, select Tools - Application Composer .	Oracle Sales Cloud
User Interface Text	In the Navigator, select Tools - User Interface Text.	All applications



Tool or Task	Navigation	Offering Availability
Page Composer	Click your user name in the global area, and select Customize <page name=""> Pages</page> under Administration.	All applications
	If the Customize <page name=""> Pages option isn't available under the Administration menu, then select Customize Pages instead.</page>	
Projects Custom Objects (Application Composer)	In the Navigator, select Tools - Projects Custom Objects .	Oracle Project Portfolio Management Cloud

Text Customization Scenarios

The following table includes possible scenarios for customizing user interface text. Compare your situation to the scenario in the table to determine the most appropriate tool for customizing text in your application.

Task	Scope	Tool or Task	Example
Simultaneously replace multiple occurrences of a word or phrase that appear on multiple pages in multiple contexts	Comprehensive. The customization affects multiple pages throughout your application. You can customize the embedded help (for example, hints) using this method.	User Interface Text	Change the word Employee to Associate on every page associated with Employee Self Service, Benefits, and Payroll.
Simultaneously replace multiple occurrences of a word or phrase that appear in parts of messages in the message dictionary	Comprehensive. The customization affects multiple messages throughout your application.	User Interface Text	Change the word Employee to Associate in every message associated with Employee Self Service, Benefits, and Payroll.
Simultaneously replace multiple occurrences of the singular and plural forms of a word or phrase that appear in messages and on pages	Comprehensive. The customization affects multiple pages and multiple messages throughout your application.	User Interface Text	Change the word Employee to Associate and Employees to Associates.
Replace a word or phrase that appears on a specific page	Targeted: A page The customization affects user interface text on a specific page or page fragment.	Page Composer	Change the word Customer to Account on two specific pages.
Replace a word or phrase that appears in a specific message in the message dictionary	Targeted: A message The customization affects part of a specific message in the message dictionary.	Manage Messages task	Change the word Recruit to Potential Employee, but only in two specific messages. All other messages continue to use the word Recruit.
Simultaneously replace a word or phrase associated with a	Targeted: A business object	User Interface Text	In Oracle Sales Cloud, change the label of the opportunity business object, from



Task	Scope	Tool or Task	Example
specific object wherever the object appears	The customization affects a specific component of a specific message in the message dictionary.		Opportunity to Deal. You want the change to affect the business object wherever it appears.
Replace words or phrases that appear in menus and menu items	Targeted: Navigator menu item text	User Interface Text	Change the menu item label from Total Compensation Statements to Compensation Package Statements.

Regardless of the tool you use to make changes, all customizations are written in a single override bundle. Hence, the latest customization overwrites the previous ones.

Note: If you customize using plain text as input, it supersedes any customizations that use the override bundle. For example, if you enter a direct string in Page Composer, then Page Composer writes these customizations to a page customization file (not string resource). Hence, such customizations overwrite the customizations in the override bundle.

Related Topics

Tools for Customizations and Extensions: Critical Choices

Bulk Text Customizations: Explained

Use the User Interface Text tool to simultaneously customize multiple occurrences of entire words or phrases in the user interface (UI).

You can use this tool to do the following activities for bulk text customization:

- Sandbox integration
- Case-sensitive and whole word searches
- Singular and plural text replacement
- Contextual previews
- Note: You must activate a sandbox to use the User Interface Text tool.

To use the User Interface Text tool, from the Navigator menu, select **Tools - User Interface Text**. Then, click **Search and Replace** to search and replace texts in bulk. The User Interface Text tool searches text on pages and in messages in the message dictionary. The search includes user assistance only if the user assistance text is in the message dictionary. The customization functionality for this tool doesn't extend to text in service oriented architecture (SOA) processes.

In the User Interface Text tool, you can:

- Search and replace
- Preview and adjust
- Save and publish



Search and Replace

After you activate a sandbox and click **Search and Replace**, enter the search text and the replacement text. You can enter the singular and plural forms of whole words or phrases. You can also use the following check boxes:

- Match Case To perform case-sensitive searches.
- Match Complete Word or Phrase To search for an exact match of your search text value.
- Note: You can't perform partial word searches, nor can you use wildcard characters as part of the search text.

Use the sample values in this table as a guide when you enter search text.

Search Text	Expected Match	Match?	Reason
Flex	flex	Yes	The application searches for any occurrence of your search string without regard to its position in the strings it searches.
			Unless you select Match Case , all matches are considered exact.
flex	flexfields Flexfields A flexfield is a A Flexfield is a	No	The application treats your search text value as a whole word. The text flex isn't the same as the text flexfields.
^Flex\$	flexfields Flexfields A flexfield is a A Flexfield is a	No	Use *string\$ in the search field to say this string must match the complete field. The application treats your search text value as the entire value of the strings it searches. Alternatively, select Match Complete Word or Phrase
^Flex\$	flex	Yes	to search for an exact match of your search text value.
	Flex		Use ^string\$ in the search field to say this string must match the complete field.
			The application treats your search text value as the entire value of the strings it searches.
			Unless you select Match Case , both matches are considered exact.



Search Text	Expected Match	Match?	Reason
			Use just ^string in the search field to say this phrase must appear at the beginning of a field.
flex credits	Flex Credits Configuration	Yes	The application searches for the
	Allow rollover of flex credits		exact spelling and sequence of words without regard to
	Flex Credits		their position in the strings it searches.
	Manage Flex Credits		
	Configuration		Unless you select Match Case , all matches are considered exact.
flex credits	flex credit shell plan	No	The application searches for the exact spelling and sequence
	Allow rollover of flex credit for		of words without regard to their position in the strings it searches.
credits configuration\$	Flex Credits Configuration	Yes	Use string\$ in the search field
	Manage Flex Credits Configuration		to say this phrase must appear at the end of a field.
			The application searches for the
			exact spelling and sequence of words, where the words appear at the end of the strings it searches.
			Unless you select Match Case , both matches are considered exact.

Include one or more of the following match categories in your search:

- User Interface Text
- Oracle Transactional Business Intelligence Repository Definition Text
- Enterprise Scheduler Text
- Global Menu Label Text
- Multipart Validation Message

After you enter the search text and replacement text, click **Preview Changes**. The tool looks for exact whole word matches.

Preview and Adjust

The preview sorts the search results and presents the matches on tabs based on match categories. Data grids on each tab present the matches in rows. You can adjust each row independently. The grids on each tab are similar, but not identical.



Caution: Some tab names on the Preview Text Changes page are followed by an asterisk (*) (for example, the Messages tab). For these tabs, once you save the preview text changes, these text changes are applied to the mainline metadata, that is, outside the sandbox. So you can't undo your changes after you preview and save them even though you are currently in a sandbox. Also, on destroying the sandbox, these changes remain as is.

Each row on all tabs includes:

- A view of the existing text and the immediately surrounding text for context. You can't edit the existing view.
- A preview of the replacement text and the immediately surrounding text for context. You can edit the preview.
- An option to exclude the row and the specific match you see in the row from the customization.

Each row on the Messages tab includes an indicator. This indicator identifies when the search text appears in a message subcomponent, not necessarily in the message body displayed in the row. You can expand the row and view subordinate rows that display the message subcomponent containing the match and the preview, in context as previously described.

For each row in the preview, you can:

- Do nothing to accept the customization as you see it.
- Select Exclude to eliminate the row from the batch customization and maintain the existing text.
- Modify the replacement text to fine-tune the customization for the specific match in the row. The row remains part of the batch customization, even if the actual customization differs from the other rows.

In the Global Menu Label Text tab, you can't update a secure JSON Web Token (JWT).

If you have multiple languages in your application and you want to make similar text changes in those languages:

- 1. Change your language preference
- 2. Search and replace text
- 3. Validate your changes

Save and Publish

After you review and adjust the matches:

- 1. Save your customizations.
- 2. Thoroughly test the run time pages to make sure that every occurrence of the text is replaced, as you wanted.
- 3. Publish the sandbox.

Note the following points:

- Do not publish a sandbox before you visually inspect and validate all pages and messages that contain text that you customized.
- Users can view:
 - Message, business intelligence, and enterprise scheduler text customizations when you save them, even if you don't publish the sandbox.
 - Page text customizations when you publish the sandbox.

Related Topics

- Using Sandboxes: Explained
- Setting Up Sandboxes: Procedure



Adding Translations of Customized Text: Overview

If you install and use multiple languages in your application and you customize text, then enter translations of the customized text for all languages. You can either add translations at run time, or export strings for offline translations. You can enter custom text translations for existing and newly added strings manually at run time.

You can use several customization tools for updating or adding strings. For example, you can use lookups to add translations at run time. However, you can use the User Interface Text tool to update all strings and enter translations, as well as to perform offline translations.

Related Topics

Translating Flexfield and Value Set Configurations: Explained

Translating Existing Strings at Run Time: Worked Example

This example demonstrates how to translate existing strings manually at run time.

The following table summarizes the key decisions for this scenario.

Decisions to Consider	In this Example
What's the sandbox name that you want to use for translating existing strings?	Sandbox1
What's your base language?	English
What's the existing text that you want to customize?	Page
What's the replacement text that you want to replace the existing text with?	Work area

Entering Custom Text Translations for Existing Strings

- 1. Activate Sandbox1.
- 2. From the Navigator menu, select Tools User Interface Text.
- 3. Click Search and Replace.
- 4. In the Search For field, enter the text, "page".
- 5. In the Replace With field, enter the text, "work area".
- 6. Click **Preview Changes** to preview and adjust the matches, as necessary.

▲ Caution: Some tab names on the Preview Text Changes page are followed by an asterisk (*) (for example, the Messages tab). For these tabs, once you save the preview text changes, these text changes are applied to the mainline metadata, that is, outside the sandbox. So you can't undo your changes after you preview and save them even though you're currently in a sandbox. Also, on destroying the sandbox, these changes remain as is.



- 7. Save your text changes.
 - Note: Repeat steps 4 to 7 for any customizations required in other installed languages.
- 8. Test and verify all messages and pages affected by the text changes. Be sure to test across all applications.
- Publish Sandbox1.Your replacement text for the existing string is now available to all users.

Translating New Strings Added Using Customization Tools: Worked Example

This example demonstrates how to translate new strings that were added using customization tools. While creating strings using customization tools, such as the Structure page, always use the same language, that is, your base language.

The following table summarizes key decisions for this scenario.

Decisions to Consider	In this Example
What's the sandbox name that you want to use for translating newly added strings?	Sandbox2
What's the language you want to translate your newly added English string to?	French
What's the newly created English text that you want to translate in French?	Computer
What's the French replacement text that you want to replace the newly created English text with?	Ordinateur

Entering Custom Text Translations for Newly Added Strings

- 1. Activate Sandbox2.
- 2. From the Navigator menu, select Tools User Interface Text.
- 3. Select French as the language you want to translate your new English string to.
- 4. In the Search For field, enter the newly created English string "computer". You must search in English because the French equivalent has not yet been created.
- 5. Enter the French string "ordinateur" as the replacement text.
- 6. Click **Preview Changes** to preview and adjust the matches, as necessary.

▲ Caution: Some tab names on the Preview Text Changes page are followed by an asterisk (*) (for example, the Messages tab). For these tabs, once you save the preview text changes, these text changes are applied to the mainline metadata, that is, outside the sandbox. So you can't undo your changes after you preview and save them even though you're currently in a sandbox. Also, on destroying the sandbox, these changes remain as is.



- 7. Save your text changes.
- **8.** Test and verify all messages and pages affected by the text changes. While testing, you must sign in with French as the language and use sandbox2.
 - Note: Repeat steps 3 to 8 for every active language.
- 9. Publish Sandbox2.

Your replacement text for the newly added string is now available to all users.

FAQs for User Interface Text Customization

Can I undo customizations that I made using the User Interface Text tool if I haven't published the sandbox?

It depends on types of text customizations in the sandbox. You can undo all text customizations done in the user interface and global menu label by destroying the sandbox before publishing it. However, you can't undo the text customizations done in messages, analyses and reports, and scheduled processes.

Can I get a report of all customized text if I want to analyze, troubleshoot, and diagnose the cause of unexpected action?

No, but you can use Customization Setup Manager to export all your application customizations to a zip file format. You can find the text customizations in files ending in ".xlf.xml". These files list all text customizations done in your application using browser-based tools, such as Application Composer, Page Composer, and Customize User Interface Text. You can use the contents of these files for diagnosis and troubleshooting purposes. These files are read-only, so you can't edit their contents.

Why didn't text in my BI reports and SOA processes change when I used the User Interface Text tool to perform comprehensive text customizations?

The bulk customizations that you perform using the User Interface Text tool affect only the text that appears on application pages and in message dictionary messages.

Theme Customization

Customizing Themes: Overview

Use themes to customize the look and feel of your application. You can change the branding logo, background colors, icon styles, and so on.

You can apply an existing theme to your application pages, or create your own theme and apply it. Use the simple theme editor, that is, the Appearance page to create or edit themes.

Prerequisites for Customizing Themes

Before customizing themes:

1. From the Navigator menu, select **Tools - Appearance**.



- 2. Activate a sandbox. If you're not in an active sandbox, click **Edit** on the Appearance page. You're prompted to activate a sandbox.
 - Note: If you're already in an active sandbox, then the Edit button doesn't appear on the Appearance page.

Once you complete customizing your themes, you can preview and test your changes, and then publish the sandbox to make your changes available to users.

Customizing Themes

Use the Appearance page to create themes and edit custom themes. For example, you can determine the following look and feel aspects of your application pages:

- Logo
- Background image
- Size and style of icons on the springboard
- Style of the cards, which appear on a page in a grid view. These cards present a summary of a single record, with attributes on the front side and optional back side. You can specify whether all cards should display with a darkcolored or a light-colored finish for users.
- Shape of buttons, menus, and tabs
- Colors for the background, global region, headings, page links, and buttons

When you edit your theme using the Appearance page, the themes for both desktop UI and simplified UI are edited simultaneously.

Apart from the editing options available on the Appearance page, if you need advanced controls to edit your themes, then use the advanced theme editor.

▲ Caution: Once you edit a theme using the advanced theme editor, you can't edit that theme using the Appearance page anymore. Also, using the advanced editor, you must separately edit your theme twice - once for the desktop UI and again for the simplified UI.

To edit a theme using the advanced editor, on the Appearance page, select **Advanced** from the Actions menu, and then select either of the following:

- Edit Desktop UI <Theme Name>
- Edit Simplified UI <Theme Name>

Related Topics

Setting Up Sandboxes: Procedure

Creating Themes: Procedure

Create your own custom themes using the simple theme editor, that is, the Appearance page.

Prerequisites

Activate a sandbox.



Creating Themes Using the Simple Theme Editor

Follow these steps:

- **1.** From the Navigator menu, select **Tools Appearance**.
- 2. On the Appearance page, select a theme as your base theme from the Appearance list.
- 3. Make changes to the theme, for example, change the logo, background, style of icons, and heading colors.
- 4. From the Actions menu, select **Save As**.
- 5. On the Save As dialog box, enter a theme name without any spaces.
- 6. Optionally, select Apply this theme.
- 7. Click **OK**. If you have selected the Apply this theme check box, then your theme is saved and set as the current theme. If you haven't selected the check box, your theme isn't applied to the application. However, the theme is saved and set as the current theme.

When you create a theme using the simple editor, two new themes are saved - one for the desktop UI and another for the simplified UI.

Related Topics

Setting Up Sandboxes: Procedure

Managing Themes: Procedure

Use the Appearance page to manage your themes. To open the Appearance page, from the Navigator menu, select **Tools - Appearance**.

You can edit custom themes, apply themes to your application pages, and delete custom themes. You can't edit or delete any predefined theme.

Prerequisites

Activate a sandbox.

Applying Themes

Select a theme from the Theme list, and click **Apply**. If the selected theme is a predefined one, then save it as a new theme, and then edit and apply the theme, as required.

Applying the Default Theme

From the Actions menu, select **Apply Default**. The default theme is applied to your application.

Deleting Themes

From the Theme list, select a custom theme that you want to delete, and then select **Delete** from the Actions menu.

Editing Themes

Follow these steps:

- 1. From the Theme list, select a custom theme that you want to edit.
- Make changes to the theme specifications, for example, change the logo, background image, icon style, and color schemes, such as heading and button colors.
- 3. Click Apply.



Note: Although you can't edit a predefined theme, you can save it as a new theme, and then edit and apply it to your application. Use the **Save As** option from the Actions menu to save a predefined theme as a new theme.

Changing the Logo and Background Watermark

Use the Appearance page to define the following:

- Branding logo, which appears above all application pages. Use an image that's as close to 119 by 25 pixels as possible. In general, an image that's wider than it's tall works best.
- Watermark, which appears in the background of all simplified pages. Use an image that's as close to 1024 by 768 pixels as possible.

You can use any of the following options to select a logo and a background image:

- File: Browse and select a file from your local machine.
- Predefined: Select a file from the list of predefined images.
- **URL**: Enter a full URL for the logo or the watermark.

Editing Themes Using the Advanced Theme Editor

Use the Theme Editor page only if you want to edit your themes using advanced controls, which are not available on the Appearance page. Unlike the simple editor, you can't use the advanced editor to edit themes for the desktop UI and the simplified UI simultaneously. So, to ensure a consistent look and feel of your application across these UIs, you must edit each theme separately for each UI.

▲ Caution: Once you edit a theme using the advanced editor, you can't edit it using the simple editor anymore. You must use only the advanced editor for subsequent edits to that theme.

Follow these steps:

- 1. On the Appearance page, select the custom theme that you want to edit using the advanced editor, from the Theme list
- 2. Click Advanced from the Actions menu, and then select either of the following options:
 - Edit Desktop UI <Theme Name> to edit the theme for the desktop UI
 - Edit Simplified UI <Theme Name> to edit the theme for the simplified UI

The theme opens in the advanced theme editor.

- 3. Edit the theme, for example, make changes to the branding area, buttons, links, and tabs.
- 4. Click Save.

Related Topics

Setting Up Sandboxes: Procedure



Changing the Logo and Color Schemes of the Application: Worked Example

This example demonstrates how to change a company logo and the color schemes of an application using the Appearance page. Users see the logo in the global area.

Note: If the company logo is changed using Page Composer, then this overwrites the changes done using the Appearance page.

The following table summarizes the key decisions for this scenario.

Decisions to Consider	In This Example
What's the name of the new theme?	MyCompany
Which existing theme are you going to base this theme on?	Default
Which image are you going to use as the new logo?	MyCompany.gif
	Tip: The recommended image size of the logo to be used for any theme is 119x25 px.

Prerequisites

Activate a sandbox.

Changing the Logo and Color Schemes

- 1. On the Appearance page, select the Default theme from the Theme list.
- 2. From the Logo list, select **File** as the type of location where your logo is stored.
- 3. Browse for and select MyCompany.gif.
- 4. For the icons, select the style as Flat Dark, the size as Large, and the corner rounding value as 5.
- 5. In the Color Scheme section, enter the values as shown in this table, or select the colors from the color palette.

Field	Value
Background Color	A9A9A9 (that is, Dark Gray)
Global Region Label Color	A52A2A (that is, Brown)
Global Region Background Color	FFFFFF (that is, White)
Heading Color	8B008B (that is, Dark Magenta)
Page Color Link	0000FF (that is, Blue)



Field	Value
Button Colors	Label: A52A2A (that is, Brown)
	Border: 000000 (that is, Black)
	Background: FFA07A (that is, Light Salmon)
Card Style	Light

- 6. From the Actions menu, select Save As.
- 7. On the Save As dialog box, enter the theme name as MyCompany.
- 8. Select Apply this theme.
- 9. Click OK.

The following two new themes are saved in the repository:

- MyCompanyalta for the simplified UI
- MyCompanyskyros for the desktop UI

Related Topics

• Setting Up Sandboxes: Procedure

FAQs for Theme Customization

What happens to my theme if changes that affect themes are made using Page Composer?

Customizations made using Page Composer overwrite the changes made using the simple or the advanced theme editor.

▲ Caution: Before using theme editor to change the look of your application, you must undo any customizations affecting the theme done using Page Composer. Otherwise, the customizations that you make using theme editor may not show up in your application as you wanted.

Infolet Customization

Customizing Infolets: Overview

Use infolets to aggregate key information for a specific area, for example, your sales pipeline, employee turnover, or other business transactions. You can click the navigation dots on the home page to open infolets and view important information at a glance.

If the navigation dots aren't available on the home page, you can enable infolets using the system options on the Structure page.



Customizing Infolets

You can open the infolets page in customization mode and perform the following customization tasks:

- · Create infolets.
- Add content to infolets, for example:
 - You can add a task flow or a performance tile report.
 - o If an infolet contains a performance tile report, then you can add a link to a detailed report in the same infolet.
 - Note: In the context of infolets, report can mean analysis.
- · Edit infolets:
 - o Edit infolet content.
 - o Add, change, or remove link to detailed report.
- Delete infolets

Enabling Infolets

You can enable infolets, that is, display the navigation dots on the home page using Set System Options on the Structure page.

Related Topics

• Personalizing Infolets: Procedure

Creating Infolets

Watch: This video tutorial shows you how to create infolets.

Creating Infolets: Procedure

Use the infolet page to create infolets and set their views.

Prerequisites

You must use the infolet page in customization mode, that is, open the infolet page in Page Composer.

Follow these steps:

- 1. Activate a sandbox.
- 2. Click a navigation dot on the home page to open the infolet page.
- 3. On the infolet page, click your user name in the global area, and select **Customize Pages** under Administration.
- 4. Select a customization layer. The infolet page opens in customization mode.



Note: Once you complete your customizations, click **Close** to leave Page Composer. After testing your changes, you must publish the sandbox to make your changes available to users.

To create an infolet:

- 1. On the infolet page, click the **Infolet Repository** icon, and select **Create Infolet**.
- 2. Enter a title for the infolet.
- 3. Set the infolet views as follows:
 - Specify the dimensions for the front view.
 - o Enable or disable the back view.
 - Enable or disable the expanded view, and specify its dimensions.
- 4. Click Save and Close.

Related Topics

• Setting Up Sandboxes: Procedure

Managing Infolets: Procedure

On the infolet page, use the options available on each infolet to manage it. You can add and edit infolet content, add and edit detailed report links to infolets, and delete infolets. While adding and editing infolet content, you can't add a business intelligence dashboard to an infolet because a dashboard report is generally bigger than an infolet.

Prerequisites

You must use the infolet page in customization mode, that is, open the infolet page in Page Composer.

Follow these steps:

- 1. Activate a sandbox.
- 2. Click a navigation dot on the home page to open the infolet page.
- 3. On the infolet page, click your user name in the global area, and select **Customize Pages** under Administration.
- 4. Select a customization layer. The infolet page opens in customization mode.
- Note: Once you complete your customizations, click **Close** to leave Page Composer. After testing your changes, you must publish the sandbox to make your changes available to users.

Adding Content to Infolets

Follow these steps:

- 1. Click **Add Content** on the infolet. This button is available on an infolet only if the infolet doesn't have any content in it.
- 2. Select a performance tile or a task flow, and click **Add**. You can browse the business intelligence (BI) catalog to find the analytics and reports that you want to add.
- 3. Close the Add Content dialog box.



Editing Infolet Content

You can edit the tile content of an infolet.

Follow these steps:

- 1. Click the **Actions** icon on the top right corner of the infolet, and select **Edit Content**.
- 2. Click **Add Content** to replace the existing content of the infolet.
- 3. Select a performance tile or a task flow, and click **Add**. You can browse the business intelligence (BI) catalog to find the analytics and reports that you want to add.
- 4. Close the Add Content dialog box.

Linking Detailed Reports to Infolets

To provide detailed information about a subject matter on an infolet, add a link to a detailed report. After you add the link, users can click anywhere in the infolet area to drill down to that detailed report. The detailed report doesn't replace the existing infolet content.

Adding Links to Detailed Reports

First, add a performance tile report to the infolet content, and then follow these steps:

- 1. Click the Actions icon on the top right corner of the infolet, and select Link Detailed Report.
- 2. On the Detailed Report page, click **Add Content**.
- 3. Select a report, and click **Add** to add it to the infolet.
- 4. Close the Add Content dialog box.
- 5. Click Done.

Editing Detailed Report Links

To edit a detailed report link in an infolet:

- 1. Click the Actions icon on the top right corner of the infolet, and select Edit Detailed Report.
- 2. On the Detailed Report page, click **Edit Report** and make changes to the detailed report.
 - Tip: To remove the report link, click **Remove Report**.
- 3. Click Done.

Deleting Infolets

To delete an infolet, click the **Actions** icon on the top right corner of the infolet, and select **Delete**.

Related Topics

- Setting Up Sandboxes: Procedure
- Personalizing Infolets: Procedure

FAQs for Infolet Customization

How can I enable infolets?

Follow these steps:

1. Select Tools - Structure from the Navigator menu.



- 2. Click Edit.
- 3. On the Sandbox Required dialog box, click Activate Sandbox.
- 4. Select a sandbox and set it as active.
- 5. On the Structure page, click Set System Options.
- 6. Select Enable Infolets.
- 7. Click Save and Close.
- Note: To make the settings changes available to users, publish the sandbox.

What's the difference between a performance tile report and a detailed report added to the infolets content?

Performance tile report shows data in the small infolet format. When you add a performance tile report to an infolet, users can see only the summary information about the subject matter. But this report doesn't provide detailed information.

To provide detailed information about the subject matter on the same infolet, add a link to a detailed report. Users can click this link to gather more information.

Design Time Page Customization

Customizing Pages Using Design Time Tools: Highlights

Developers can customize pages at design time, for example, to change the color theme for all pages. These changes include complex ones that you need to deploy into the run time environment.

Note: Design time customizations and extensions aren't available in Oracle Cloud implementations.

Design Time Tools

 Use Oracle Application Development Framework (Oracle ADF) Skin Editor to change the look and feel of your application. Refer to the Oracle Fusion Applications Extensibility Guide for Developers.

See: Customizing the Oracle Fusion Applications Skin

- Edit the UI Shell template in Oracle JDeveloper to do customizations that you can't do using Page Composer. Refer
 to the:
 - Oracle Fusion Applications Extensibility Guide for Developers

See: Editing the UI Shell Template

Oracle Fusion Applications Developer's Guide

See: Introduction to Implementing the UI Shell



Customizing the Same Page with Multiple Tools: Points to Consider

Using both Oracle JDeveloper and Page Composer, design time customizations and run time customizations can coexist on a page.

Note: Design time customizations aren't available in Oracle Cloud implementations.

When you use multiple tools to edit the same page, consider these points:

- You can't use both tools at the same customization layer.
- If you customize the same component with both tools, the customization at the higher level layer takes precedence.

For example, suppose you use Page Composer to change a field label from Employee to Associate in the site layer. Meanwhile, someone uses Oracle JDeveloper to change the same label from Employee to Worker in the global layer. The global layer is the base customization layer, as it's only for design time customizations and applies to all users. Your users see Associate, not Worker, because the site layer is at a higher level than the global layer.

- Run time customizations aren't automatically visible in Oracle JDeveloper. To view them, you must:
 - a. Export the customizations from the run time environment to a Java archive (JAR) file.
 - **b.** Import the customizations into the Oracle JDeveloper customization application workspace.

For more information about importing run time customizations into Oracle JDeveloper, see the Oracle Fusion Applications Extensibility Guide for Developers.

Related Topics

Customization Layers: Explained





4 Flexfields for Custom Attributes

Flexfields: Overview

A flexfield is an extensible set of placeholder fields associated with business objects and placed on the application pages. You can use flexfields to extend the business objects and meet enterprise data management requirements without changing the data model or performing any database programming. Flexfields help you to capture different data on the same database table.

For example, an airline manufacturer may require specific attributes for its orders that aren't predefined. Using a flexfield for the order business object, you can create and configure the required attribute.

Flexfields that you see on the application pages are predefined. However, you can configure or extend the flexfields, or modify their properties. Users see these flexfields as field or information attributes on the UI pages. To use flexfields, search for and open the **Define Flexfields** task list in the Setup and Maintenance work area. You can use the following tasks contained within it:

- Manage Descriptive Flexfields: Expand the forms on the application page to accommodate additional information
 that is important and unique to your business. You can use a descriptive flexfield to collect custom invoice details on
 a page displaying invoices.
- Manage Extensible Flexfields: Establish one-to-many data relationships and make application data contextsensitive. The flexfields appear only when the contextual data conditions are fulfilled. Thus, extensible flexfields provide more flexibility than the descriptive flexfields.
- Manage Key Flexfields: Store information combining several values, such as a number combination. The key
 flexfields represent objects such as accounting codes and asset categories.
- Manage Value Sets: Use a group of values to validate the data entered in the flexfields.
 - Note: You can manage value sets within the Manage Descriptive Flexfields or Manage Extensible Flexfields tasks

For more information about specific predefined flexfields, open the Setup and Maintenance work area, and use the tasks in the Define Flexfields task list.

Types of Flexfields

The following three types of flexfields provide a means to customize the applications features without programming:

- Descriptive
- Extensible
- Key

Related Topics

Modules in Application Taxonomy: Explained



Configuring Flexfields: Overview

Configuring a flexfield ranges from identifying the need for extending a business object with custom attributes to integrating the custom attributes into the deployment. In the case of key flexfields, configuring the flexfield involves identifying value set assignments and determining segment structures.

Overall Process for Configuring Custom Attributes

For descriptive and extensible flexfields, the overall configuration process involves the following:

- 1. Use the Highlight Flexfields feature from the Administration menu to find flexfields on pages associated with business objects.
- 2. Plan the flexfield configuration.
- 3. Plan flexfield validation.
- 4. Define the attributes by configuring the flexfield segments.
 - a. Use the Manage Extensible Flexfields or Manage Descriptive Flexfields tasks, or use the Configure Flexfield icon button directly on the page where the flexfield is highlighted. For simple configurations, use the Add Segment, Add Context Value, and Edit Segment icon buttons directly on the page where the flexfield is highlighted.
 - **b.** Optionally, validate the flexfield configuration.
 - c. Optionally, deploy the flexfield to a sandbox for initial testing.
- 5. Deploy the flexfield to the mainline metadata to display the custom attributes on the application pages and to make them available for integration with other tools such as Oracle Business Intelligence.
- 6. Perform the necessary steps to integrate the custom attributes into the technology stack.

A simple configuration is limited to such actions as adding a format-only field or adding a field with a basic list of values.

Overall Process for Configuring Custom Keys

Using key flexfields, you can configure intelligent key codes comprised of meaningful parts according to your business practices. You configure the key flexfield to have one segment for each part that makes up your key code.

For key flexfields, the overall configuration process involves the following:

- 1. Use the Highlight Flexfields feature from the Administration menu to find flexfields on pages associated with business objects.
- 2. Plan the flexfield configuration.
- 3. Plan the flexfield validation.
- 4. Define the value sets before configuring the key flexfield segments by going to the Manage Value Sets task.
- 5. Define the key flexfield structures and their segments, and define structure instances for each structure.
 - a. Use the Manage Key Flexfields task or the Configure Flexfield icon button directly on the page where the flexfield is highlighted.
 - **b.** Optionally, validate the flexfield configuration.
 - c. Optionally, deploy the flexfield to a sandbox for initial testing.



- **6.** Deploy the flexfield to the mainline metadata to display it on the application pages and to make it available for integration with other tools such as Oracle Business Intelligence.
- 7. Perform the necessary steps to integrate the flexfield into the technology stack.

Flexfield Components: Explained

A flexfield is made up of several data entities that store and render information pertaining to flexfield configuration.

Flexfields are made up of the following components:

- Segments
- Value Sets
- Contexts
- Structures

Segments

A segment is a field within a flexfield and represents a single table column of your database. When configuring a flexfield, define the appearance and meaning of individual segments. Segments represent attributes of information. Segments can appear globally wherever the flexfield is implemented, or based on a structure or context. Each segment captures a single atomic value and represents an attribute of information.

The characteristics of a segment vary based on the type of flexfield in which it's used.

- In key flexfields, a segment describes a characteristic of the entity. For example, a part number that contains details about the type, color, and size of an item.
- In a descriptive or extensible flexfield, a segment represents an information attribute on the application page.
 For example, details about a device containing components, some of which are global while the remaining are contextually dependent on the category of the device.

Value Sets

Users enter values into segments while using an application. A value set is a named group of values that validate the content of a flexfield segment. You configure a flexfield segment with a value set to enforce entries of only valid values for that segment.

The configuration involves the following tasks:

- Defining the values in a value set, including characteristics such as the length and format of the values.
- Specifying formatting rules or values from an application table or predefined list.

Multiple segments within a flexfield, or multiple flexfields, can share a single value set.

Contexts

Context-sensitive flexfield segments are available to an application based on a context value. You define contexts as part of configuring a flexfield. Users see global segments as well as any context-sensitive segments that apply to the selected context value.



In descriptive flexfields and extensible flexfields, you can reuse the context-sensitive segments that are based on the database columns, in multiple contexts.

Structures

Key flexfields have structures. Each key flexfield structure is a specific configuration of segments. Adding or removing segments, or rearranging their order, produces a different structure. You can reuse the segments that are based on the database columns, in multiple structures.

Flexfields at Run Time: Explained

Business objects have an associated descriptive or extensible flexfield. Using these, you can create custom attributes for the business object at run time. Some business objects have an associated key flexfield for configuring flexible multiple part keys.

Finding Flexfields on a Page

At run time, the custom attributes you define as flexfield segments appear in the application page just like any other attribute. However, each type of flexfield appears in a different way.

The following characteristics help you determine the type of flexfield on the application page:

- Descriptive flexfield segments appear as label and field pairs or as a table of fields that correspond to the column headers. The fields represent the flexfield segments and accept values that derive from the segment's assigned value set.
- Extensible flexfield segments appear grouped within labeled regions, where each grouping is a context and the region labels are the context names.
- Key flexfields appear in the application page as a field with a key flexfield icon, where the field's value is a collection
 of segments.

To locate flexfields on a page, in the global area, select your user name and under the Settings and Actions menu, select **Highlight Flexfields**. The page renders in a special mode, displaying the location of flexfields, if any, on the page. Do the following:

- Hover over the Information icon to view flexfield details.
- Click the Configure Flexfield icon to manage the flexfield using the Manage Flexfields task.
- Click the Add Context Value, Add Segment, or Edit Segment icons to add a context value or edit a global or context-sensitive flexfield segment. This applies to both descriptive and extensible flexfields.
- Note: Not all flexfields are available for creating custom attributes. For example, some flexfields are protected, and you either can't edit their configurations at all, or can do only limited changes to them. Consult the product-specific documentation in Oracle Fusion Applications Help to verify whether there are any restrictions on using the flexfield.

All segments of a single flexfield are grouped together by default. The layout and positions of the flexfield segments depend on where the application developer places the flexfield on the page. Flexfields may also be presented in a separate section of the page, in a table, or on their own page or a dialog box. You can use Oracle Composer to edit the layout, position, or other display features of the flexfield segments.

When you no longer want to view the flexfields on a page, select **Unhighlight Flexfields** from the Administration menu.



Customizing Flexfields Using Page Composer: Explained

Using Page Composer, you can create customizations to flexfields that are specific to a page.

In Page Composer, to customize:

- Extensible flexfields, open the page in Source view, and look for a region that is bound to an EffContextsPageContainer task flow. This is the container for the extensible flexfield attributes and contexts. To view the flexfield code and identifying information, open the properties panel for the region. To customize any component within the region, select the desired tag and click Edit.
- Descriptive flexfields, open the page in Source view, and look for <descriptiveFlexfield> elements. Open the properties panel for the element to view the flexfield code and identifying information. Within the properties panel, you may customize properties for the global and context-sensitive segments or re-order the segments on the page.

Flexfields and Oracle Applications Cloud Architecture: How They Work Together

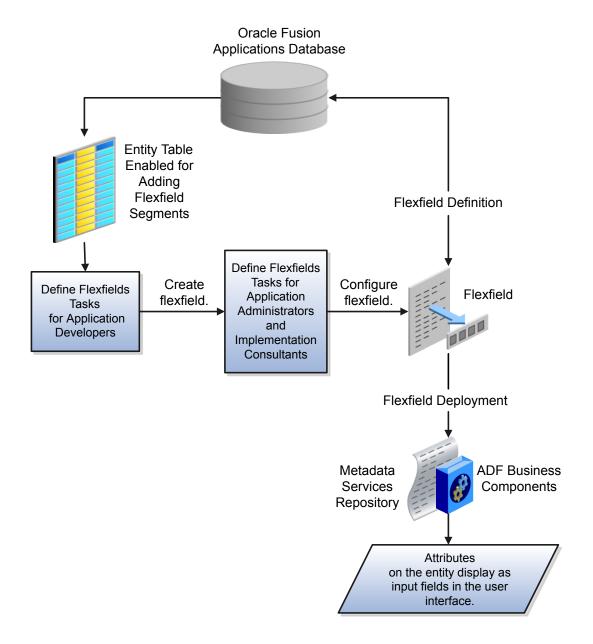
To capture additional data, administrators or implementors configure flexfield segments that represent attributes of business objects. Business objects are enabled for both descriptive flexfields and extensible flexfields.

The following figure shows the layers involved in configuring a flexfield:

- The business entity table and metadata in the database.
- The ADF business component objects. These are derived from the metadata and stored in Oracle Metadata Services (MDS) repository.
- The user interface where fields defined by the flexfield segments are rendered.



The flexfield definition consists of all the metadata defined during configuration and stored in the database.



Application developers create a flexfield and register it so that it's available for configuration. Administrators and implementation consultants configure segments and other properties of the available flexfields. This information is stored as additional flexfield metadata in the database. Deploying the flexfield generates ADF business components based on the flexfield metadata in the database.

The following aspects are important in understanding how flexfields and Oracle Applications Cloud architecture work together:

- Integration
- Deployment



- Import and export
- Run time
- Patching

Integration

The attributes that you add by configuring flexfields are available throughout the Oracle Fusion Middleware technology stack. You can use the flexfield segment's Application Programming Interface (API) to identify segments and integrate the flexfields in the following:

- User interface pages
- Service-oriented Architecture (SOA) infrastructure
- Oracle Business Intelligence
- Extended Spread Sheet Database (ESSbase)

Flexfield configurations are preserved across application updates.

Deployment

The metadata for the flexfield is stored in the application database as soon as you save your configuration changes. Deploying the flexfield generates the ADF business components so that the run time user interface reflects the latest flexfield definition in the metadata.

Importing and Exporting

Using the Setup and Maintenance work area, you can import and export flexfields across the implementation site. The deployment status must be either Deployed or Deployed to sandbox. Therefore, before you attempt migration, verify and ensure that a flexfield is successfully deployed.

Run Time

The latest definitions of a flexfield reflect on the user interface at run time only if the flexfield is deployed. When the user interface accesses a business object, the deployed flexfield definition identifies the attributes associated with the captured values. On a page, if you add display customizations for a flexfield using Oracle Composer, the same flexfield segments can appear differently on different pages.

Patching

Flexfield configurations are preserved during patching and upgrading.

Flexfields and Value Sets: Highlights

Before you use flexfields to create custom attributes, you should be familiar with the customization layers and the customization life cycle of Oracle Applications Cloud. In addition to the extensive help content available about configuring flexfields, consider the resources below for adding flexfields to business components and alternatives to flexfields where flexfields can't be enabled.



For more information about specific predefined flexfields, open the Setup and Maintenance work area, and use the tasks in the Define Flexfields task list. For customization not available through the tasks and user interface pages, contact My Oracle Support at http://www.oracle.com/pls/topic/lookup?ctx=acc=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc=trs if you are hearing impaired.

Note: Don't use Oracle JDeveloper to customize flexfields.

Before Configuring Flexfields

You can add custom attributes to a business object using a flexfield, if a flexfield has been registered for that object by developers.

 For Oracle Sales Cloud, use Application Composer to add custom attributes instead of using descriptive and extensible flexfields.

Deploying Flexfields

- For information about synchronizing the updated XML schema definition (XSD) files in MDS repositories for each SOA application, refer to the Oracle Fusion Applications Extensibility Guide for Developers.
 See: Customizing SOA Composite Applications
 - Oracle ADF services used by SOA composites expose the Web Services Description Language (WSDL) schemas where deployed flexfields are stored.

Oracle Business Intelligence

- For information about importing business intelligence-enabled flexfield changes into the Oracle Business Intelligence repository, refer to the Oracle Transactional Business Intelligence Administrator's Guide.
 - See: Enabling Flexfields for Business Intelligence Reporting
 - See: Importing Changes to Flexfields Automatically

Related Topics

- Exporting and Moving Customizations: Points to Consider
- Defining Fields: Explained

Flexfield Management

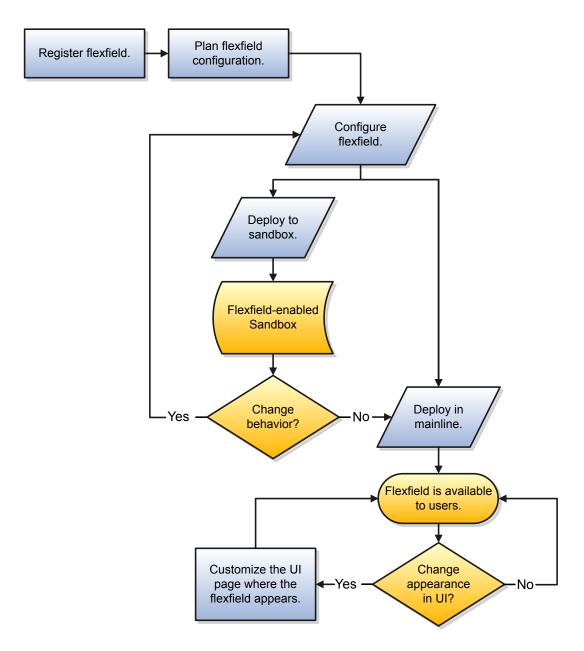
Managing Flexfields: Points to Consider

Managing flexfields involves registering, planning, and configuring flexfields.

You plan and configure the registered flexfields provided in your applications by applications developers. How you configure flexfield segments determines how the flexfield segments appear to users. Optionally, you can customize the UI page to change how the flexfield segments appear to users on that page.



The figure shows the processes involved in making flexfields available to users. The tasks in the Define Flexfields activity let administrators configure and deploy flexfields. After you configure and deploy a flexfield to a sandbox, deploy it again to the mainline metadata so that it's available to the users.



Consider the following aspects of managing flexfields:

- Registering flexfields
- Planning flexfields
- Configuring flexfields
- Enabling a flexfields segment for business intelligence
- Deploying flexfields



- Optionally changing a flexfield segment's appearance in a user interface page
- Identifying flexfields on a run time page and troubleshooting

Registering Flexfields

A flexfield must be registered before it can be configured. Therefore, application development registers flexfields so that they are available to administrators and implementation consultants for configuration. The registration involves reserving columns of entity tables for use in flexfields. For more information about registering flexfields, see Oracle Fusion Applications Developer's Guide.

Planning Flexfields

Before you begin planning flexfields, determine what type is appropriate to your needs, and which business objects are available for customizing flexfields. All flexfields consist of segments which represent attributes of an entity. The value a user enters for an attribute is stored in a column of the entity table. Carefully plan flexfields before configuring them. Before configuring new segments for your flexfields, be sure to plan their implementation carefully.

If you have determined that a business object supports flexfields, and those flexfields have been registered, you can begin planning their configuration. Note the code name of the flexfield you intend to configure so that you can find it easily in the Define Flexfield activity. In some cases you can customize how the flexfield appears on the page. See Oracle Applications Cloud Help for specific products to determine any restrictions on using product-specific flexfields.

Configuring Flexfields

Administrators or implementors configure flexfields so they meet the needs of the enterprise. Some flexfields require configuration to make an application operate correctly. You can configure flexfields using the following methods:

- Go to the manage flexfield tasks in the Setup and Maintenance work area.
- Use the Highlight Flexfields command in the Administration menu while viewing a run time page.
 - Use the **Configure Flexfield** icon button to manage all aspects of a flexfield, such as change a segment's sequence number or configure a flexfield segment's business intelligence label.
 - Use the Add Segment and Edit Segment icon buttons to add and edit descriptive or extensible flexfield segments with simple configurations.
 - Use the Add Context icon button to add descriptive or extensible flexfield context values.

Configuring a flexfield includes the following:

- Defining value sets against which the values entered by users are validated
- Defining the structure or context of the segments in the flexfield
- Specifying the identifying information for each segment
- Specifying the display properties such as prompt, length and data type of each flexfield segment
- Specifying valid values for each segment, and the meaning of each value within the application
- ▼ Tip: You can create value sets while creating descriptive and extensible flexfield segments. However, define value sets before configuring key flexfield segments that use them, because you assign existing value sets while configuring key flexfield segments.

When creating table-validated, independent, dependent, or subset value sets while creating descriptive and extensible flexfield segments, you can optionally specify to display the description of the selected value to the right of the segment at run time. You can assign sequence order numbers to global segments and to context-sensitive segments in each context. Segment display is always in a fixed order based on the segments' sequence numbers. You cannot enter a number for one



segment that is already in use for a different segment. Therefore, you may consider numbering the segments in multiples, such as 4, 5, or 10, to make it easy to insert new attributes.

A flexfield column is assigned to a new segment automatically, but you can change the assignment before saving the segment. If you must set a specific column assignment for a segment, create that segment first to ensure that the intended column isn't automatically assigned to a different segment.

Enabling a Flexfield Segment for Business Intelligence

You can enable flexfield segments for business intelligence if the flexfield is registered in the database as an Oracle Business Intelligence-enabled flexfield. For more information about enabling segments for business intelligence, see points to consider when enabling descriptive, extensible, and key flexfield segments for business intelligence. For extensible flexfield segments, you can't assign labels to equalize segments across contexts that are semantically equivalent.

Deploying Flexfields

Once you have configured a flexfield, you must deploy it to make the latest definition available to run time users. In the Define Flexfields tasks, you can deploy a flexfield using either of the following commands:

- The Deploy Flexfield command deploys a flexfield to the mainline metadata. This command is for general use in a test or production environment.
- The Deploy to Sandbox command deploys a flexfield to sandbox. This command is for confirming that the flexfield is correctly configured before deploying it to the mainline metadata.

In Highlight Flexfields mode, when using the:

- Add Context, Add Segment, and Edit Segment tools for extensible flexfields, use the Save command to save your changes. Then use the Deploy command to deploy the flexfield to the mainline metadata
- Add Segment and Edit Segment tools for descriptive flexfields, use the Save and Deploy command to save your changes. Then deploy the flexfield to the mainline metadata

Once deployed, the deployment status indicates the state of the currently configured flexfield relative to the last deployed definition.

Optionally Changing a Flexfield Segment Appearance

The flexfield attributes that you define integrate with the user interface pages where users access the attributes' business object. Application development determines the UI pages where business objects appear and the display patterns used by default to render flexfield segments.

After a flexfield has been deployed to the mainline MDS repository so that it appears on application pages, you can customize it on a per-page basis using Page Composer. For example, you can hide a segment, change its prompt or other properties, or reorder the custom global attributes so that they are interspersed with the core attributes in the same parent layout. You can customize the appearance of descriptive and extensible flexfield segments in the UI page using Page Composer, once the flexfield is deployed to the mainline metadata.

If the applications are running in different locales, you can provide different translations for translatable text, such as prompts and descriptions. Enter translations using the locale that requires the translated text. In the global area, click your user name and from the **Settings and Actions** menu, select **Set Preferences**. Then change the text to the translated text for that locale.

Identifying Flexfields on a Run Time Page

The **Highlight Flexfields** command in the Administration menu of the Setup and Maintenance work area identifies the location of flexfields on the run time page by displaying an **Information** icon button for accessing details about each flexfield.



Even if a descriptive or extensible flexfield isn't yet deployed and no segments appear on the run time page in normal view, the flexfield appears in the Highlight Flexfield view for that page. For descriptive flexfields, the segments as of the last deployment appear. For extensible flexfields, any segments and contexts that have been saved but not yet deployed also appear as disabled.

Highlight Flexfields accesses the current flexfield metadata definition. Use the highlighted flexfield's **Configure Flexfield** icon button to manage flexfields directly. Alternatively, note a highlighted flexfield's name to search for it in the tasks for managing flexfields.

For more information about creating flexfields and adding them to a UI page, see the Oracle Fusion Applications Developer's Guide. For more information about customizing flexfield segment appearance with Page Composer, see guidance on customizing existing pages in the Oracle Fusion Applications Extensibility Guide.

Flexfield Segment Properties: Explained

Independent of the value set assigned to a segment, segments may have properties that affect how they are displayed and how they function.

The following aspects are important in understanding

- Display properties
- Properties related to segment values
- Properties related to search
- Range validation segments
- Rule validation of segment values
- Naming conventions

Display Properties

The following table summarizes display properties.

Property	Description
Enabled	Whether the segment can be used.
Sequence	The order the segment appears in relation to the other configured segments.
Prompt	The string to be used for the segment's label in the user interface.
Display type	The type of field in which to display the segment.
Selected and deselected values	If the display type is check box, 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 as measured in visible number of lines when the display type is a text area.
Read only	Whether the field should display as read-only, not editable text.



Property	Description
Description help text	The field-level description help text to display for the field. Use description help text to display a field-level description that expands on or clarifies the prompt provided for the field. If description help text is specified, a Help icon button is displayed next to the field in the run time application. The description help text is displayed when the user hovers over the Help icon button.
Instruction help text	The field-level instruction help text to display for the field. Use instruction help text to provide directions on using the field. If instruction help text is specified, it's appears in an in-field help note window when users move the cursor over the field.

Properties Related to Search

Extensible flexfield segments can be marked as selectively required in search using the indexed property. The indexed property requires users to enter a value before conducting a search on the attribute represented by the indexed segment. A database administrator must create an index on the segment column representing the indexed attribute.

Range Validation of Segments

Range validation enables you to enforce an arithmetic inequality between two segments of a flexfield. For example, a product must be ordered before it can be shipped. Therefore, the order date must be on or before the ship date. Also, the order date segment value must be less than or equal to the ship date segment value. You can use range validation to ensure this relationship.

The conditions for range validation are as follows:

- Segments must be configured for range validation in pairs, one with the low value and one with the high value.
- Both segments must be of the same data type.
- Both segments must be parts of the same structure in a key flexfield or parts of the same context in a descriptive flexfield or extensible flexfield.
- The low value segment must have a sequence number that is lesser than that of the high value segment.
- Non-range validated segments can exist between a range validated pair, but range validated pairs cannot overlap or be nested.

You can configure as many range validated pairs as you want within the same flexfield. Your application automatically detects and applies range validation to the segment pairs that you define, in sequence order. It must detect a low value segment first, and the next range validated segment that it detects must be a high value segment. These two segments are assumed to be a matching pair. The low value and the high value can be equal.

Rule Validation of Segment Values

Validation rules on descriptive and extensible flexfield segments determine how an attribute is validated. The value entered for an attribute on a business object may must match a specified format or be restricted to a list of values. Use a value set to specify the validation rules.

Value set validation is required for global segments and context-sensitive segments, and optional for context segments. In the case of context segments, the application may validate a value instead of the value set validating the value against the context segment. However the application entered values must match exactly the valid context segment values. If the context segment values are a superset or subset of the input values, you must assign a table-validated value set or independent value set to validate context values.



When you configure a descriptive flexfield segment, you can specify a constant to use for setting the initial value. The initial value can be an available parameter. For every planned segment, list the constant value or parameter, if any, to use for the initial value.

Naming Conventions

Enter a unique code, name, and description for the segment. These properties are for internal use and not displayed to end users. You can't change the code after the segment is created.

The Application Programming Interface (API) name is a name for the segment that isn't exposed to users. The API name is used to identify the segment in various integration points including web services, rules, and business intelligence. Use alphanumeric characters only with a leading character. For example, enter a code consisting of the characters A-Z, a-z, 0-9 with a non-numeric leading character. The use of spaces, underscores, multi-byte characters, and leading numeric characters isn't permitted. You can't change the API name after the segment has been created.

Flexfields Segments: How They Are Rendered

Flexfield segments appear on pages as attributes of business objects.

Settings That Affect Flexfield Segment Display

When you configure flexfield segments, the value you enter for the segment's display type determines how the segment appears at run time.

How Display Type Values Appear

The following figures represent how the display types render on the UI at run time. Each display type screenshot is assigned an alphabet that maps to the display type and its description in the table.

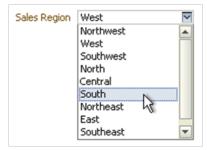


This figure contains the representation of a check box, a drop-down list, a list of values, and a search box.

A. Check Box



B. Drop-down List



C. List of Values

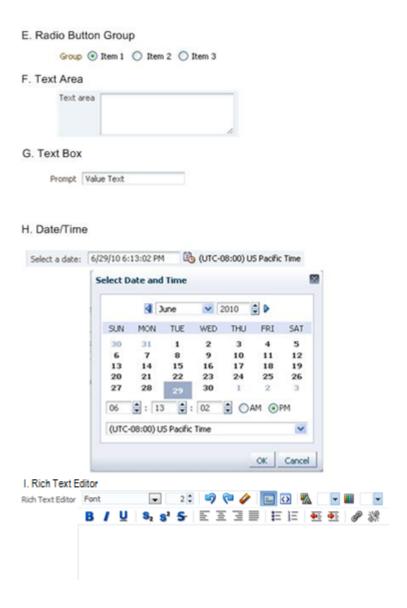


D. Pop-up List of Values



This figure contains the representation of a radio button group, text area, text box, date and time, and rich text editor.





This figure contains the representation of a color palette and a static URL field.



J. Color



K. Static URL



The following table describes each display type.

Display Type	Figure Reference	Description
Check Box	А	The field appears as a check box. If the user selects the check box, the checked value is used. Otherwise, the deselected value is used.
List	В	The field appears as a list of values available to the user for selection.
List of Values	С	The field appears as a list of values available to the user for selection. The user can also click Search to find more values.
Text Field with Search	D	The field appears as a text field with a Search icon button. The users can type a value in the text field or they can click the Search icon button to open another window for searching.



Display Type	Figure Reference	Description
Radio Button Group	E	The field appears as a set of radio buttons. The user can select one button. Selecting a button deselects any previously selected button in the set.
Text Area	F	The field appears as a text area in which the user can type multiple lines of text. The display width and height specify the visible width and number of lines in the text area, respectively.
Text Box	G	The field appears as a text field in which the user can type a single line of text. The display width controls the width of the text box.
Date Time	Н	The field enables the 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 in 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.
Rich Text Editor	I	The field appears as a text area in which the user can enter and edit multiple lines of formatted text. The display width and height specify the visible width and number of lines in the rich text editor, respectively.
		Note: This display type is available for extensible flexfields only.
Color	J	The field displays a color palette for the user to select a color at run time and assign it to the segment. During setup, this display type appears in the list for selection only if:
		 You are working on an extensible flexfield segment. The value set for the segment is set to ORA_FND_COLOR_#RRGGBB.
Static URL	К	The field appears as a text field in which users can enter a fixed URL that opens the web page when clicked.
		Note: The length of the URL must not exceed 255 characters.
Hidden		The field isn't displayed.



Flexfields and Value Sets: How They Work Together

Value sets are specific to your enterprise. When gathering information using flexfields, your enterprise's value sets validate the values that your users enter based on how you defined the value set.

You can assign a value set to any number of flexfield segments in the same or different flexfields. Value set usage information indicates which flexfields use the value set.

The following aspects are important in understanding how flexfields and value sets work together:

- Defining value sets
- Shared value sets
- Deployment

Defining Value Sets

As a key flexfield guideline, define value sets before configuring the flexfield, because you assign value sets to each segment as you configure a flexfield. With descriptive and extensible flexfields, you can define value sets when adding or editing a segment.

Note: Ensure that changes to a shared value set are compatible with all flexfield segments that use the value set.

Shared Value Sets

When you change a value in a shared value set, the change affects the value set for all flexfields that use that value set. The advantage of a shared value set is that a single change propagates to all usages. The drawback is that the change shared across usages may not be appropriate in every case.

Value Set Values

To configure custom attributes to be captured on the value set values screen in the Manage Value Sets task, configure the Value Set Values descriptive flexfield. The object's code is FND_VS_VALUES_B. This flexfield expects the context code to correspond to the value set code. For each value set, you can define a context whose code is the value set code, and whose context-sensitive segments are shown for the values of that value set. By default, the context segment is hidden since it maps to the value set code and is not expected to be changed.

You can also define global segments that are shown for all value sets. However, this would be quite unusual since it would mean that you want to capture that attribute for all values for all value sets.

Deployment

When you deploy a flexfield, the value sets assigned to the segments of the flexfield provide users with the valid values for the attributes represented by the segments.

Defaulting and Deriving Segment Values: Explained

To populate a flexfield segment with a default value when a row is created, specify a default type of constant or parameter and a default value.



To synchronize a segment's value with another field's value whenever it changes, specify the derivation value to be 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 from a parameter, consider making the attribute read-only, as values entered by users are lost whenever the parameter value changes.

When defaulting or deriving a default value from a parameter, only those attributes designated by development as parameters are available to be chosen.

Different combinations of making the segments read only or editable in combination with the default or derivation value or both, have different effects.

Initial run time behavior corresponds to the row for the attribute value being created in the entity table. If the default value is read only, it cannot subsequently be changed through the user interface. If the default value isn't read only, users can modify it. However, if the segment value is a derived value, a user-modified segment value is overwritten when the derivation value changes.

Default Type	Default value specified?	Derivation value specified?	Initial run time behavior	Run time behavior after parameter changes
None	No	Yes	No initial segment value	The changed parameter derivation value updates segment value
Constant	Yes	No	Default segment value	N/A
Constant	Yes	Yes	Default segment value	The changed parameter derivation value updates segment value
Parameter	Yes	No	The default segment value is the parameter's default value	N/A
Parameter	Yes	Yes, and same as default value	The default segment value is the parameter's default and derivation value	The changed parameter derivation value updates segment value
Parameter	Yes	Yes, and different from default value	The default segment value is the parameter's default value	The changed parameter default value doesn't update segment value. Only the changed derivation value updates the segment value.



Flexfield Usages: Explained

The flexfield usage specifies the table with which the flexfield and its segments are associated.

A flexfield can have multiple usages. However, the first table registered for a flexfield indicates the master usage. Segments are based on the master usage. Other usages of the same table for the same flexfield use the same segment setup, though the column names may have a differentiating prefix.

On the Manage Descriptive Flexfields and Manage Extensible Flexfields pages, click the **Show Entity Usages** icon for a specific flexfield to view its entity usage. On the Manage Value Sets page, you can view the flexfield usages for a selected value set.

Extensible Flexfields

For extensible flexfield contexts, you can configure a different usage. The usage of an extensible flexfield context determines the scenarios or user interfaces in which the segments of a context appear to end users. For example, the Supplier page displays an extensible flexfield's supplier usage and the Buyer page for the same flexfield displays the buyer usage. Then, a context that is associated only with the supplier usage appears only on the Supplier page and not on the Buyer page.

Value Sets

The usage of value sets specifies the flexfields having segments where the identified value set is assigned.

FAQs for Flexfield Management

How can I access predefined flexfields?

Search for predefined flexfields using the Define Flexfields task list:

- 1. In the Setup and Maintenance work area, search for the **Define Flexfields** task list and expand it to view the tasks.
- 2. Open the task that corresponds to the flexfields you are searching for.
- **3.** Enter any of the search parameters and click **Search**.
 - Tip: If you don't know the flexfield name or the code, use the **Module** field to filter search results.
- 4. Click a flexfield to view its details.

Why can't I edit my flexfield or value set configuration?

Your flexfield or value set configuration may be protected. Application developers mark some configurations as protected, indicating that you can't edit them.

Some examples of configurations that may be protected are:

- Descriptive flexfields
- Extensible flexfield contexts
- Extensible flexfield pages
- Value sets



Why did my page not display any flexfield?

For a flexfield to be available in the page, it must be registered by developers and also deployed. The segments appear on the page only after you have successfully deployed the flexfield.

A flexfield's deployment status indicates whether the flexfield segments are available to end users. The flexfield segments seen by end users in the run time correspond to the flexfield definition that was last deployed successfully.

For information about registering flexfields, see the Oracle Fusion Applications Developer's Guide. Some business objects haven't been designed to support flexfields. For information about how to enable business objects with flexfield capability, see Getting Started with Flexfields in the Oracle Fusion Applications Developer's Guide.

Note: Oracle Sales Cloud doesn't support flexfields.

To add custom attributes to these applications, you may use Application Composer. For more information, see the product-specific documentation.

Why did my flexfield changes not appear in the run time UI?

The ADF business components or artifacts of a flexfield, which are generated into an Oracle Metadata Services (MDS) Repository when the flexfield is deployed, are cached within a user session. You must sign out and sign back in again to view flexfield definition changes reflected in the run time application user interface page.

How can I enable flexfield segments for Oracle Social Network Cloud Service?

When you manage Oracle Social Network Objects during setup and maintenance, search for the business object that includes descriptive flexfields. Select the attributes that are defined as flexfield segments and enable them.

Flexfield Deployment

Flexfield Deployment: Explained

Deployment generates or refreshes the Application Development Framework (ADF) business component objects that render the flexfield in a user interface. The deployment process adds custom attributes to the Web Services Description Language (WSDL) schemas exposed by Oracle ADF services and used by SOA composites. Flexfields are deployed for the first time during the application provisioning process. After you configure or change a flexfield, you must deploy it to make the latest definition available to users.

If a descriptive flexfield is enabled for business intelligence, the deployment process redeploys the flexfield's business intelligence artifacts.

You can deploy a flexfield to a sandbox for testing or to the mainline metadata for use in a test or production run time environment. You can deploy extensible flexfields as a background process.

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. You must sign out and sign back in to Oracle Applications Cloud to see the changes you deployed in the run time.

The following aspects are important in understanding flexfield deployment:

Deployment Status



- Initial Deployment Status
- Metadata Validations
- Metadata Synchronization
- Deployment as a Background Process
- Export of Artifacts from Flexfield MDS

Deployment Status

Every flexfield has a deployment status.

A flexfield can have the following deployment statuses:

Deployment Status	Meaning
Edited	The flexfield metadata definition hasn't been deployed yet. Updates of the metadata definition aren't applied in the run time environment yet.
Patched	The flexfield metadata definition has been modified through a patch or a data migration action, but the flexfield hasn't yet been deployed. So, the updated definition isn't reflected in the run time environment.
Deployed to Sandbox	The current metadata for the flexfield is deployed in ADF artifacts and available as a flexfield-enabled sandbox. The status of the sandbox is managed by the Manage Sandboxes task available to the Administrator menu of the Setup and Maintenance work area.
Deployed	The current metadata for the flexfield is deployed in ADF artifacts and available to users. No changes have been made to the flexfield after being deployed to the mainline metadata.
Error	The deployment attempt in the mainline metadata failed.

Note: Whenever a value set definition changes, the deployment status of a flexfield that uses that value set changes to edited. If the change results from a patch, the deployment status of the flexfield changes to patched.

Initial Deployment Status of Flexfields

The Oracle Applications Cloud implementation loads flexfield metadata into the database. This initial load sets the flexfield status to Edited. During installation, the application provisioning process deploys the flexfields of the provisioned applications, setting their status to Deployed if no errors occur.

In a provisioned application, deployed flexfields are ready to use. In some cases, flexfield availability at run time requires setup, such as defining key flexfields.

Metadata Validation

Use the Validate Metadata command to view possible metadata errors before attempting to deploy the flexfield. Metadata validation is the initial phase of all flexfield deployment commands. By successfully validating metadata before running the deployment commands, you can avoid failures in the metadata validation phase of a deployment attempt. The deployment process ends if an error occurs during the metadata validation phase. Metadata validation results don't affect the deployment status of a flexfield.



Metadata Synchronization

When an extensible or descriptive flexfield is deployed, the deployment process regenerates the XML schema definition (XSD). As a result, the custom attributes are available to web services and the SOA infrastructure.

After deploying a flexfield configuration, you must synchronize the updated XML schema definition (XSD) files in the MDS repositories for each SOA application.

Note: To synchronize the updated XSD files in the MDS repositories in Oracle Cloud implementations, log a service request using My Oracle Support at http://support.com/

Deployment as a Background Process

You can deploy extensible flexfields offline as a background process and continue working in the session without having to wait for the deployment to complete. You can queue up several extensible flexfields and deploy as a background process. The flexfields are deployed, one at a time, in the order that you deploy them to the queue. You must deploy extensible flexfields with more than 30 categories as a background process.

You can remove an extensible flexfield from the deployment queue with the Cancel Background Deployment command. When an extensible flexfield is deployed in a background process, its offline status indicates that the flexfield is in a background deployment process. A flexfield's offline status is cleared and its deployment status updated when the background deployment process has completed.

Export of Artifacts from Flexfield MDS

You can export business components from MDS for descriptive, extensible, or key flexfields, mainly for use in troubleshooting issues with flexfields. Use **Download Flexfield Archive** on the Manage Flexfields page to export MDS artifacts of the selected flexfield, and import them to an archive on your local computer. You can use these archived business components of flexfields for troubleshooting purposes.

Alternatively, export the deployed artifacts using exportMetadata WLST.

Flexfield Deployment Status: How It Is Calculated

Flexfield deployment status indicates how the flexfield metadata definition in the Oracle Fusion Applications database relates to the Application Development Framework (ADF) business components generated into an Oracle Metadata Services (MDS) Repository.

The following aspects are important in understanding how flexfield deployment status is calculated:

- Settings that affect flexfield deployment status
- How deployment status is calculated

Settings That Affect Flexfield Deployment Status

If you have made a change to a flexfield and expect a changed deployment status, be sure you have saved your changes. No settings affect flexfield deployment status.

How Deployment Status Is Calculated

If the flexfield definition has been edited through the Define Flexfields activity task flows, the status is Edited. The latest flexfield metadata definition in the Oracle Fusion application diverges from the latest deployed flexfield definition. Any change,



including if a value set used in a flexfield changes, changes the deployment status to Edited. If a flexfield has never been deployed, its status is Edited.

Note: When an application is provisioned, the provisioning framework attempts to deploy all flexfields in that application.

If you deploy the flexfield to a sandbox successfully, the status is Deployed to Sandbox. The latest flexfield metadata definition in the Oracle Fusion application matches the metadata definition that generated ADF business components in a sandbox MDS Repository. Whether the sandbox is active or not doesn't affect the deployment status. If the flexfield was deployed to a sandbox and hasn't been edited or redeployed to the mainline metadata since then, the status remains Deployed to Sandbox independent of whether the sandbox is active, or who is viewing the status.

If you deploy the flexfield successfully to the mainline metadata, the status is Deployed. The latest flexfield metadata definition in the Oracle Fusion application matches the metadata definition that generated ADF business components in a mainline MDS Repository. Change notifications are sent when a flexfield is deployed successfully to the mainline metadata.

If either type of deployment fails so that the current flexfield definition isn't deployed, the status is Error. The deployment error message gives details about the error. The latest flexfield metadata definition in the Oracle Fusion application likely diverges from the latest successfully deployed flexfield definition.

If the flexfield definition has been modified by a patch, the status is Patched. The latest flexfield metadata definition in the Oracle Fusion application diverges from the latest deployed flexfield definition. If the flexfield definition was Deployed before the patch and then a patch was applied, the status changes to Patched. If the flexfield definition was Edited before the patch and then a patch was applied, the status will remain at Edited to reflect that there are still changes (outside of the patch) that aren't yet in effect.

When a deployment attempt fails, you can access the Deployment Error Message for details.

Deploying a Flexfield-Enabled Sandbox: How It Works With Mainline Metadata

The flexfield definition in a sandbox corresponds to the flexfield metadata definition in the Oracle Fusion Applications database at the time the flexfield was deployed to the sandbox. When the flexfield is ready for end users, the flexfield must be deployed to the mainline metadata.

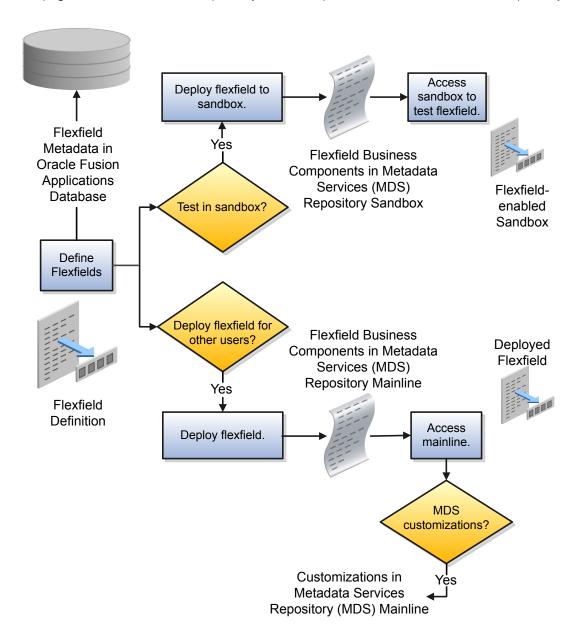
A flexfield-enabled sandbox uses the following components.

- Flexfield metadata in the Oracle Applications Cloud database
- Flexfield business components in a sandbox Oracle Metadata Services (MDS) repository
- User interface customizations for the flexfield in the mainline MDS repository

The figure shows the two types of deployment available in the Manage Flexfield tasks of the Define Flexfields activity. Deploying a flexfield to a sandbox creates a sandbox MDS Repository for the sole purpose of testing flexfield behavior. The sandbox is only accessible to the administrator who activates and accesses it, not to users generally. Deploying a flexfield to the mainline metadata applies the flexfield definition to the mainline MDS Repository where it is available to end users. After



deploying the flexfield to the mainline metadata, customize the page where the flexfield segments appear. Customization of the page in the sandbox MDS Repository cannot be published to the mainline MDS Repository.



Sandbox Metadata Services Repository Data

Deploying the flexfield to a sandbox generates the Application Development Framework (ADF) business components of a flexfield in a sandbox MDS Repository for testing in isolation.

Caution: Don't customize flexfield segment display properties using Page Composer in a flexfield-enabled sandbox as these changes will be lost when deploying the flexfield to the mainline metadata.



Mainline Metadata Services Repository Data

The Oracle Fusion Applications database stores the single source of truth about a flexfield. When the flexfield is deployed, the ADF business component objects that implement the flexfield in the run time user interface are generated in the mainline MDS Repository from this source.

Related Topics

· Managing Customizations Using Sandboxes: Explained

Deploying a Flexfield to a Sandbox: Points to Consider

Deploying a flexfield to a sandbox creates a flexfield-enabled sandbox. Each flexfield-enabled sandbox contains only one flexfield.

You can test the run time behavior of a flexfield in the flexfield-enabled sandbox. If changes are needed, you return to the Define Flexfield tasks to change the flexfield definition.

When you deploy a flexfield to sandbox, the process reads the metadata about the segments from the database, generates flexfield Application Development Framework (ADF) business component artifacts based on that definition, and stores in the sandbox only the generated artifacts derived from the definition.

When you deploy a flexfield sandbox, the process generates the name of the flexfield sandbox, and that flexfield sandbox is set as your current active sandbox. When you next sign in to the application, you can see the updated flexfield configurations. The Oracle Fusion Applications global area displays your current session sandbox.

Note: Unlike a standalone sandbox created using the Manage Sandboxes tool, the sandbox deployed for a flexfield contains only the single flexfield. You can manage flexfield sandboxes, such as setting an existing flexfield sandbox as active or deleting it, using the Manage Sandboxes tool.

When you deploy a flexfield to the mainline metadata after having deployed it to the sandbox, the sandbox-enabled flexfield is automatically deleted.

Sandbox MDS Repository Data

The sandbox data lets you test the flexfield in isolation without first deploying it in the mainline metadata where it could be accessed by users.

▲ Caution: Don't customize flexfield segment display properties using Page Composer in a flexfield-enabled sandbox as these changes will be lost when deploying the flexfield to the mainline metadata.

Managing a Flexfield-Enabled Sandbox

When you deploy a flexfield as a sandbox, that flexfield-enabled sandbox automatically gets activated in your user session. When you sign back in to see the changes, the sandbox is active in your session.

You can only deploy a flexfield to a sandbox using the Define Flexfields task flow pages.

You also can use the Manage Sandboxes feature in the Administration menu of the Setup and Maintenance work area to activate and access a flexfield-enabled sandbox.



Note: Whether you use the Define Flexfields or Manage Sandboxes task flows to access a flexfield-enabled sandbox, you must sign out and sign back in before you can see the changes you deployed in the run time.

You cannot publish the flexfield from the sandbox to the mainline metadata. You must use the Define Flexfields task flow pages to deploy the flexfield for access by users of the mainline metadata because the flexfield configuration in the mainline metadata is the single source of truth.

Related Topics

Managing Customizations Using Sandboxes: Explained

Deploying Flexfields Using the Command Line: Explained

You can use the Manage Key Flexfields, Manage Descriptive Flexfields, and Manage Extensible Flexfields tasks to deploy flexfields. You can also use WebLogic Server Tool (WLST) commands for priming the Oracle Metadata Services (MDS) Repository with predefined flexfield artifacts and for deploying flexfields.

The table describes the available commands.

WebLogic Server Tool Command	Description
deployFlexForApp	Deploys all flexfields for the specified enterprise application. Only flexfields whose status is other than deployed are affected by this command, unless the option is enabled to force all flexfields to be deployed, regardless of deployment status.
	Initial application provisioning runs this command to prime the MDS Repository with flexfield artifacts.
deployFlex	Deploy a single flexfield regardless of deployment status
deployPatchedFlex	Deploys flexfield changes that have been delivered using a flexfield Seed Data Framework (SDF) patch. Deploys flexfields that have a Patched deployment status.
deleteFlexPatchingLabels	Displays MDS label of flexfield changes for viewing and deleting patching labels.
validateFlexDeploymentStatus	Displays list containing flexfields that aren't deployed or failed deployment.

Executing these commands outputs a report at the command line. The report provides the following information for every flexfield that is processed.

- Application identity (APPID)
- Flexfield code
- Deployment result, such as success or error

In case of errors, the report lists the usages for which errors occurred. If a run time exception occurs, the output displays the trace back information. For each WLST flexfield command, adding the reportFormat='xml' argument returns the report as an XML string.



Consider the following aspects of command-line deployment.

- Preparing to use the WLST flexfield commands
- Using the deployFlexForApp command
- Using the deployFlex command
- Using the deployPatchedFlex COmmand
- Using the deleteFlexPatchingLabels Command
- Using the validateFlexDeploymentStatus COMMand
- Closing WLST and checking the results

Preparing To Use the WLST Flexfield Commands

You can only execute the WLST flexfield commands on a WebLogic Administration Server for a domain that has a running instance of Oracle Fusion Middleware Extensions for Oracle Application.

For more information about deploying the Oracle Fusion Middleware Extensions for Oracle Application to the server domains, see the Oracle Fusion Applications Developer's Guide.

Ensure that the AppMasterDB data source is registered as a JDBC data source with the WebLogic Administration Server and points to the same database as the ApplicationDB data source.

Start the WebLogic Server Tool (WLST) if not currently running.

UNIX:

sh \$JDEV HOME/oracle common/common/bin/wlst.sh

Windows:

wlst.cmd

Connect to the server, replacing the user name and password arguments with your WebLogic Server user name and password.

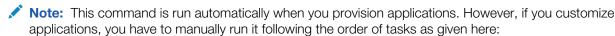
```
connect('wls_username', 'wls_password', 'wls_uri')
```

The values must be wrapped in single-quotes. The wls_uri value is typically T3://localhost:7101.

For more information about the WLST scripting tool, see the Oracle Fusion Middleware Oracle WebLogic Scripting Tool.

Using the deployFlexForApp Command

The deployFlexForApp command translates the product application's predefined flexfield metadata into artifacts in the MDS Repository.



- 1. Configure your application to read the flexfield artifacts from the MDS Repository.
- 2. Run the deployFlexForApp command.
- 3. Sign in to the application.

This sequence of steps is required even if there is no predefined flexfield metadata.



This command doesn't deploy flexfields that have a status of Deployed unless the force parameter is set to 'true' (the default setting is 'false').

For more information about priming the MDS partition with configured flexfield artifacts, see the Oracle Fusion Applications Developer's Guide.

From the WLST tool, execute the following commands to deploy the artifacts to the MDS partition, replacing product application shortname with the application's short name wrapped in single-quotes.

```
deployFlexForApp('product application shortname'[, 'enterprise id'] [,'force'])
```

In a multi-tenant environment, replace <code>enterprise_id</code> with the Enterprise ID to which the flexfield is mapped. Otherwise, replace with <code>'None'</code> or don't provide a second argument.

To deploy all flexfields regardless of their deployment status, set force to 'true' (the default setting is 'false'). To deploy all flexfields in a single-tenant environment, you either can set enterprise id to 'None', or you can use the following signature:

```
deployFlexForApp(applicationShortName='product_application_shortname',force='true')
```

The application's short name is the same as the application's module name. For more information about working with application taxonomy, see the Oracle Fusion Applications Developer's Guide.

Using the deployFlex Command

From the WLST tool, execute the following command to deploy a flexfield, replacing flex_code with the code that identifies the flexfield, and replacing flex_type with the flexfield's type, either descriptive flexfield, key flexfield, or extensible flexfield. The values must be wrapped in single-quotes.

```
deployFlex('flex code', 'flex type')
```

Optionally, execute the following command if the flexfield is an extensible flexfield, and you want to deploy all the flexfield's configurations.

Note: By default, extensible flexfields are partially deployed. That is, only the pages, contexts, or categories that had recent changes, are deployed.

```
deployFlex('flex_code', 'flex_type', ['force_Complete_EFF_Deployment'])
where, forceCompleteEFFDeployment=None
```

Using the deployPatchedFlex Command

Use the deployPatchedFlex command for situations where the patching framework doesn't initiate the command, such as when an application has been patched offline.

If the installation is multi-tenant enabled, the command deploys all patched flexfields for all enterprises. This command isn't intended to be initiated manually.

Check with your provisioning or patching team, or the task flows for managing flexfields, to verify that the flexfield has a Patched deployment status.

From the WLST tool, execute the following command to deploy the artifacts to the MDS partition.

```
deployPatchedFlex()
```

Execute the following command to deploy all flexfields that have either a READY status or an ERROR status.

```
deployPatchedFlex(mode='RETRY')
```



Using the deleteFlexPatchingLabels Command

Whenever you deploy flexfield changes to MDS using the deployPatchedFlex() WLST command, an MDS label is created in the format FlexPatchingWatermarkdate+time. Use the deleteFlexPatchingLabels command to inquire about and delete these labels.

From the WLST tool, execute the deleteFlexPatchingLabels () command with no arguments to delete the flexfield patching labels.

To output a list of flexfield patching labels, execute the command with the infonly argument, as follows:

deleteFlexPatchingLabels(infoOnly='true')

Using the validateFlexDeploymentStatus Command

The validateFlexDeploymentStatus() WLST command checks the deployment status of all flexfields in an Oracle Fusion Applications deployment.

validateFlexDeploymentStatus()

Use this command to verify that all flexfields in the current instance of provisioned Java EE applications are deployed.

Closing WLST and Checking the Results

To close the tool, execute the command: disconnect().

Optionally, sign in the application, open user interface pages that contain flexfields, and confirm the presence of flexfields for which configuration exists, such as value sets, segments, context, or structures.

Manage Value Sets

Value Sets: Explained

A value set is a group of valid values that you assign to a flexfield segment to control the values that are stored for business object attributes.

An end user enters a value for an attribute of a business object while using the application. The flexfield validates the value against the set of valid values that you configured as a value set and assigned to the segment.

For example, you can define a required format, such as a five digit number, or a list of valid values, such as green, red, and blue.

Flexfield segments are usually validated, and typically each segment in a given flexfield uses a different value set. You can assign a single value set to more than one segment, and you can share value sets among different flexfields.

Note: Ensure that changes to a shared value set are compatible with all flexfields segments using the value set.

The following aspects are important in understanding value sets:

- Managing value sets
- Validation
- Security



- Precision and scale
- Usage and deployment
- Protected value set data

Managing Value Sets

To access the Manage Value Sets page, use the Manage Value Sets task, or use the Manage Descriptive Flexfields and Manage Extensible Flexfields tasks for configuring a segment, including its value set. 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.

Validation

The following types of validation are available for value sets:

- Format only, where end users enter data rather than selecting values from a list
- Independent, a list of values consisting of valid values you specify
- · Dependent, a list of values where a valid value derives from the independent value of another segment
- Subset, where the list of values is a subset of the values in an existing independent value set
- Table, where the values derive from a column in an application table and the list of values is limited by a WHERE clause

A segment that uses a format only value set doesn't present a list of valid values to users. Adding table validated value sets to the list of available value sets available for configuration is considered a custom task.

Note: For the Accounting Key Flexfield value sets, you must use independent validation only. If you use other validations, you can't use the full chart of accounts functionality, such as data security, reporting, and account hierarchy integration.

Security

Value set security only works in conjunction with usage within flexfield segments. You can specify that data security be applied to the values in flexfield segments that use a value set. Based on the roles provisioned to users, data security policies determine which values of the flexfield segment end users can view or modify.

The application of value set security has the following conditions:

- At the value set level: The value set is the resource secured by data security policies. If a value set is secured, every usage of it in any flexfield is secured. It isn't possible to disable security for individual usages of the same value set.
- Applies to independent, dependent, or table-validated value sets.
- Applies mainly when data is being created or updated, and to key flexfield combinations tables for query purposes. Value set security doesn't determine which descriptive flexfield data is shown upon querying.
- Security conditions defined on value sets always use table aliases. When filters are used, table aliases are always
 used by default. When predicates are defined for data security conditions, make sure that the predicates also use
 table aliases.

For key flexfields, the attributes in the view object that correspond to the code combination ID (CCID), structure instance number (SIN), and data set number (DSN) cannot be transient. They must exist in the database table. For key flexfields, the SIN segment is the discriminator attribute, and the CCID segment is the common attribute.



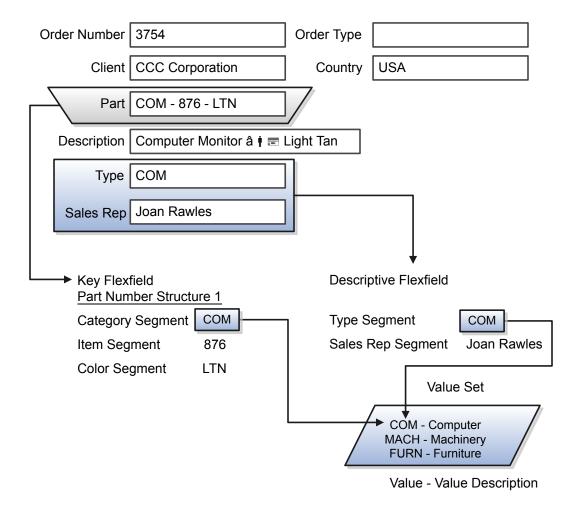
Precision and Scale

If the data type of a value set is Number, you can specify the precision (maximum number of digits user can enter) or scale (maximum number of digits following the decimal point).

Usage and Deployment

The usage of a value set is the flexfields where that value set is used. The deployment status of flexfields in which the value set is used indicates the deployment status of the value set instance.

The figure shows a value set used by a segment in a key flexfield and the context segment of a descriptive flexfield.



For most value sets, when you enter values into a flexfield segment, you can enter only values that already exist in the value set assigned to that segment.

Global and context-sensitive segment require a value set. You can assign a value set to a descriptive flexfield context segment. If you specify only context values, not value sets for contexts, the set of valid values is equal to the set of context values.



Protected Value Set Data

Application developers may mark some value sets as protected, indicating that you can't edit them.

You can edit only value sets that are not marked as protected. You can't edit or delete protected value sets. If the value set type supports values (such as independent, dependent or subset value sets), then you can't add, edit, or delete values.

Note: There is no restriction on references to protected value sets. Value sets, protected or not, may be assigned to any flexfield segment. Likewise, other value sets may reference protected value sets; for example, an unprotected dependent value set may reference a protected independent value set.

Related Topics

- Chart of Accounts: How Its Components Fit Together
- What's the difference between a lookup type and a value set?

Defining Value Sets: Critical Choices

Validation and usage of value sets determine where and how users access valid values for attributes represented by flexfield segments.

Tip: As a flexfield guideline, define value sets before configuring the flexfield, because you can assign value sets to each segment as you configure a flexfield. With descriptive and extensible flexfield segments, you can create value sets when adding or editing a segment on the run time page where the flexfield appears.

The following aspects are important in defining value sets:

- · Value sets for context segments
- Format-only validation
- Interdependent value sets
- Table validation
- Range
- Security
- Testing and maintenance

Value Sets for Context Segments

When assigning a value set to a context segment, you can only use table-validated or independent value sets.

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.

Format Only Validation

The format only validation type enables users to enter any value, as long as it meets your specified formatting rules. The value must not exceed the maximum length you define for your value set, and it must meet any format requirements for that value set.



For example, if the value set permits only numeric characters, users can enter the value 456 (for a value set with maximum length of three or more), but can't enter the value ABC. A format only value set doesn't otherwise restrict the range of different values that users can enter. For numeric values, you can also specify if a numeric value should be zero filled or how may digits should follow the radix separator.

Interdependent Value Sets

Use an independent value set to validate data against a list that isn't stored in an application table, and not dependent on a subset of another independent value set. You cannot specify a dependent value set for a given segment without having first defined an independent value set that you apply to another segment in the same flexfield. Use a dependent value set to limit the list of values for a given segment based on the value that the user has defined for a related independent segment. The available values in a dependent list and the meaning of a given value depend on which value was selected for the independently validated segment.

For example, you could define an independent value set of the states in the USA with values such as CA, NY, and so on. Then you define a dependent value set of cities in the USA with values such as San Francisco and Los Angeles that are valid for the independent value CA. Similarly, New York City and Albany are valid for the independent value NY. In the UI, only the valid cities can be selected for a given state.

Because you define a subset value set from an existing independent value set, you must define the independent value set first. Users don't have to select a value for another segment first to have access to the subset value set.

Independent, dependent, and subset value sets require a customized list of valid values. Use the Manage Values page to create and manage a value set's valid values and the order in which they appear.



💡 Tip: You can customize the Manage Value Sets page to capture additional attributes for each valid value by adding context-sensitive segments in a new context for FND VS VALUES B descriptive field.

Table Validation

Typically, you use a table-validated set when the values you want to use are already maintained in an application table, such as a table of supplier names. Specify the table 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. 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 isn't language-dependent so that the value's invariant ID (an ID that doesn't change) is saved in the transaction table. The run time displays the corresponding translated text from the value column for the run time session's locale.

Table validation lets you enable a segment to depend upon multiple prior segments in the same context structure. You cannot reference other flexfield segments in the table-validated value set's WHERE clause. That is, the WHERE clause cannot reference SEGMENT.segment_code or VALUESET.value_set_code.

Table-validated value sets have unique values across the table, irrespective of bind variables. The WHERE clause fragment of the value set is considered if it doesn't have bind variables. If it has bind variables, the assumption is that the values are unique in the value set. If you use table validated value sets for key flexfields, then you can't use all integration functionalities supported for key flexfields, such as:

- Data security
- Oracle Transactional Business Intelligence (OTBI)
- Extended Spread Sheet Database (ESSbase)
- Tree or hierarchy integration



To use these integration functionalities for key flexfields, you must use independent value sets only.

Range

In the case of format, independent, or dependent value sets, you can specify a range to limit which values are valid. You can specify a range of values that are valid within a value set. You can also specify a range validated pair of segments where one segment represents the low end of the range and another segment represents the high end of the range.

For example, you might specify a range for a format-only value set with format type Number where the user can enter only values between 0 and 100.

Security

In the case of independent and dependent values, you can specify that data security be applied to the values in segments that use a value set. Based on the roles provisioned to users, data security policies determine which values of the flexfield segment users can view or modify.

To enable security on a value set, specify a database resource, typically the code value for the value set. Using the Manage Database Security Policies task, specify conditions, such as filters or SQL predicates, and policies that associate roles with conditions. You can use a filter for simple conditions. For more complex conditions, use a SQL predicate.

Value set data security policies and conditions differ from data security conditions and policies for business objects in the following ways:

- You can grant only read access to 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=70.

When you enable security on table-validated value sets, the security rule that is defined is absolute and not contingent upon the bind variables (if any) that may be used by the WHERE clause of the value set. For example, suppose a table-validated value set has a bind variable to further filter the value list to x, y and z from a list of x, y, z, xx, yy, zz. The data security rule or filter written against the value set must not assume anything about the bind variables. Instead the whole list of values must be available and you write the rule, for example, to permit x, or to permit y and z. By default in data security, all values are denied and show only rows to which access has been provided.

Testing and Maintenance

You don't have to define or maintain values for a table-validated value set, as the values are managed as part of the referenced table or independent value set, respectively.

You cannot manage value sets in a sandbox.

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. In the UI pages for managing value sets, the value set's usages show which flexfields are affected by the value set changes.

If your application has more than one language installed, or there is any possibility that you might install one or more additional languages for your application in the future, select **Translatable**. This doesn't require you to provide translated values now, but you cannot change this option if you decide to provide them later.

Planning Value Sets: Points to Consider

The value sets you create and configure depend on the valid values on the business object attributes that will use the value set. When creating value sets, you first give the value set a name and description, and then define the valid values of the set.



The following aspects are important in planning value sets:

- List of values
- Plain text input
- Value ranges
- Value format specification
- Security

List of Values

You can use one of the following types of lists to specify the valid values for a segment:

- Table column
- · Custom list. Also include a sub list.
- Dependent custom list

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 depending on other segments in the same structure.

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. You can also specify a subset of an existing independent value set as the valid values for a segment. 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.

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 previously selected segment value. 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.

For lists of values type value sets, you can additionally limit the valid values that an end user can select or enter by specifying format, minimum value, and maximum value. For list of values type 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.

Plain Text Input

Use a format-only value set when you want to allow end users to enter any value, as 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 or 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.

Value Ranges

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 Number as the format type where the end user can enter only the values between 0 and 100. Or, you might create a format-only value set with Date as the format type where the end user can enter only dates for a specific year, such as a range of 01-JAN-93 to 31-DEC-93. 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.

Value Format

Flexfield segments commonly require some kind of format specification, regardless of validation type. Before creating a value set, consider how you will specify the required format.



The following table shows options for validation type and value data type.

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

Note: You cannot change the text value data type to a translated text value subtype after creating a value set. If there is any chance you may need to translate displayed values into other languages, choose Translated text. Selecting the Translated text subtype doesn't require you to provide translated values.

Value Sets for Context Segments

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.

Security

When enabling security on a value set, the data security resource name is an existing value set or one that you want to create. The name typically matches the code value for the value set. You cannot edit the data security resource name after you save your changes.

Related Topics

What's the difference between a lookup type and a value set?

Table-Validated Value Sets and Bind Variables: Points to Consider

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

The following bind variables refer to flexfield elements:

: {SEGMENT.<segment_code>}



- :{CONTEXT.<context_code>;SEGMENT.<segment_code>}
- :{VALUESET.<value_set_code>}
- :{FLEXFIELD.<internal code>}
- :{PARAMETER.<parameter code>}

Segment Code

:{SEGMENT.<segment code>}

This bind variable refers to the ID or value of a segment where <segment_code> identifies the segment. Where referring to the ID, the value set is ID-validated. Where referring to the value, the value set isn't ID-validated. The data type of the bind value is the same as the data type of the segment's column.

For both descriptive and extensible flexfields, the segment must be in the same context as the source segment. The source segment contains the WHERE clause. For descriptive flexfields, if the segment is global, then the source segment must be global.

The segment must have a sequence number that is less than the sequence number of the target segment with this bind variable. 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. 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> = :{SEGMENT1}.

Context Code

: {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 must be in the same category or in an ancestor category. It cannot be a multiple-row context.
- The <segment_code> identifies the segment. The data type of the bind value is the same as the data type of the segment's column.
- Note: 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.

The framework of extensible flexfields doesn't perform any additional validation related to mismatched values for segments defined with cross context bind parameters. Administrators must populate the correct pair of segment values.

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.

Value Set Code

:{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.



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

:{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.

Flexfield Parameters

: { 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.

Table-Validated Value Set: Worked Example

In an application user interface, you want to display a list of values that allow customers to enter satisfaction scores. The value column name is 1, 2, 3, 4, 5 and the value column description is Extremely Satisfied, Satisfied, and so on. Users can pick the appropriate value or description which stores the corresponding name so the name value can be used in a calculation expression.

In this case, you can use the FND_LOOKUPS table as the basis for a table-validated value set. The lookup meaning corresponds to the Value Column Name and the lookup description corresponds to the Description Column Name. The properties of the value set are as follows:



Property	Value
FROM clause	FND_LOOKUPS
WHERE clause	lookup_type = 'CN_ XX_ CUST_ SATISFACT_ SCORE'
ID column	lookup_code
Value column	meaning
Description column	description
Enable Flag column	enabled_flag
Start Date column	start_ date_active
End Date column	end_ date_active
Order by	display_ sequence

After completing this task, you should have created your customer satisfaction value set for the Incentive Compensation page of your implementation project.

Creating a Value Set Based on a Lookup

- 1. From the Setup and Maintenance work area, find the Manage Value Sets task and click the **Go to Task** icon button.
- 2. On the Manage Value Sets page, click the **Create** icon button.
- 3. On the Create Value Set page, enter the following values:
 - a. In the Value Set Code field, enter CN_XX_CUSTOMER_SATISFACTION_SCORES
 - **b.** In the Description field, enter Customer satisfaction score.
 - c. In the Module field, select Search....
 - d. In the Search and Select: Module subwindow, enter Incent in the User Module Name field
 - e. Select Incentive Compensation.
 - f. Click OK.
- **4.** On the Create Value Set page, enter the following values:
 - a. In the Validation Type field, select Table.
 - **b.** In the Value Data Type field, select Character.
 - c. In the Definition section FROM Clause field, enter FND_LOOKUPS.
 - d. In the Value Column Name field, enter DESCRIPTION.
 - e. In the Description Column Name field, enter MEANING.
 - f. In the ID Column Name field, enter LOOKUP CODE.
 - g. In the Enabled Flag Column Name field, enter 'Y'.
 - h. In the Start Date Column Name field, enter START DATE ACTIVE.
 - i. In the End Date Column Name field, enter END_DATE_ACTIVE.
 - j. In the WHERE Clause field, enter LOOKUP_TYPE = 'CN_XX_CUST_SATISFACT_SCORE'.
- 5. Click Save and Close.
- **6.** In the Manage Value Sets page, click **Done**.



Adding Attributes to the Manage Value Sets Page: Procedures

You can add attributes to independent, dependent, and subset value sets. The attributes appear on the Manage Value Sets page where you can store additional information about each valid value. To display attributes on an application page, you must programmatically modify the application.

To add attributes and subsequently view them on the Manage Value Sets page, perform the following steps:

- 1. Using the Manage Descriptive Flexfields task, find the FND_VS_VALUES_B flexfield and open it for editing.
- 2. Click Manage Contexts.
- 3. Create a new context and use the value set code for the context code.
- 4. Add new attributes as context-sensitive segments and save the changes.
- **5.** Deploy FND_VS_VALUES_B to run time.
- 6. Sign out and sign back in.
- 7. Open the Manage Value Sets page to view the new attributes.

Importing Value Set Values: Procedure

You can import a file containing values that you want to edit or add to a given independent or dependent value set.

For example, uploading a hundred values may be more efficient than creating them individually using the Manage Value Sets task. However, for just a few values, it may be quicker to perform the relevant tasks.

Importing Value Set Values

To import value set values:

1. Create a flat file containing the values in the value set that you want to add or update.

Note:

- When creating the file, you must specify an existing value set code to which you want to add values
 or edit existing values. If the value set does not exist, add the value set using the appropriate Manage
 Value Sets setup task in the Setup and Maintenance work area.
- The file that you create must adhere to the formatting and content requirements for creating flat files containing value set values.
- 2. Upload the flat file to the content repository using the Files for Import and Export page.
- 3. Import the file using the appropriate Manage Value Sets setup task in the Setup and Maintenance work area. To import the file:
 - **a.** Click **Actions Import** in the Manage Value Sets page.
 - b. In the File Name field, enter the name of the flat file you uploaded using the Files for Import and Export page.
 - c. In the Account field, select the user account containing the flat file.
 - d. Click Upload.
 - Note: Alternatively, you can import the file using either of the following methods:
 - Run the Upload Value Set Values scheduled process.
 - Use the Applications Core Metadata Import web service. For more information on the Applications Core Metadata Import web service, see the SOAP Web Services guide for your cloud services.



Related Topics

Files for Import and Export: Explained

Requirements for Flat Files to Upload Value Set Values: Explained

You can import large volumes of value set value data from the content repository. To upload value set values to the content repository, create a flat file containing the values in the value set that you want to add or update. You upload these flat files to the content repository using the Files for Import and Export page.

General Requirements

The first line of the flat file must contain the column names for the value set value data, including all mandatory columns, and separated by the '|' (pipe) character. Each subsequent line should contain a row of data specified in the same order as the column names, also separated by the '|' character.

The requirements for creating flat files vary with the type of value sets:

- Independent value sets
- Dependent value sets

Independent Value Set

A flat file for uploading values for independent value sets must contain the following mandatory columns:

Column Name	Data Type
ValueSetCode	VARCHAR2(60)
Value	VARCHAR2(150)
Enabled Flag	VARCHAR2(1), Y or N

Note: You can also specify optional columns.

Examples:

To upload values to a COLORS independent value set with the minimum columns, you can use the following flat file:

```
ValueSetCode | Value | EnabledFlag
COLORS | Red | Y
COLORS | Orange | Y
COLORS | Yellow | Y
```

To upload values to a STATES independent value set with more (optional) columns, you can use the following flat file:

```
ValueSetCode | Value | Description | EnabledFlag
STATES | AK | Alaska | Y
STATES | CA | California | Y
STATES | WA | Washington | Y
```



Dependent Value Sets

A flat file for uploading values for dependent value sets must contain the following mandatory columns:

Column Name	Data Type
Value Set Code	VARCHAR2(60)
Independent Value	VARCHAR2(150)
Value	VARCHAR2(150)
Enabled Flag	VARCHAR2(1), Y or N

Note: You can also specify optional columns.

Example:

To upload values to a CITIES dependent value set (dependent on the STATES independent value set), you can use the following flat file:

```
ValueSetCode | IndependentValue | Value | EnabledFlag
CITIES | AK | Juneau | Y
CITIES | AK | Anchorage | Y
CITIES | CA | San Francisco | Y
CITIES | CA | Sacramento | Y
CITIES | CA | Los Angeles | Y
CITIES | CA | Oakland | Y
```

Additional Optional Columns

In addition to the mandatory columns, you can add the following optional columns for both dependent and independent value sets:

Column Name	Туре
Translated Value	VARCHAR2(150), for use in value sets that are translatable
Description	VARCHAR2(240)
Start Date Active	DATE, formatted as YYYY-MM-DD
End Date Active	DATE, formatted as YYYY-MM-DD
Sort Order	NUMBER(18)
Summary Flag	VARCHAR2(30)
Flex Value Attribute1 Flex Value Attribute20	VARCHAR2(30)



Column Name	Туре
Custom Value Attribute1 Custom Value Attribute10	VARCHAR2(30)

Related Topics

Files for Import and Export: Explained

Upload Value Set Values Process

This process uploads a flat file containing value set values for flexfields. You can use this scheduled process to upload a file containing values you want to edit or add to an existing independent or dependent value set. This process is useful for adding or updating large volumes of value set value data in an automated or recurring fashion. For example, you can upload a hundred values on a recurring basis when scheduled as a recurring process. This method may be more efficient than using the one-time Import action in the Manage Value Sets tasks in the Setup and Maintenance work area. However, for an ad hoc task of uploading a hundred values, it may be quicker to use the Import action in the relevant tasks.

Run this process from the Scheduled Processes Overview page. You can run it on a recurring basis whenever the flat file in the content repository account is updated.

You must create the flat file containing the values data, and upload the flat file to the content repository using the Files for Import and Export page.

Parameters

Flat File Name

Enter the name of the flat file you uploaded using the Files for Import and Export page.

Account

Select the user account containing the flat file in the content repository to upload.

Related Topics

Files for Import and Export: Explained

• Scheduled Processes: Explained

Translating Flexfield and Value Set Configurations: Explained

When you first configure a flexfield or segment, the translatable text that you enter, such as prompts and descriptions, is stored as the text for all installed locales. You may then provide a translation for a particular locale. If you don't provide a translation for a given locale, then the value that was first entered is used for that locale.

To translate the text for a particular locale, log in with that locale or specify the locale by selecting **Settings and Actions - Personalization - Set Preferences** in the global area. Then, update the translatable text in the flexfield using the Manage



Descriptive Flexfields task, Manage Key Flexfields task, or Manage Extensible Flexfields task. Your modifications change the translated values only for the current session's locale.

After you complete the translations, deploy the flexfield.

You can define translations for a dependent value set or an independent value set, if it is of type Character with a subtype of Translated text. You define the translations by setting the current session to the locale for which you want to define the translation and using the Manage Value Sets task to enter the translated values and descriptions for that locale.

For a table value set for which the underlying table supports multiple languages and for which the value set's value column is based on a translated attribute of the underlying table, you can define translated values using the maintenance task for the underlying table. For more information on using multilanguage support features, see the Oracle Fusion Applications Developer's Guide.

FAQs for Manage Value Sets

What happens if a value set is security enabled?

Value set security is a feature that enables you to secure access to value set values based on the end user's role in the system.

As an example, suppose you have a value set of US state names. When this value set is used to validate a flexfield segment, and users can select a value for the segment, you can use value set security to restrict them to selecting only a certain state or subset of states based on their assigned roles in the system.

For example, Western-region employees may choose only California, Nevada, Oregon, and so on as valid values. They cannot select non-Western-region states. Eastern-region employees may choose only New York, New Jersey, Virginia, and so on as valid values, but cannot select non-Eastern-region states. Value set security is implemented using Oracle Fusion Applications data security.

How can I set a default value for a flexfield segment?

When you define or edit a flexfield segment, you specify a default value from the values provided by the value set assigned to that segment.

You can set the default value for a descriptive flexfield segment to be a parameter, which means the entity object attribute to which the chosen parameter is mapped provides the initial default value for the segment.

You can set the default value to be a constant, if appropriate to the data type of the value set assigned to the segment.

In addition to an initial default value, you can set a derivation value for updating the attribute's value every time the parameter value changes. The parameter you choose identifies the entity object source attribute. Any changes in the value of the source attribute during run time are reflected in the value of the segment.

If the display type of the segment is a check box, you can set whether the default value of the segment is checked or unchecked.

Manage Descriptive Flexfields



Descriptive Flexfields: Explained

Use descriptive flexfields to add custom attributes to business object entities, and define validation for them.

All the business object entities that you can use in the application are enabled for descriptive flexfields. However, configuring descriptive flexfields is an optional task.

Context

A descriptive flexfield can have only one context segment to provide context sensitivity. The same underlying database column can be used by different segments in different contexts.

For example, you can define a Dimensions context that uses the following attributes:

- ATTRIBUTE1 column for height
- ATTRIBUTE2 column for width
- ATTRIBUTE3 column for depth

You can also define a Measurements context that uses the same columns for other attributes:

- ATTRIBUTE1 column for weight
- ATTRIBUTE2 column for volume
- ATTRIBUTE3 column for density

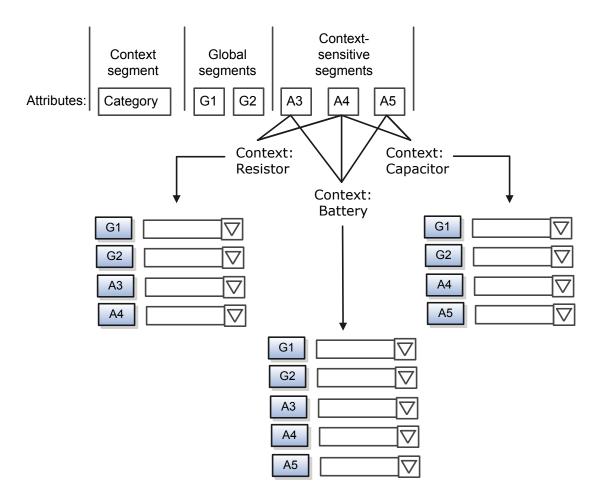
Segments and Contexts

Descriptive flexfield segments are of the following types:

Segment Type	Run Time Appearance
Global segment	Always available
Context segment	Determines which context-sensitive segments are displayed
Context-sensitive segment	Displayed depending on the value of the context segment



In the figure, a descriptive flexfield has one context segment called Category for which there are three values: Resistor, Battery, and Capacitor. Additionally, the descriptive flexfield comprises two global segments that appear in each context, and three context-sensitive segments that only appear in the specific context.



Application development determines the number of segments available for configuring. During implementation, configure the flexfield by determining the following:

- Attributes to add using the available segments
- Context values
- The combination of attributes in each context

Value Sets

For each global and context-sensitive segment, you configure the values permitted for the segment. Based on it, the values that end users enter are validated, including interdependent validation among the segments.

Protected Descriptive Flexfield Data

Application developers may mark some data configurations in a descriptive flexfield as protected, indicating that you can't edit them.



Planning Descriptive Flexfields: Points to Consider

Once you have identified a flexfield to configure, plan the configuration in advance. Compile a list of the UI pages and other artifacts in your deployment that are affected by the configuration. Verify that you are provisioned with the roles needed to view and configure the flexfield. View the flexfield using the Highlight Flexfields command in the Administration menu while viewing the run time page where the flexfield appears. Plan how you will deploy the flexfield for test and production users. Review the tools and tasks available for managing flexfields, such as the Define Flexfields task list, Manage Sandboxes, and Highlight Flexfields for adding and editing flexfield segments.

Planning a descriptive flexfield can involve the following tasks:

- 1. Identify existing parameters.
- 2. Identify existing context values and whether the context value is derived.
- 3. Identify custom attributes and plan the descriptive flexfield segments, segment properties, and structure.
- 4. Plan validation rules.
- 5. Plan initial values.
- 6. Plan attribute mapping to Oracle Business Intelligence objects.

Identify Existing Descriptive Flexfield Parameters

Some descriptive flexfields provide parameters that can be used to specify the initial value of a descriptive flexfield segment. The parameter is external reference data, such as a column value or a session variable. For example, if a flexfield has a user email parameter, you can configure the initial value for a customer email attribute to be derived from that parameter.

Review the list of available parameters in the **Derivation Value** field in the Create Segment page for a descriptive flexfield. If you decide to use one of the parameters to set an initial value, select that parameter from the **Derivation Value** drop-down list when you add the descriptive flexfield segment.

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, access the Edit Descriptive Flexfield task to view the list of configured context values for the flexfield. The Derivation Value field in the Context Segment region displays a list of available parameters. If context values have been preconfigured, see Oracle Applications Cloud Help for product-specific information about the use of those values.

Plan the Segments, Segment Properties, and Structure

Identify the custom attributes you need for a business object to determine the segments of the descriptive flexfield. Determine the segment properties such as the prompt, display type, or initial value.

The structure of the descriptive flexfield is determined by its global, context, and context-sensitive segments. Plan a global segment that captures an attribute for every instance of the business object. Plan a context for segments that depend on a condition of situation applying to a particular instance of the business object. Plan context-sensitive segments to capture attributes that are relevant in the context.

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.



Plan Validation Rules

Define each segment'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 create it either before configuring the flexfield or while creating or editing a segment.

When determining a segment's validation rules, consider the following questions:

- What is the data type character, date, date and time, or number?
- Does the segment require any validation beyond data type and maximum length?
- Should a character type value be restricted to digits, or are alphabetic characters allowed?
- Should alphabetic characters automatically be changed to uppercase?
- Should numeric values be zero-filled?
- How many digits can follow the radix separator of a numeric value? In base ten numerical systems the radix separator is decimal point.
- Does the value need to fall within a range?
- Should the value be selected from a list of valid values? 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 segment?
 - o Is the list of valid values a subset of another flexfield segment's list of values?

Plan Initial Values

For every segment, list the constant value or SQL statement, if any, to use for the initial value of the custom attribute.

Plan How Segments Map to Oracle Business Intelligence Objects

You can extend descriptive flexfields into Oracle Transactional Business Intelligence (OTBI) for ad hoc reporting purposes. Determine the descriptive flexfield segments to be made available for reporting, and select the **BI Enabled** check box accordingly on the Manage Descriptive Flexfields page. You must run a process to extend the BI enabled segments into OTBI. For more information about extending the BI enabled segments into OTBI, see the Setup and Configuration chapter in the Oracle Transactional Business Intelligence Administrator's Guide.

Depending on the reporting needs, you may map similar context-sensitive attributes from different contexts to the same attribute in OTBI. For example, there may be a segment tracking the Product Color attribute in different contexts of a context sensitive descriptive flexfield. You can use segment labels to map these context-sensitive attributes together by defining a segment label and updating the BI Label list accordingly.

Managing Descriptive Flexfields: Points to Consider

Configuring descriptive flexfields involves managing the available flexfields registered with your Oracle Applications Cloud database and configuring their flexfield-level properties, defining and managing descriptive flexfield contexts, and configuring global and context-sensitive segments.

Every descriptive flexfield is registered to include a context segment, which you may choose to use or not.

In general, configuring descriptive flexfields involves:

1. Creating segment labels for business intelligence enabled flexfields.



- 2. Configuring global segments by providing identity information, the initial default value, and the display properties.
- 3. Configuring the context segment by specifying the prompt, whether the context segment should be displayed, and whether a value is required.
- **4.** Configuring contexts by specifying a context code, description, and name for each context value, and adding its context-sensitive segments, each of which is configured to include identifying information, the column assignment, the initial default value, and the display properties.

The following aspects are important in understanding descriptive flexfield management:

- Segments
- Adding segments to highlighted descriptive flexfields
- Usages
- Parameters
- Delimiters
- Initial Values
- Business Intelligence

Segments

You can assign sequence order numbers to global segments and to context-sensitive segments in each context. Segment display is always in a fixed order. You cannot enter a number for one segment that is already in use for a different segment.

Value sets are optional for context segments and follow specific guidelines:

- The value set that you specify for a context segment consists of a set of context codes.
- Each context code corresponds to a context that is appropriate for the descriptive flexfield.
- The value set must be independent or table-validated.
- If table-validated, the WHERE clause must not use the VALUESET.value_set_code or SEGMENT.segment_code bind variables.
- The value set must be of data type Character with the maximum length of values being stored no larger than the context's column length.
- If you don't specify a value set for a context segment, the valid values for that context segment are derived from the
 context codes. The definition of each context segment specifies the set of context-sensitive segments that can be
 presented when that context code is selected by the end user.
- For reasons of data integrity, you cannot delete an existing context. Instead, you can disable the associated context value in its own value set by setting its end date to a date in the past.
- You can configure the individual global segments and context-sensitive segments in a descriptive flexfield. These segment types are differentiated by their usage, but they are configured on application pages that use most of the same properties.

Adding Segments to Highlighted Descriptive Flexfields

When you highlight flexfields on a run time page and use an **Add Segment** icon button to create a segment, the segment code, name, description, table column, and sequence number are set automatically. If you use an **Add Segment** icon button to configure descriptive flexfield segments, you cannot use an existing value set. Value sets are created automatically when you add the segments. You can enter the valid values, their descriptions, and the default value or specify the formatting constraints for the value set, such as minimum and maximum values.

Depending on display type, the value set you create with the **Add Segment** icon button is either an independent value set or a format-only value set. The table shows which type of value set is created depending on the segment display component you select.



Display Component	Value Set Created with Add Segment
Check Box	Independent
Drop-down List	Independent
List of Values	Independent
Radio Button Group	Independent
Text Field With Search	Independent
Text box	Format Only
Text area	Format Only
Date/Time	Format Only

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

Usages

Descriptive flexfield usages allow for the same definition to be applied to multiple entities or application tables, such as a USER table and a USER_HISTORY table. Descriptive flexfield tables define the placeholder entity where the flexfield segment values are stored once you have configured the descriptive flexfield. When you configure a flexfield, the configuration applies to all its usages.

Parameters

Some descriptive flexfields provide parameters, which are attributes of the same or related entity objects. Parameters are public arguments to a descriptive flexfield. Parameters provide outside values in descriptive flexfield validation. You use parameters to set the initial value or derivation value of an attribute from external reference data, such as a column value or a session variable, rather than from user input. Parameters can be referenced by the logic that derives the default segment value, and by table-validated value set WHERE clauses.

Delimiters

A segment delimiter or separator visually separates segment values when the flexfield is displayed as a string of concatenated segments.

Initial Values

The SQL statement defining an initial value must be a valid statement that returns only one row and 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 more information about using bind variables, see the help for value sets.

Business Intelligence

Selecting a global, context, or context-sensitive segment's BI Enabled check box specifies that the segment is available for use in Oracle Business Intelligence.

When the flexfield is imported into Oracle Business Intelligence, the label you selected from the BI Label drop-down list equalizes the segment with segments in other contexts, and maps the segment to the logical object represented by the label.

Enabling Descriptive Flexfield Segments for Business Intelligence: Points to Consider

A descriptive flexfield that is registered in the database as enabled for Oracle Business Intelligence (BI) includes a BI Enabled setting for each of its segments. When a global, context, or context-sensitive segment is BI-enabled, it is available for use in Oracle Business Intelligence.

The following aspects are important in understanding BI-enabled flexfield segments:

- Flattening business components to use BI-enabled segments in Oracle BI
- Equalizing segments to prevent duplication and complexity in the flattened component
- Mapping attributes of flattened business components to logical objects in Oracle BI
- Managing the labels that map segments to logical objects in Oracle BI

After you deploy a business intelligence-enabled flexfield, use the Import Oracle Fusion Data Extensions for Transactional Business Intelligence process to import the flexfield changes into the Oracle Business Intelligence repository. Users can make use of the newly-generated attributes in business intelligence applications. For example, a user can generate a report that includes attributes added by the descriptive flexfield. For additional information about logical objects and import, refer to the Oracle Transactional Business Intelligence Administrator's Guide.

Flattening

When you deploy a business intelligence-enabled descriptive flexfield, the deployment process generates an additional set of flattened Application Development Framework (ADF) business components in addition to the usual ADF business components and ADF faces run time artifacts that are generated during deployment. The flattened business components include attributes for business intelligence-enabled segments only. Flattening means each custom column in each context shows up as an attribute in an Oracle Business Intelligence folder.

Flattened components include one attribute for the BI-enabled context-segment, and one attribute for each business intelligence-enabled global segment. For BI-enabled context-sensitive segments, consider the following:

• If you assigned a label to the segment, the flattened components include an additional single attribute representing segments with that label.



• If you didn't assign a label, the flattened components include a discrete attribute for each BI-enabled contextsensitive segment in each context.

Mapping to Logical Objects in Business Intelligence

You can simplify reporting by representing similar segments as a single logical object in Business Intelligence.

If you assign a label to any set of context-sensitive segments that serve the same purpose in different contexts, you can consolidate or equalize the segments into a single attribute. This prevents duplication and the extra workload and complexity that result from the flattening process. 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.

Non-labeled context-sensitive segments aren't equalized across context values, so the flattened components include a separate attribute for each context-sensitive segment for each context value. It may not be possible to equalize similarly labeled segments if they have incompatible data types or value set types.

Assign a label to a global segment, context segment, or context-sensitive segment to map the corresponding attribute in the flattened components to a logical object in Oracle Business Intelligence. Using labels to map segments to BI logical objects minimizes the steps for importing the flexfield into Oracle Business Intelligence.

Note: Assigning a label to a context-sensitive segment serves to equalize the attribute across contexts, as well as map the equalized attribute to business intelligence.

Managing Labels

You may assign a predefined label (if available) to segments or create new labels for assignment, as needed. Specify a code, name, and description to identify each label. In the BI Object Name field, enter the name of the logical object in Oracle Business Intelligence to which the segment label should map during import. Specifying the BI logical object minimizes the steps for importing the flexfield into Oracle Business Intelligence and helps to equalize context-sensitive segments across contexts.

If no labels are assigned to a BI-enabled segment, or the BI Object Name on the assigned label doesn't exist in business intelligence, you must manually map the segment to the desired logical object when importing into Oracle Business Intelligence.

In addition, context-sensitive segments without labels cannot be equalized across context values. The flattened components include a separate attribute for each non-labeled context-sensitive segment in each context.

Importing to Oracle Business Intelligence Repository

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 flexfield object changes. When you import the metadata into the Oracle Business Intelligence repository, you must do so as the FUSION_APPS_BI_APPID user.

To import flexfield changes into the Oracle Business Intelligence repository in Oracle Cloud implementations, run the Import Oracle Fusion Data Extensions for Transactional Business Intelligence process. For additional information about import, refer to the Oracle Transactional Business Intelligence Administrator's Guide.

Note: When you import a flexfield into the Oracle Business Intelligence repository, you see both <name>_ and <name>_ attributes for each segment, along with some other optional attributes. The <name> attribute contains the value. The <name>_ attribute contains the value set that the value comes from, and is used for linking to the value dimension. You must import both attributes.



Manage Extensible Flexfields

Extensible Flexfields: Explained

Extensible flexfields are like descriptive flexfields, with some additional features.

- You can add as many context-sensitive segments to the flexfield as you need. You aren't restricted by the number of columns predefined and registered for the flexfield.
- You can configure a one-to-many relationship between the entity and its extended attribute rows.
 - o A row of data can have multiple contexts associated with it.
 - A row of data can have multiple occurrences of the same context.
- You can configure attributes in groups to form a context so that the attributes in the context always appear together
 in the user interface.
- You can use existing hierarchical categories so that entities inherit the contexts that are configured for their parents. Contexts are reusable throughout categories.
- Application development has registered some extensible flexfields to support view and edit privileges. For such
 flexfields, you can specify view and edit privileges at the context level to control who sees the attributes and who can
 change the attributes' values.

When you configure a context for multiple rows per entity, the segments are displayed as a table.

Unlike descriptive flexfields, the extension columns corresponding to extensible flexfields segments are part of extension tables, separate from the base application table. Unlike descriptive flexfield contexts, the set of attributes in an extensible flexfield context remains constant and doesn't differ by context value.

An extensible flexfield describes an application entity, with the run time ability to expand the database that implementation consultants can use to define the data structure that appears in the application.

Extensible flexfields support one-to-many relationships between the entity and the extended attribute rows.

To get a list of predefined extensible flexfields, open the Setup and Maintenance work area, and use the Manage Extensible Flexfields task.

The following aspects are important in understanding extensible flexfields:

- Usages
- Categories
- Pages
- Security
- Protected Extensible Flexfield Data

Usages

As with descriptive flexfields, you can define multiple usages for an extensible flexfield, which enables several application tables to share the same flexfield.



For example, a flexfield for shipping options can be used by both a Supplier table and a Buyer table. In addition, you can associate a context with one, some, or all of the flexfield's usages. Thus, with the shipping information example, you can associate a warehouse context with the Supplier usage, a delivery location context with the Buyer usage, and a ship-via context with all usages.

Usages include security information for applying no security to user access or enforcing view and edit privileges. Some product-specific extensible flexfields have specialized usage fields beyond those for security.

Categories

You can configure multiple extensible flexfield contexts and group the contexts into categories. All extensible flexfields have at least one category. For some extensible flexfields, you can configure a hierarchy of categories. A child category in the hierarchy can inherit contexts from its parent category.

You can define categories for extensible flexfields, and you can associate any combination of contexts with a given category.

For example, the Electronics and Computers category hierarchy might include a Home Entertainment category, which in turn might include an Audio category and a TV category, and so on. The Home Entertainment product might have contexts that specify voltage, dimensions, inputs and outputs. Contexts are reusable within a given extensible flexfield. For example, the dimensions context could be assigned to any category that needs to include dimensional information.

Pages

Extensible flexfields let you combine contexts into groups known as pages, which serve to connect the contexts so they will always be presented together in the application user interface.

Each application page corresponds to one extensible flexfield category, with a separate region of the page for each associated context.

Security

When you configure a flexfield, you set the privileges for a context at the usage level by selecting actions for the view and edit privileges of a context usage.

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 don't have privileges.

All extensible flexfields have a base data security resource. Some data security resources for extensible flexfields are preconfigured with actions that you can use to specify access privileges. If no action is preconfigured, a security administrator can create actions and policies to support access control on the extensible flexfield attributes.

Some extensible flexfields have a translatable option; these flexfields also have a translation data security resource.

Protected Extensible Flexfield Data

Application developers may mark some data configurations in an extensible flexfield as protected, indicating that you can't edit them.

If an extensible flexfield is partially protected, then you can't edit the protected portions of the flexfield's configuration. For example:

- If an extensible flexfield context is protected, you can't edit its:
 - Context details



- Context segments
- Context usages
- If an extensible flexfield page is protected, you can't:
 - Edit the page details or delete the page
 - Edit the contexts associated with the page

Note:

- There is no restriction on page references to protected contexts. Custom pages you create may contain any context, whether protected or not.
- There is a restriction on category references to protected contexts. If a context is protected, you can't add it to
 or delete it from any category.

Planning Extensible Flexfields: Points to Consider

Once you have identified a flexfield to configure, plan the configuration in advance. Compile a list of the UI pages and other artifacts in your deployment that are affected by the configuration. Verify that you are provisioned with the roles needed to view and configure the flexfield. View the flexfield using the Highlight Flexfields command in the Administration menu while viewing the run time page where the flexfield appears. Plan how you will deploy the flexfield for test and production users. Review the tools and tasks available for managing flexfields, such as the Define Flexfields task list, Manage Sandboxes, and Highlight Flexfields for adding and editing flexfield segments.

Planning an extensible flexfield can involve the following tasks:

- 1. Identify a hierarchical structure of categories.
- 2. Identify existing context values.
- 3. Identify custom attributes and plan the extensible flexfield segments, segment properties, and structure.
- 4. Plan validation rules.
- 5. Plan initial values.
- 6. Plan security.
- 7. Plan attribute mapping to Oracle Business Intelligence objects.

Category Hierarchy Structure

Existing category hierarchy structures provide the framework for planning what segments to add to an extensible flexfield as custom attributes of an entity.

Some Oracle Fusion applications provide user interfaces to create and manage an extensible flexfield's category hierarchy.

Contexts and Existing Context Values

If related custom attributes can be grouped together, plan adding the attributes as a context of segments, and plan the order in which the attributes should appear.

Some extensible flexfields have preconfigured context values. Region headers displayed in a the user interface page or pages that contain the flexfield segments identify existing contexts. Using the Manage Extensible Flexfields task, find and open the flexfield for editing to view the list of configured context values.

See product-specific information for guidance in using preconfigured context values.



Plan the Segments and Segment Properties

List all the custom attributes that you want to add as extensible flexfield segments.

For each segment, define properties, including the indexed property.

Plan Validation Rules

Define each segment'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 create it either before you configure the flexfield or at the same time that you create or edit a segment.

When determining a segment's validation rules, consider the following questions:

- What is the data type character, date, date and time, or number?
- Does the segment require any validation beyond data type and maximum length?
- Should a character type value be restricted to digits, or are alphabetic characters allowed?
- Should alphabetic characters automatically be changed to uppercase?
- Should numeric values be zero-filled?
- How many digits can follow the radix separator of a numeric value? In base ten numerical systems the radix separator is decimal point.
- Does the value need to fall within a range?
- Should the value be selected from a list of valid values? 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 segment?
 - o Is the list of valid values a subset of another flexfield segment's list of values?

Plan Initial Values

For every segment, list the constant value or SQL statement, if any, to use for the initial value of the custom attribute.

Plan Security

Determine what privileges to set for view and edit access to context attributes, such as providing all end users with view access but only managers with edit access.

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, such as needing to secure each context with a different privilege, then you can create more fine-grained actions.

Plan Which Segments Map to Oracle Business Intelligence Objects

If an extensible flexfield has been enabled for Oracle Business Intelligence, you can make the attributes available for use in Oracle Business Intelligence applications.



Managing Extensible Flexfields: Points to Consider

Configuring extensible flexfields involves managing the available flexfields registered with your application database.

The following sequence describes how to configure extensible flexfields:

- 1. Configuring contexts by creating each context segment and the context-sensitive segments for each context segment, and providing the following for each segments:
 - a. Identifying information
 - b. Column assignment
 - c. Initial default value
 - **d.** Display properties
- 2. Configuring context usages and usage security by selecting actions to which users should have access:
 - View
 - Edit
 - o None, if no special privileges should be enforced.
- 3. Configuring categories and category details.
- 4. Associating contexts with a category.
- 5. Creating logical pages for a category.

The following aspects are important in understanding extensible flexfield management:

- Contexts and pages
- Categories
- Initial values
- · Adding segments to highlighted extensible flexfields
- Indexed segments
- Security
- Deployment

Contexts and Pages

Each context is displayed to end users as a region containing its context-sensitive segments. You can specify instruction help text to display instructions that explain how to use the region and its attributes to end users. Instruction help text is displayed at the top of the context region. A context can be defined as single row or multi row. Single row contexts are the same as descriptive flexfields contexts. A single row context has only one set of context-sensitive segments. A multi-row context enables you to associate multiple sets of values with the same object instance.

For example, for a BOOK table, you could create a multi row context named chapters that contains a segment for chapter and a segment for number of pages. Multiple chapters can then be associated with each book in the BOOK table.

For contexts that store multiple rows, you can uniquely identify each row by having the values in each row form a unique key.

If flexfield has a category hierarchy, then you can leverage the hierarchy to reuse contexts for similar entities, such as similar items in a product catalog.

Set the context to translatable so that free-form text entered by end users is stored in the language of the user's locale, and different translations of that text can be stored in other languages. Segments in the translated contexts should utilize formatonly value sets for storing free-form, user-entered text.



Set the context security to give an end user view or edit access to a context. The context's task flow and region appear in the user interface only for users with view access. With edit access, an end user can edit the context's attribute values. With no action specified for a usage, no special privileges are enforced through the context's configuration.

Define logical pages to group contexts together in the user interface. For a given category, you may create one or more logical pages. You may add one or more of the category's associated contexts to each of the category's logical pages.

You can specify:

- The sequence of the contexts within each page.
- The sequence in which the logical pages appear.
- Instruction help text to display instructions that explain how to use the page to end users. Instruction help text is displayed at the top of the logical page, preceding all of its context regions.

Categories

A category is a grouping of related data items that can be considered to belong together. You can associate any combination of contexts with a given category. Extensible flexfields with more than 30 categories must be deployed as a background process.

A category hierarchy logically organizes a set of categories. For example, the Electronics and Computers category hierarchy might include a Computer category and a Home Entertainment category, which in turn might include an Audio category and a TV category, and so on.

A category can be a child or sibling of an existing category. The hierarchy can be as simple or as complex as desired, with any combination of zero or more sibling categories and zero or more child categories. If no category is defined, the data items are grouped under a single predefined default category.

Each category has associated contexts that store relevant information about a data item in that category. For example, a Home Entertainment product has contexts that specify Voltage, Dimensions, Inputs and Outputs. Contexts are reusable within a given extensible flexfield. Then, the Dimensions context could be assigned to any category that needs to include dimensional information.

If a hierarchy includes child categories, each child category inherits the contexts from its parent category; for example, the Home Entertainment category inherits Voltage and Dimensions from the Electronics and Computers category.

Each extensible flexfield is associated with a particular category hierarchy. Consider category hierarchies to be defining framework for extensible flexfields and their contexts. A category hierarchy specifies which contexts are valid for each category.

An extensible flexfield can include multiple contexts which you define to support a given category. These contexts can be suitable for various purposes, but within a particular category, some contexts might be considered to be related to, or dependent on, each other. You can combine these contexts into groups known as logical pages, and determine the sequence in which the pages appear. This serves to connect the contexts so they will always be presented together and in a particular order in the application user interface.

For example, the Home Entertainment category might have an Electrical Specifications page that contains the Voltage, Inputs and Outputs contexts, and a Physical Specifications page that contains the Dimensions and Form Factor contexts.

Initial Values

The SQL statement defining an initial value must be a valid statement that returns only one row and 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 more information about using bind variables, see the help for value sets.

Adding Segments to Highlighted Extensible Flexfields

When you highlight flexfields on a run time page and use an **Add Segment** icon button to create a segment, the segment code, name, description, table column, and sequence number are set automatically. If you use an **Add Segment** icon button to configure extensible flexfield segments, you can't use an existing value set. Value sets are created automatically when you add segments. You can enter the valid values, their descriptions, and the default value or specify the formatting constraints for the value set, such as minimum and maximum values.

Depending on display type, the value set you create with the **Add Segment** icon button is either an independent value set or a format-only value set. The table shows which type of value set is created depending on the segment display component you select.

Display Component	Value Set Created with Add Segment
Check Box	Independent
Drop-down List	Independent
List of Values	Independent
Radio Button Group	Independent
Text Field With Search	Independent
Text box	Format Only
Text area	Format Only
Rich Text Editor	Format Only
Date/Time	Format Only

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



Indexed Segments

You can designate an extensible flexfield segment as indexed so that it's one of the selectively required attributes a user can use in an attribute search. If you indicate in the Manage Extensible Flexfield UI page that a segment should be indexed, the column representing the segment must be added to the database index. Commonly, a database administrator (DBA) adds columns to the database index.

When an extensible flexfield with indexed segments is deployed, search task flows are generated along with the other flexfield artifacts and specify the indexed attributes as selectively required. In the deployed extensible flexfield's search task flow, an end user must specify at least one of the indexed attributes in the search criteria. This prevents non-selective searches, which could cause performance issues.

For example, if you index the memory and processor attributes and ensure that the corresponding columns in the database are indexed, a user can search an item catalog for computers by entering processor or memory or both as a search criteria. No search is performed if an end user enters an attribute that isn't indexed as a search criterion.

Security

An extensible flexfield's base data security resource typically has a name with an _B suffix. The translation data security resource is a view of a translation table that typically has a name with an _VL suffix.

If a flexfield supports the translatable option and has a translation data security resource, make sure 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.

Deployment

You can only deploy extensible flexfields using the Manage Extensible Flexfields task. You can deploy extensible flexfields offline as a background process and continue working in the session without having to wait for the deployment to complete. You can queue up several extensible flexfields and deploy as a background process. The flexfields are deployed, one at a time, in the order that you deploy them to the queue. You must deploy extensible flexfields with more than 30 categories as a background process.

You can remove an extensible flexfield from the deployment queue with the Cancel Background Deployment command. When an extensible flexfield is deployed in a background process, its offline status indicates that the flexfield is in a background deployment process. A flexfield's offline status is cleared and it's deployment status updated when the background deployment process has completed.

Note: The Offline Status column refreshes when you perform a new search in the Manage Extensible Flexfields task.

Enabling Extensible Flexfield Segments for Business Intelligence: Points to Consider

An extensible flexfield that is registered in the database as enabled for Oracle Business Intelligence (BI) includes a BI Enabled setting for each of its segment instances. When a segment instance is BI-enabled, it's available for use in Oracle Business Intelligence.

The following aspects are important in understanding BI-enabled extensible flexfield segments.

Flattening business components to use BI-enabled segments in Oracle BI



Mapping attributes of flattened business components to logical objects in Oracle BI

After you deploy a business intelligence-enabled flexfield, use the Import Oracle Fusion Data Extensions for Transactional Business Intelligence process to import the flexfield changes into the Oracle Business Intelligence repository. Users can make use of the newly-generated attributes in business intelligence applications. For additional information about logical objects and import, refer to the Oracle Transactional Business Intelligence Administrator's Guide.

Flattening

When you deploy a business intelligence-enabled extensible flexfield, the deployment process generates an additional set of flattened business components for use in business intelligence. The flattened business components include attributes for business intelligence-enabled segment instances only.

If you assigned a label to a segment, the flattened components include a single attribute representing all segment instances with that label. If you didn't assign a label, the flattened components include a discrete attribute for each BI-enabled segment instance in each structure.

Importing to Oracle Business Intelligence Repository

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 flexfield object changes. When you import the metadata into the Oracle Business Intelligence repository, you must do so as the FUSION_APPS_BI_APPID user. To import flexfield changes into the Oracle Business Intelligence repository in Oracle Cloud implementations, run the Import Oracle Fusion Data Extensions for Transactional Business Intelligence process. For additional information about import, refer to the Oracle Transactional Business Intelligence Administrator's Guide.

▼ Tip: When you import a flexfield into the Oracle Business Intelligence repository, you see both <name>_ and <name>_ attributes for each segment, along with some other optional attributes. The <name>_ attribute contains the value. The <name>_ attribute contains the code of the value set that the value comes from, and is used for linking to the value dimension. You must import both attributes.

Managing Extensible Flexfield Categories: Points to Consider

Categories are a way of extending the number of context-sensitive segments for a flexfield beyond the columns reserved for flexfield segments.

An Items extensible flexfield has a category for each item and each category can have one or more contexts. The laptop item belongs to the Computers category. Since extensible flexfields are mapped to separate extension tables, not just to columns as with descriptive flexfields, the thirty reserved columns on the extensible flexfield table let you define up to thirty context-sensitive segments for each context.

If you add a Dimensions context to the Computers category, thirty segments are available. But if you need to add more than thirty attributes, create another context and associate it to the same category. You could now add an Electronics Attributes context to the same Computers category in which you create another thirty segments.

You can continue creating more contexts and adding them to the Computers category. In this way your laptop computer item can be extended with as many attributes as you need, because it is mapped to a category and you can keep adding contexts to that category.

A descriptive flexfield on an items table with thirty columns reserved for segments can only have a single context. Once you configure the columns for that one context, you cannot create any more segments.



Predefined and Preconfigured Categories

How you structure the flexfield configuration depends on how categories are defined for the flexfield. If the extensible flexfield is preconfigured with one category, associate all your contexts and pages with that category. If a product-specific extensible flexfield is preconfigured with several categories, associate your contexts and pages with those categories. If the extensible flexfields provide user interfaces for configuring multiple categories, associate a context with more than one category using inheritance.

Some products provide an activity or task for creating and maintaining categories for an extensible flexfield. See product-specific information to determine if you can create categories for the flexfield.

You can view a flexfield's category hierarchies by using either the Highlight Flexfields feature or the Manage Extensible Flexfields task to find and open the flexfield for editing.

Disabling Categories

While configuring an extensible flexfield, you can disable a category. The Enabled column in the Category table of the Edit Extensible Flexfield page, indicates which categories are enabled.

Note: When you deploy an extensible flexfield that has a disabled category, that category and its descendant categories aren't deployed. Contexts and their segments are deployed only if they belong to at least one enabled category.

Contexts

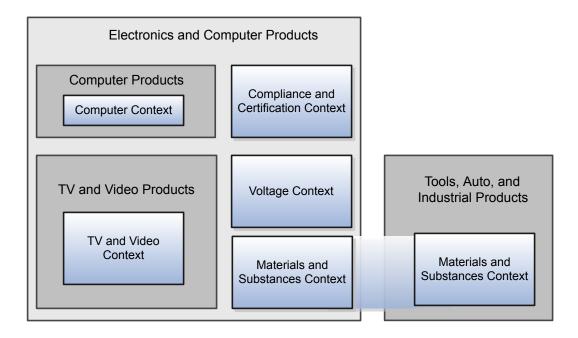
Group similar custom attributes into contexts. The group is displayed together in a region. The region's header is the context value

If a category hierarchy exists for the flexfield, then you can leverage the hierarchy to reuse contexts for similar entities, such as similar items in a product catalog.

The figure shows the Item Extended Attributes flexfield, which uses the category hierarchy feature to reuse contexts. The flexfield's Electronics and Computers category contains contexts for compliance and certification, voltage, and materials and substances. The TV and Video subcategory and the Computer Products subcategory inherit the Electronics and Computer



contexts in addition to having their own contexts. The Materials and Substances context belongs to both the Electronics and Computer Products category and the Tools, Auto, and Industrial Products category.



The table shows an example of category hierarchy for an extensible flexfield.

Display Name	Code	Description
Electronics and Computers	PROD_ ELECTRONICS	Electronics and Computers
TV and Video	PROD_TV_VIDEO	Television 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 store voltage information for all electronic and computer items, associate a Voltage context with the Electronics and Computers category. Both the TV and Video subcategory and the Computers subcategory then inherit the Voltage context from the parent Electronics and Computers category.



Configuring an Item Extended Attributes Flexfield: Example

The Item Extended Attributes flexfield provides segments for extending the Item business object. In the Manage Extensible Flexfields task, you configure your product business object to include a Technical Specifications logical page in the user interface for the Electronics and Computers category of items.

In this example, your configuration of this flexfield groups custom attributes into the following contexts:

- Materials and Substances
- · Compliance and Certification
- Voltage

Scenario

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
 - o 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 table summarizes key decisions for this scenario.

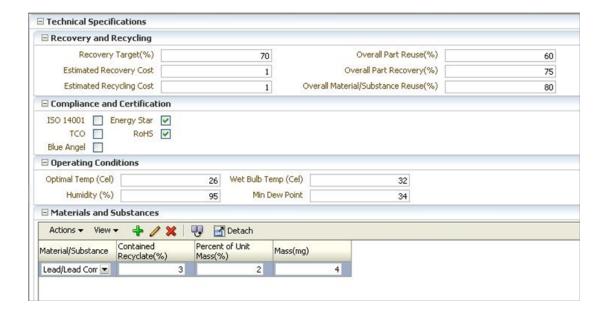




Collecting Technical Specifications

Your product inventory pages for electronics and computers require a technical specifications page. Your product inventory pages for furniture require a furniture specifications page and an assembly instructions page. Items in both the electronics and computer category, and in the furniture category, share attributes for specifying materials and substances.

The figure shows a Technical Specifications logical page in the user interface for the Electronics and Computers category, with attributes in the context of Recovery and Recycling, Compliance and Certification, Operating Conditions, and Materials and Substances. The Materials and Substances context is configured for multiple rows so your users can select all the materials and substances required to make a single product, displayed as attribute values in a table.



Analysis

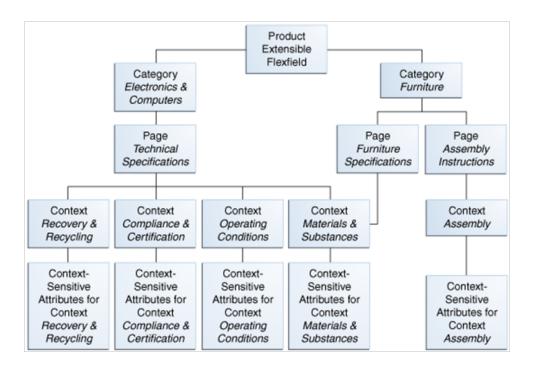
You use logical pages to arrange how the contexts appear in the user interface. Use a context to store all the materials and substances required to make a single product. You can configure a context to store multiple rows per entity. The multiple rows are displayed in a table, as for the Materials and Substances context.

The Technical Specifications logical page contains the attributes for the four contexts.

- Recovery and Recycling
- Compliance and Certification
- Operating Conditions
- Materials and Substances



In the figure, the Furniture category is configured to include a Furniture Specifications logical page and an Assembly Instructions logical page. The two categories (Electronics & Computers and Furniture) share the Materials & Substances context.



Configure Security for the Item Flexfield Configuration

The following table shows an example of data security policies for the Item flexfield.

Data Security Resource	Policy	Role	Actions	Condition
ITEM_EFF_B	А	VOLTAGE_SPEC	edit_ nontrans_ voltage_ctx	All values
ITEM_EFF_VL	В	COMPLIANCE_SPEC	edit_ trans_ compliance_ ctx	All values
ITEM_EFF_VL	С	COMPUTER_SPEC	edit_ trans_attrs	ComputerCategoryFilter
ITEM_EFF_VL	D	TELEVISION_ SPEC	edit_ trans_attrs	TVCategoryFilter

The following table shows the privileges for three of the flexfield's contexts.

Context	Edit Privilege	View Privilege
Voltage	edit_ nontrans_ voltage_ctx	NONE



Context	Edit Privilege	View Privilege
Compliance and Certification	edit_ trans_ compliance_ ctx	NONE
Materials and Substances	edit_ trans_attrs	NONE

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, the Materials and Substances context 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.

FAQs for Manage Extensible Flexfields

Why did the extensible flexfield context not appear at run time?

If a deployed extensible flexfield context doesn't appear in the user interface, verify that the context is associated with one of the category's pages defined for the extensible flexfield.

Manage Key Flexfields

Key Flexfields: Explained

Key flexfields provide a means to capture a key such as a part number, a job code, or an account code. A key flexfield consists of one or more segments, where each segment can have a meaning.

For example, a part number 10-PEN-BLA-450 might correspond to a black pen from supplier #450 sold by division #10 (office supplies). Behind the scenes, the application uses a unique number, 13452, for this part, but the user always sees the 10-PEN-BLA-450 part number.

The following aspects are important to understanding key flexfields:

- Architecture
- Segments and segment labels
- Structures
- Segment and structure instances
- Combinations
- Dynamic combination creation
- Security



Key flexfields aren't optional. You must configure key flexfields to ensure that your applications operate correctly. You configure and maintain key flexfield definitions with the Manage Key Flexfields task. To get a list of predefined key flexfields, open the Setup and Maintenance work area, and use the Manage Key Flexfields task. For information about specific key flexfields, see the help for the product where the associated business component is implemented.

Architecture

Flexfield metadata is stored in the flexfield metadata tables. When you configure a key flexfield, you define metadata about the key flexfield covering aspects such as:

- Segments are in a structure
- Structures in the flexfield
- Value sets in each segment

Based on the flexfield metadata, actual part numbers are captured at run time as a combination of segment values and stored in a combinations table. A combinations table contains all the segment columns for a flexfield, a unique ID column, and a structure instance number column. The structure instance number column differentiates multiple arrangements of the segment columns. For example, a part number containing multiple segments can be represented by a key flexfield. A part number key flexfield has a corresponding combinations table. In that table, the flexfield stores a list of the complete codes, with each segment of the code in a column, with the corresponding unique ID and structure instance number for the code. When users define a new part number or maintain existing part numbers in the parts catalog, they directly maintain rows in the combinations table.

The foreign key table contains a different business entity than the combinations table. For example, the business entity in the foreign key table is order lines or invoice lines that contain foreign key references to parts for ordering. Any number of foreign key tables can reference a particular entity represented by a key flexfield.

Segments and Segment Labels

A key flexfield contains segments and a segment label identifies a particular segment within a key flexfield. Segment labels are defined and made available by the product development. A segment contains the following details:

- A prompt
- A short prompt
- Display width
- The sequential position of the segment within the key flexfield structure
- The range type
- Column name of the attribute being stored by the segment
- A default value set
- A label for the segment

Applications identify a particular segment for some purpose such as security or computations. Segment name or segment order cannot reliably identify a segment because key flexfield segments can be configured to appear in any order with any prompts. A segment label functions as a tag for a segment.

For example, the requirement is to identify which segment in the accounting flexfield contains balancing information and which segment contains natural account information. A segment label determines which segment you are using for natural account information. When you define your accounting flexfield, you must specify which segment label apply to which segments. Some labels must be unique, and cannot be applied to more than one segment in each structure. Other labels are required, and must be applied to at least one segment in each structure.



A segment label helps a user searching for segments, such as the Cost Center label for all segments across key flexfields that store a value for the cost center.

Structures

A key flexfield structure definition includes the number of segments and their order.

In some applications, different users like to see different segment structures for the same flexfield. A key flexfield can have multiple structures if registered to support more than one structure.

The flexfield can display different fields for different users based on a data condition in your application data, such as the value of another field entered by the user or the user's role. For example, the correctly formatted local postal address for customer service inquiries differs based on locale. A postal address key flexfield could display different segments and prompts for different users based on a location condition in your application data, such as the user's role or a value entered by the user.

Each structure can have one or more segments. Thus a segment is a child of a structure. To store a particular segment, such as Cost Center, in two different structures, you must define the segment separately in each structure. Each structure may have one or more structure instances. Each instance of a structure shares the same number and order of segments, but differs in the values or value sets used in validating the segments.

Structure and Segment Instances

You can define multiple configurations of a key flexfield structure. These structure instances have the same segment structure, in the same sequence order. They differ primarily in how each segment is validated. You define a structure instance for each key flexfield and each key flexfield structure instance.

The segments in a key flexfield structure instance are segment instances. A segment instance is a segment with a specific value set assigned to it. If a key flexfield is registered with a tree structure, you can specify a tree code for a segment instance.

Combinations

A combination is a complete code, or combination of segment values that makes up the code, that uniquely identifies an object.

For example, each part number is a single combination, such as PAD-YEL-11x14 or 01-COM-876-7BG-LTN. In these combinations, the hyphen is the segment separator. If you have ten parts, define ten combinations. A valid combination is an existing or new combination that can be used because it's currently active and doesn't violate cross-validation or security rules. A combination has different segments depending on the flexfield structure being used for that combination. Any combination is associated with only one particular flexfield structure.

Many applications refer to a key flexfield combination by using the name of the entity or the key flexfield itself. For example, Assets uses the asset key flexfield and refers to one of its combinations as an asset key or asset key flexfield. In another example, Oracle Fusion General Ledger refers to combinations of the accounting flexfield as account or GL account.

Each key flexfield has one corresponding table, known as the combinations table, where the flexfield stores a list of the complete codes, with one column for each segment of the code, together with the corresponding unique ID number (an account combination ID) for that code. Then, other tables in the application have a column that stores just the unique ID for the code. For example, you may have a part number code, such as PAD-YEL-11x14. The Parts combinations table stores that code along with its ID, 57494. If your application lets you take orders for parts, you might then have an Orders table that stores orders for parts. That Orders table would contain a single column that contains the part ID, 57494, instead of several columns for the complete code PAD-YEL-11x14. Typically, one combinations page maintains the key flexfield, where the key flexfield is the representation of an entity in your application. Maintain individual combinations, such as part numbers in the combinations page.



Dynamic Combination Creation

Dynamic combination creation is the insertion of a new valid combination into a combinations table from a page other than the combinations page. Dynamic combination creation may be enabled at the following levels.

Level Of Dynamic Combination Creation	Controlled By:
Flexfield	Application development
Each usage or reference to the key flexfield	Application development
Structure instance	Administrators and implementation consultants
Other	Administrators and implementation consultants

If your key flexfield or certain usages or references of the key flexfield don't permit dynamic combination creation, you may control whether dynamic combination creation is enabled for each structure instance. If enabled, a user can enter a new combination of segment values using the flexfield window from a foreign key page. For example, when entering a transaction, a GL user can enter a new expense account combination for an account that doesn't yet exist. Your application creates the new account by inserting the new combination into the combinations table behind the scenes. Assuming that the new combination satisfies any existing cross-validation rules, the flexfield inserts the new combination into the combinations table, even though the combinations table isn't the underlying table for the foreign key page.

Planning Key Flexfields: Points to Consider

Your first step in planning your key flexfields is to determine which key flexfields your application requires.

Your plan should include:

- The purpose of the key flexfield
- The number and length of its available segment columns
- · Whether your key flexfield allows more than one structure
- Whether more than one structure must be defined
- The number, order and length of your segments for each structure

Consider the following aspects in planning flexfields:

- Before you begin
- Access to flexfield-related tasks
- Restrictions
- · Validation rules for flexfield segments

Before You Begin

Once you have identified a flexfield to configure, plan the configuration in advance. Compile a list of the UI pages and other artifacts in your deployment that are affected by the configuration. Verify that you are provisioned with the roles needed to



view and configure the flexfield. View the flexfield using the Highlight Flexfields command in the Administration menu while viewing the run time page where the flexfield appears. Plan how you will deploy the flexfield for test and production users.

Review the tools and tasks available for managing flexfields, such as the Define Flexfields task list and Manage Sandboxes.

If you plan to use value sets, create them before configuring the key flexfield. You cannot create value sets for key flexfields at the time that you add and configure key flexfield segments.

Access to Flexfield-Related Tasks

To access tasks for configuring flexfields and value sets, you must be provisioned with roles that entitle you to access the tasks in the Define Flexfields task list or tasks for managing product-specific flexfields. Contact your security administrator for details. For information about product-specific flexfield tasks, such as Manage Fixed Assets Key Flexfields, consult the product-specific documentation in Oracle Fusion Applications Help.

Restrictions

If you plan to use value sets, create them before configuring the flexfield.

Plan your key flexfield configuration to scale to your enterprise needs. For example, if you expect to disable old cost centers and enable new ones frequently, plan a larger maximum size for your cost center value set so that you can have more available values. One thousand available values for a 3-character value set provides more room for changes than 100 available values for a 2-character value set.

Note the code name of the flexfield you intend to configure so you can find it easily in the Define Flexfield task list or tasks for managing product-specific key flexfields.

In some cases you can customize how the flexfield appears on the page.

See Oracle Fusion Applications Help for specific products to determine any restrictions on using product-specific key flexfields.

Reporting

If you want to report on your data by certain criteria or sub-entities, such as account number or project or region, consider making that sub-entity a distinct segment, rather than combining it with another sub-entity, so that you can categorize and report on smaller discrete units of information.

Managing Key Flexfields: Points to Consider

Consider the plans for a key flexfield, security, and resulting run time pages when configuring key flexfields.

Planning

Plan structures carefully and allow for future needs. Don't change the number, order, and maximum length of segments once you have acquired flexfield data.

Structure Delimiters

A delimiter separates the segments when they appear to end users. The delimiter value of a structure specifies the character used to visually separate segment values when the key flexfield is displayed as a string of concatenated segments in the UI.

Choose the delimiter value of your key flexfield carefully so that it doesn't conflict with the flexfield data. For example, if your data frequently contains periods, such as in monetary or numeric values, don't use a period as your segment separator. Any character you expect to appear frequently in your segment values or descriptions isn't a good choice for the delimiter. If you



change the configuration of a key flexfield, such as the delimiter, the change affects the previously stored key flexfields with that structure.

Security

Oracle Fusion data security enforces value set security.

Within key flexfields, value set security applies to the selection of the individual segment values in the segment list of values. When selecting a key flexfield segment value from the combinations table, data security allows display of only the combinations whose segment values you have access to. Applications development controls whether or not value set security rules propagate to the foreign key table. By default they do.

Run Time Pages

Application development determines the user interface (UI) pages used to render flexfields. The types of key flexfield UI pages are as follows:

- Combinations pages where the underlying entity objects use the combinations table itself
- Foreign key pages where the underlying entity objects contain a foreign key reference to the combinations table
- Partial usage pages where some or all of the key flexfield's segment columns are in a product table

The same key flexfield can be used in different ways on different pages.

A page with a foreign key reference has a base table or view that contains a foreign key reference to a combinations table with the actual flexfield segment columns. This lets you manipulate rows containing code combination IDs (CCID).

A page with partial usage of a key flexfield presents segments that are defined on a product's transactional table in addition to being defined on a combinations table. In the case of a partial usage page, it is possible that only part of the configuration is visible. This enables the key flexfield to behave more like a descriptive flexfield.

A code combination maintenance page or combinations page presents the combinations table. This enables directly creating and maintaining code combinations. The combinations table contains all key flexfield segment columns and a unique ID column.

A typical application has only one combinations page. An application might not have a combinations page if it doesn't support maintenance by administrators.

A page containing a search region enables end users to select which attributes of the key flexfield view object to use as criteria to search for flexfield metadata.

For example, you can configure seven segments for the Account key flexfield. In a foreign key reference page, end users see the typical key flexfield picker with all seven segments where they can search for combinations. In a partial usage page using the same key flexfield, end users potentially could see only a single segment such as the Cost Center labeled segment, or they might see multiple segments but displayed as individual segments rather than as a picker for choosing combinations.

For more information on key flexfield pages, see the Oracle Fusion Applications Developer's Guide.



Key Flexfield Structures: Explained

A key flexfield structure arranges the segments of a key so that you can reuse a single key flexfield in multiple combinations of the same segments or a subset of those segments. Multiple instances of a single structure can accommodate differences in the value sets assigned to the structure's segments.

The structure determines the following aspects of a key flexfield:

- The segments to include
- The order of the segments
- Segment labels on the included segments
- Properties for each segment applied to the instances of the segments in an instance of the structure

Managing Key Flexfield Structures

All the segments defined for a key flexfield are available to be included in a key flexfield structure.

You can define as many segments as there are defined segment columns in your key flexfield combinations table. Ensure that you add segments in the order that your key requires. Once deployed, the order cannot be changed.

Enable segments to indicate that they are in use. A flexfield doesn't display disabled segments in run time. To protect the integrity of your data, disable a segment if you have already used it to enter data.

Key Flexfield Structure Instances and Segment Instances: Explained

A key flexfield structure can have one or more alternate structure instances.

The instances of a key flexfield structure share the following aspects of the structure:

- The same set of segments
- The same arrangement of segments
- The same properties at the segment and structure levels

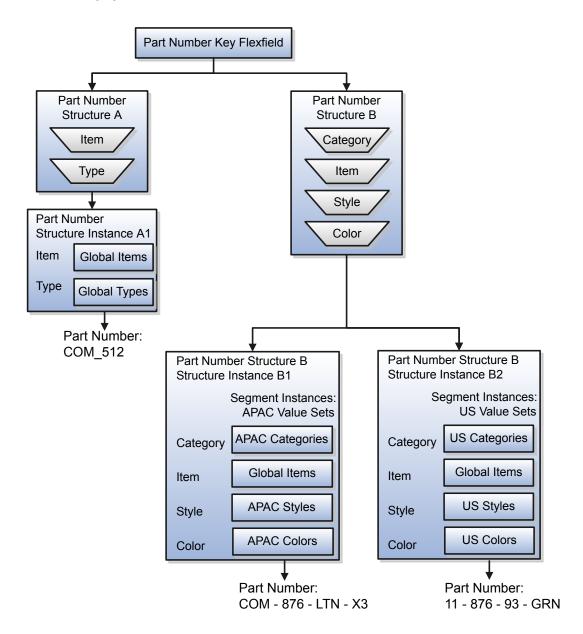
The differences among structure instances include whether dynamic combination creation is allowed. Likewise, at the structure instance level, differences among segment instances are based on the following:

- Value set
- Default type and default value
- Tree code
- Whether the segment is any of the following:
 - Required
 - Displayed
 - Enabled for business intelligence
 - Optional or required as a query criterion

For example, you can use one group of value sets for the US and another for France.



The following figure shows two structures instances for a part number structure.



The structures differ in the number of segments and the segment separators used. The structure instances of a structure share all properties defined for that structure. However, the structure instances may vary if the properties are defined at the structure instance or segment instance level. For example, the value set assigned to the segment instances.

Query Required Segment Instances

You can designate a key flexfield segment instance as query required to make it a selectively required attribute. A user can use it a key flexfield combination search. If you indicate on the Manage Key Flexfields UI page that a segment instance requires indexing, add the column representing the segment to the database index. Commonly, a database administrator (DBA) adds columns to the database index.



Following deployment, the combination picker of the key flexfield displays the query required attributes as selectively required. An user must specify at least one of the query required attributes in the search criteria. This prevents unnecessary searches that could cause performance issues.

For example, you mark the cost center and account attributes as query required and ensure that the corresponding columns in the database are indexed. A user can search for combinations by entering cost center or account or both as search criteria. No search is performed if a user doesn't enter at least one guery required attribute as search criteria.

Tip: Index the Structure Instance Number column on your combinations table to improve run time performance.

Dynamic Combinations

If a key flexfield supports dynamic combination creation, you can select to enable this feature by selecting **Dynamic Combination Creation Allowed**. As a result, users enter values at run time that produce new account combinations for the flexfield. If **Dynamic Combination Creation Allowed** isn't enabled, new valid combinations can only be entered using the combinations table for the flexfield.

Trees

You may define a tree code for the value set assigned to the segment instance. When you assign the tree code to the segment instance, tree hierarchy search operations are available on the segment values.

For a segment instance to be based on a tree, the following must be true.

- Application development registered the key flexfield with a tree structure. The tree structure may be fixed across all segments in the flexfield, or may vary across segments.
- A tree code for that tree structure exists.
- The tree code includes tree versions containing the values of the value set assigned to the segment instance.
- You assign the required tree code directly to the segment instance.

If these conditions are satisfied, different segment instances that use the same value set can be assigned the same or different tree codes. They use a different hierarchy definition over the same values.

Cross-Validation Rules: Explained

You can control the creation of new key flexfield code combinations by defining cross-validation rules. A cross-validation rule defines validation across segments and enforces whether a value of a particular segment can be combined with specific values of other segments to form a new combination.

The table compares segment validation to cross-segment validation:

Type of validation	Type of control
Segment validation	Controls the values you can enter for a particular segment
Cross-segment validation	Controls the combinations of values that administrators and end users can create for key flexfields



Note: You can use cross-validation rules for any key flexfield that has cross-validation enabled. See the documentation for your key flexfield to determine if it supports cross validation.

Cross-validation rules prevent the creation of combinations with values that shouldn't coexist in the same combination. For example, your company requires that all revenue accounts must have a specific department. Therefore, account combinations that have revenue account values, such as all values between 4000 and 5999, must have a corresponding department value other than 000, which indicates no department is specified. You can define cross-validation rules that disallow creation of combinations with incompatible segments, such as 4100-000 or 5000-000.

Alternatively, suppose your accounting key flexfield has an Organization segment with two possible values, 01 and 02. You also have a Natural Account segment with many possible values, but company policy requires that Organization 01 uses the natural account values 001 to 499 and Organization 02 uses the natural account values 500 to 999. You can create cross-validation rules to ensure that users cannot create a general ledger account with combinations of values such as 02-342 or 01-750.

The following aspects are important to understanding cross-validation rules:

- Rule Definitions
- Enforcement
- Timing

Rule Definitions

Cross-validation rules consist of the following information:

Rule Feature	Purpose
Name	Uniquely identifies cross-validation rules in a deployment.
Description	Helps administrators identify the purpose of the rule.
Error message	Explains why the attempted combination violates the rule.
Start Date, End Date	Indicates the period of time when the rule is in effect.
Enabled	Determines whether the rule is enforced.
Condition filter	Determines the conditions under which an enabled cross-validation rule should be evaluated.
Validation filter	Determines the validation that the rule enforces when that condition is met.

When the event specified in the condition filter is applicable, the validation filter condition must be satisfied before the combination can be created. If the event specified in the condition filter isn't applicable, then the combination is considered to pass the rule and the rule won't be evaluated even if it is enabled.

Note: If you don't specify any statement in the condition filter, then the condition is always true and the rule is always evaluated.



Enforcement

Cross-validation prevents creation of invalid combinations by administrators using maintenance pages and end users using dynamic insertion in foreign key pages.

Enabled rules are enforced when there is an attempt to create a new combination of segment values. Disabled rules are ignored. Deleting the rule has the same effect, but you can re-enable a disabled rule.

Timing

When users attempt to create a new combination, the key flexfield evaluates any cross-validation rules that are enabled and in effect.

Note: Cross-validation rules have no effect on combinations that already exist. The flexfield treats any existing invalid combinations that pre-date the rule as valid.

If you want to prevent users from using previously existing combinations that are no longer valid according to your cross-validation rules, manually disable those combinations using the combinations page for that key flexfield.

When defining a cross-validation rule, specify a start and end date to limit the time when the rule is in effect. The rule is valid for the time including the From and To dates.

Cross-Validation Rules: Points to Consider

When you need key flexfield combinations of segment values validated across segments, you can optimize your cross-validation rules to improve the experience of administrators and end users.

Consider the following when defining cross-validation rules:

- Filters
- Rule Complexity
- Maintenance

Filters

A cross-validation rule includes a condition filter and a validation filter.

The rule is evaluated using the following logic: If the condition filter is satisfied, then validate that the validation filter is satisfied.

- 1. The condition filter describes the event under which the rule will be evaluated. If the event specified in the condition filter isn't applicable, then the rule won't be evaluated even if it is enabled.
- 2. When the event specified in the condition filter is applicable, the validation filter condition must be satisfied before the combination can be created.

For example, if your organization has determined that a certain company value, Operations, cannot use a specific cost center, Marketing, you can define a cross-validation rule to validate your combinations.

- 1. The rule evaluates the company condition filter.
- 2. When company is equal to Operations, the rule evaluates the cost center validation filter.
- 3. When cost center is equal to Marketing, the rule prevents a combination from being created.
- **4.** The error message you defined for the rule displays to inform the user that the attempted combination violates the rule.



Note: This rule doesn't affect the creation of combinations with Marketing cost center and company values other than Operations.

Rule Complexity

For optimal performance and ease of understanding, define several simple validation rules instead of using one complex rule. Simple validation rules let you provide a more specific error message and are easier to maintain over time.

Avoid rules that control validation across more than two segments, where possible. While you can define cross-validation rules that span two or more segments, keep in mind that it becomes more difficult to interpret cross-validation error messages and correct invalid key flexfield combinations as your rules encompass more segments.

Maintenance

To maintain consistent validation, review existing key flexfields when you update your cross-validation rules. Regardless of your current validation rules, Oracle Fusion Applications accept a key flexfield combination if the combination already exists and is enabled. Therefore, to ensure accurate validation, you must review your existing combinations and disable any combinations that don't match the criteria of your new rules.

💡 Tip: To keep this type of key flexfield maintenance to a minimum, decide upon your cross-validation rules when you first set up your key flexfield structure. Define cross-validation rules before creating combinations and before combinations are used in transactions.

If you want to prevent users from using previously existing combinations that are no longer valid according to your crossvalidation rules, disable those combinations using the combinations page.

Creating a Cross-Validation Rule: Example

Create cross-validation rules to prevent specific combinations of segment values in account combinations. For example, you can prevent a particular cost center from being combined with a specific company value. Cross-validation rules only affect the creation of new account combinations.

Scenario

Your company, InFusion America Inc. doesn't have a marketing department. Create a cross-validation rule to prevent InFusion America Inc. company value 01 from being combined with marketing department value 300 in an account combination.

- 1. Navigate to the Setup and Maintenance work area. Search for and select the Manage Cross-Validation Rules task.
- 2. Select the chart of accounts for InFusion America and click Add Row.
- 3. Enter a unique rule name, such as IFAM01 and a description, such as Do not combine Marketing department, 300 with InFusion America company 01.
- 4. Specify an effective starting date of today and enable the rule.
- 5. Click the Condition Filter icon and enter Company equal to 01. The cross-validation rule evaluates if company 01 is entered, and if so, then the validation process continues to evaluate the rule.
 - Note: If you don't specify a statement in the condition filter, then the rule is always evaluated.
- 6. Click the Validation Filter icon and enter Cost Center is not equal to 300. When the rule is evaluated, an account combination must contain a cost center other than 300 before it can be created.



- 7. Enter the error message: Cost Center 300 is not allowed with Company 01. The message displays in the relevant user interfaces and processes when an account combination can't be created because it violates the rule.
- 8. Click Save and Close.

Enabling Key Flexfield Segments for Business Intelligence: Points to Consider

A key flexfield registered in the database as enabled for Oracle Business Intelligence (BI) includes a BI Enabled setting for each of its segment instances. When a segment instance is BI-enabled, it's available for use in Oracle Business Intelligence.

The following aspects are important in understanding BI-enabled key flexfield segments.

- Flattening business components to use BI-enabled segments in Oracle BI
- Equalizing segments to prevent duplication and complexity in the flattened component
- Mapping attributes of flattened business components to logical objects in Oracle BI
- Managing the labels that map segments to logical objects in Oracle BI

After you deploy a business intelligence-enabled flexfield, use the Import Oracle Fusion Data Extensions for Transactional Business Intelligence process to import the flexfield changes into the Oracle Business Intelligence repository. Users can make use of the newly-generated attributes in business intelligence applications. For additional information about logical objects and import, refer to the Oracle Transactional Business Intelligence Administrator's Guide.

Flattening

When you deploy a business intelligence-enabled key flexfield, the deployment process generates an additional set of flattened business components for use in business intelligence. The flattened business components include attributes for business intelligence-enabled segment instances only.

If you assigned a label to a segment, the flattened components include a single attribute representing all segment instances with that label. If you didn't assign a label, the flattened components include a discrete attribute for each BI-enabled segment instance in each structure.

Mapping to Logical Objects in Business Intelligence

You can simplify reporting by representing similar segments as a single logical object in Business Intelligence. If you assign a label to segments that serve the same purpose in different structures, you can consolidate the segments into a single attribute. This prevents duplication and the extra workload and complexity that result from the flattening process. For example, an organization may have more than one definition of its key accounting flexfield to support different requirements for accounting reporting. A US Accounting Flexfield structure may have a segment called Subaccount to track project expenditures. The same type of information may be tracked in a UK accounting flexfield structure with a segment called Project. Equalize these two segments to create a single list of values for reporting.

Non-labeled segments aren't equalized across context values, so the flattened components include a separate attribute for each segment for each structure. It may not be possible to equalize similarly labeled segments if they have incompatible data types or value set types.

Assign a label to a segment to map the corresponding attribute in the flattened components to a logical object in Oracle Business Intelligence. Using labels to map segments to BI logical objects minimizes the steps for importing the flexfield into Oracle Business Intelligence. Assigning a label to a segment serves to equalize the attribute across structures, as well as map the equalized attribute to business intelligence.



Managing Labels

You may assign a predefined label (if available) to segments or create labels for assignment, as needed. Specify a code, name, and description to identify each label. In the BI Object Name field, enter the name of the logical object in Oracle Business Intelligence to which the segment label should map during import. Specifying the BI logical object minimizes the steps for importing the flexfield into Oracle Business Intelligence and helps to equalize context-sensitive segments across structures.

If no labels are assigned to a BI-enabled segment, or the BI Object Name on the assigned label doesn't exist in business intelligence, you must manually map the segment to the required logical object when importing into Oracle Business Intelligence. In addition, segments without labels cannot be equalized across structures. The flattened components include a separate attribute for each non-labeled segment in each structure.

Importing to Oracle Business Intelligence Repository

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. Then propagate the flexfield object changes. When you import the metadata into the Oracle Business Intelligence repository, you must do so as the FUSION_APPS_BI_APPID user.

To import flexfield changes into the Oracle Business Intelligence repository in Oracle Cloud implementations, run the Import Oracle Fusion Data Extensions for Transactional Business Intelligence process. For additional information about import, refer to the Oracle Transactional Business Intelligence Administrator's Guide.

Note: When you import a flexfield into the Oracle Business Intelligence repository, you see both <name>_ and <name>_ attributes for each segment, along with some other optional attributes. The <name>_ attribute contains the value. The <name>_ attribute contains the value set that the value comes from, and is used for linking to the value dimension. You must import both attributes.

Key Flexfields: Example

A key flexfield can capture expense account information.

Scenario

When entering details for each expense, the user specifies an account to which the expense is charged.

Entering Expense Accounts

A user interface for entering expenses gives the user the option of selecting an expense account that identifies the cost center and other details needed for processing the expense.

Analysis

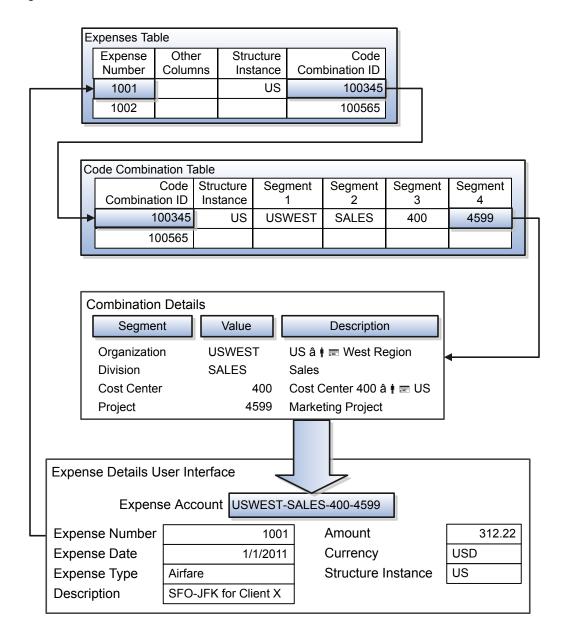
The expense account field is a foreign key reference to a code combination (EXPENSE_LINES.EXPENSE_ACCOUNT = ACCOUNTS.CCID).

Code Combinations Table for Entering Accounts and Employees

The code combinations table supports entering account information, such as for expense accounts.



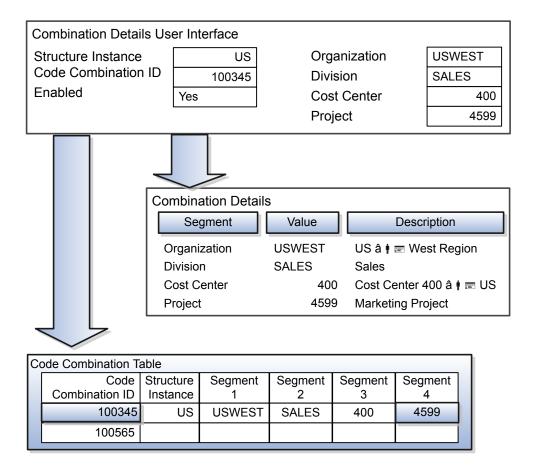
The figure shows the origin in the code combinations table of the account specified by the user. The code combination ID record stores the information of the key flexfield segments used to assemble the expense account based on the key flexfield configuration.



The combinations page, which is the maintenance page for the key flexfield, is for managing rows in the combinations table. In this example, managing the combinations means adding or editing account numbers that adhere to the key flexfield metadata rules.

The figure shows the code combination details for the example expense account reflected in the flexfield configuration and the code combinations table.





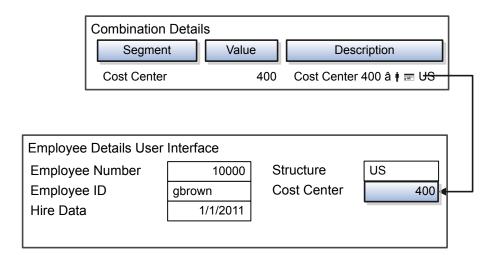
If dynamic combination creation isn't enabled, then when entering an expense line, the user can only select an account that already exists in the ACCOUNTS (combinations) table. If they require an account that doesn't exist, they must consult with the appropriate application administrator who can add the account to the combinations table.

If dynamic combination creation is enabled, then when entering an expense line, the user can either select a pre-existing account, or type in a new account that is created dynamically on the fly in the ACCOUNTS (combinations) table. Once the new combination is created, the same user can refer to it on the expense line.

When managing employee information, the user specifies the cost center that the employee belongs to. The cost center field corresponds to a single, labeled segment of the Account Key Flexfield and has metadata defined such as the allowable value set for that segment.

In this figure, instead of specifying a cost center ID reference to an account, only the Cost Center segment is used and the value is stored directly on the employee table.





Related Topics

• Creating One Chart of Accounts Structure with Many Instances: Example





5 Navigator and Springboard Customization

Customizing the Navigator and Springboard: Overview

You can customize the Navigator and springboard, as well as define settings for the home page and springboard using the Structure page. To address needs specific to your organization, you can create or edit categories and page entries for the Navigator and springboard. For example, you may want to link page entries to web pages or external applications.

Categories and Page Entries

Categories and page entries are the navigator menu items. A page entry is the navigator link or springboard icon that opens a page. A page is a single screen to perform related tasks. A few page entries may be grouped under a category. Depending on the number of page entries that you have access to, the page entries can appear at the top level (not in any category folder) on the springboard. If you have only one page entry under a category, then that page entry icon appears at the top level (not under any category) on the springboard. However, such page entry icons appear under their respective categories on the Navigator menu.

While creating or editing a page entry or category, you can use the **Show on Welcome Springboard** field to specify whether to display them on the springboard. So, not all page entries and categories may appear on the springboard. The Navigator menu may have more page entries and categories than the springboard. If a page entry appears on both the Navigator menu and springboard, then you can use either of them to open the page. For page entries that don't appear on the springboard, use the Navigator menu to open those pages.

Customizing the Navigator and Springboard Using the Structure Page

For customizing the Navigator and springboard, use the Structure page to do the following tasks on categories and page entries:

- Create
- Edit
- Show or hide
- Reorder
- Note: If a page has both desktop and simplified versions, by default, users open the simplified version of the page from the Navigator menu or springboard. However, to open the desktop page instead of the simplified page, set the Desktop Pages Version Enabled profile option (FND_CLASSIC_INTERFACE) to Yes.

Prerequisites for Customizing the Navigator and Springboard

Before customizing the Navigator and springboard using the Structure page:

1. From the Navigator menu, select **Tools - Structure**.



- Activate a sandbox. If you're not in an active sandbox, click Edit on the Structure page. You're prompted to activate a sandbox.
 - \bigcirc **Tip:** If you're already in an active sandbox, then the Edit button doesn't appear on the Structure page.

If prompted, select a customization layer to determine the scope of users that your changes affect.

Related Topics

- Setting Up Sandboxes: Procedure
- · Navigating in the Application: Explained
- Setting Profile Option Values: Procedure

Customizing the Navigator and Springboard

Watch: This tutorial gives you an overview of the Navigator and springboard and shows you how to customize their categories and page entries.

Creating Categories and Page Entries for Navigation: Procedure

Use the Structure page to create categories and page entries for customizing the Navigator and springboard. From the Navigator menu, select **Tools - Structure**.

You can do either of the following:

- Create a category and then create a page entry in that category.
- Create a page entry in an existing category or at the top level (not in any category).

Creating Categories and Page Entries

To create a category or a page entry:

- 1. Click Create, and select Create Category or Create Page Entry.
- 2. Enter a name for the category or page entry. For a category, the one available icon is already selected for you.
- **3.** If you're creating a page entry:
 - Search and select an icon for the page entry.
 - Select the category in which you want to place the new page entry.
- **4.** Select **Yes**, **No**, or **EL Expression** in the Visible field:
 - Yes: The category or page entry appears on the Navigator. It can also appear on the springboard, depending on what you select in step 6.
 - No: The category or page entry doesn't appear on the Navigator and springboard.



- EL expression: The evaluation of the EL expression decides whether the category or page entry will appear on the Navigator and springboard.
- 5. If you have selected EL Expression for the Visible field, click the **Edit** icon next to the Visible drop-down list, and enter a value or expression. Don't include spaces or double quotes in the EL expression.

For example, depending on the user role that you want to display the categories or page entries for, enter the EL expression as described in the following table:

Who can see the category or page entry	EL Expression and Example
Only users having any of the specific roles	#{securityContext. userInRole[' <rolename>']} #{securityContext. userInRole['ORAFND APPLICATION_ ADMINISTRATOR_ JOBORA_ PER_ EMPLOYEE_ ABSTRACT']}</rolename>
Only users not having any of the specific roles	#{!(securityContext. userInRole[' <rolename>']} #{!(securityContext. userInRole['ORAFND APPLICATION_ ADMINISTRATOR_ JOBORA_ PER_ EMPLOYEE_ ABSTRACT'])}</rolename>
Only users having all of the specific roles	#{securityContext. userInAllRoles[' <rolename>']} #{securityContext. userInAllRoles['ORAFND APPLICATION_ ADMINISTRATOR_ JOBORA_ PER_ EMPLOYEE_ ABSTRACT']}</rolename>
Only users not having all of the specific roles	#{!(securityContext. userInAllRoles[' <rolename>']} #{!(securityContext. userInAllRoles['ORAFND APPLICATION_ ADMINISTRATOR_ JOBORA_ PER_ EMPLOYEE_ ABSTRACT'])}</rolename>

- Note: Categories and page entries are evaluated on all pages. So roles used in the EL expression for the Visible field must be available in all application stripes, that is, in CRM, FSCM, and HCM.
- **6.** For a page entry, if you have set the Visible field to Yes or EL Expression, then set the Show on Welcome Springboard field. The value of this field is evaluated to determine if the page entry will actually appear on the springboard.
 - Yes: The page entry appears on the springboard.
 - Suppose you have specified an EL expression in the Visible field. Then, even on setting the Show on Welcome Springboard field to Yes, the page entry may not appear on the springboard. The evaluation of the EL expression decides whether these items will actually appear on the springboard.
 - No: The page entry doesn't appear on the springboard.
 - EL expression: The evaluation of the EL expression decides whether the page entry will appear on the springboard.
- 7. If you're creating a category, then click **Save and Close**. If you're creating a page entry, then perform steps 8 to 10 instead of this step.
- 8. Select any of the following link types for the page entry:
 - An application page.



- A dynamic URL of an external page (outside your application) where the host, port, or context root might change.
- A Static URL of an external page (outside your application) where the host, port, or context root doesn't change.
- 9. Based on the link type, specify the required details to configure the link.
- 10. Click Save and Close.

Configuring Links for Page Entries: Procedure

While creating a page entry or editing a custom page entry for the Navigator and springboard using the Structure page, you can determine what the page entry links to.

Use the Create Page Entry or Edit Page Entry page to link a page entry to any of the following link types:

- Your application page.
- A dynamic URL of an external page (outside your application) where the host, port, or context root might change.
 - You can determine the host and port details, which a dynamic URL starts with, from a lookup based on the application name.
- A static URL of an external page (outside your application) where the host, port, or context root doesn't change.
 Static URLs don't require lookups.
- A secure token URL of a partner application, to which secure tokens are added for enhanced security.

Linking to Application Pages

This option is applicable for non-Cloud implementations only. For Oracle Cloud implementations, you can configure the links to application pages while creating your pages using the Page Integration Wizard page.

To link a page entry to one of your application pages:

- 1. Select the Application Page link type.
 - If there is another page entry that links to the same application page, then you can enter the same details for all fields, except parameters.
- 2. Enter the focus view ID of the target page.
 - You can get the Focus View ID from the value of the page's view id attribute in the product's public_html/WEB-INF/adfc-config.xml file.
- 3. Select the name of the web application.
 - This is the application name that you had entered while creating this third party application using the Setup and Maintenance work area.
- 4. For a secure access to the target application page from the page entry, provide the secured resource name and the name of the policy store's application stripe. An example of a secured resource name is oracle.apps.view.pageDefs.CaseList_Form_Attach_UIShellPagePageDef. When a user clicks the link, the application checks the secured resource and the Lightweight Directory Access Protocol (LDAP) policy store. Then, the application determines whether the user has the privilege to view the page.

You can get 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.



5. If the page takes parameters, then you can enter a semicolon-delimited string of name=value pairs (for example, org=m1; context=s1) in the Page Parameters List field.

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. An application page may display or act differently based on the parameters that are passed in. For example, if you're opening a page from one category on the springboard or Navigator, the parameter might be set to status=Open. Whereas, if you're opening the page from another category, the parameter might be set to status=Closed.

Linking to Dynamic URLs

You can link a page entry to an external web site or application that has a frequently changing host, port, or context root. Instead of updating the link to each application, you can update the details of the web application in the topology registration. This change affects all page entries that contain dynamic links pointing to that web application.

For example, say you need to link to a test version of an application. So, you use the dynamic URL link type. When you move the application from test to production environment, just change the host and port details of the web application in the topology registration. This change will affect all page entries that contain dynamic links pointing to the web application.

As a prerequisite, use the Register Enterprise Applications task to register the web application in the topology.

To link a page entry to a dynamic URL:

- 1. Select the Dynamic URL link type.
- 2. Specify the name of the web application and the destination for web application.

For example, say, you need to link to a complete URL: http://example:9011/myApp/faces/Page1.

Then:

- The name of the web application added to topology will be: myApp (the value that would eventually appear in the web application list). The protocol host, port, and context root values of the URL will be: http://example:9011/myApp.
- The destination for the web application will be: /faces/Page1.

After linking a page entry to a dynamic URL, when you click the page entry, the target page opens in a new browser window or tab.

Linking to Static URLs

You can link a page entry to an external web site or application that has a constant host, port, or context root.

To link a page entry to a static URL:

- 1. Select the Static URL link type.
- 2. Specify the URL destination.

For example, you can use a static URL to link to http://www.oracle.com.

Linking to Static URLs with Secure Destinations

To link a page entry to a secure token URL of a partner application (that is, outside your application):

- 1. Select the Static URL link type.
- 2. Select Secure Destination.



- 3. Select the name of the web application.
- 4. Enter the destination for the web application. A HTTPS protocol is required to access the application.
- **5.** Enter the name of the secure token. The secure token expires if the user session is inactive. So, users need to refresh the page to regenerate the tokens.

The application validates the secure token and uses it to authenticate web services within the end user context. Using this mode of customized access, a partner can directly perform an action or display information to the specific user without any additional authentication.

Related Topics

• Creating Pages for Hosting Third Party Applications: Procedure

Managing Categories and Page Entries for the Navigator and Springboard: Procedure

Use the Structure page to edit and reorder the existing categories and page entries for customizing the Navigator and springboard.

Editing Categories and Page Entries

Follow these steps:

- 1. From the Navigator menu, select **Tools Structure**
- 2. On the Structure page, click the name link for the category or page entry.
 - Tip: You can use the search panel on the Structure page to find the category or page entry you want to edit.

If you get a "webApp value not defined" error message on clicking a category or page entry, verify whether the application is in the topology tables. For more information, refer to the Oracle Fusion Applications Administrator's Guide

- 3. On the Edit Page Entry page or the Edit Category page, make the required changes.
- 4. Click Save and Close.

You can make the following changes to a category or page entry:

- Rename a category or page entry.
 - Note: If a category or page entry was created using a different tool, then you can't change its name using the Structure page.
- Change the icon for a page entry.
 - You can't change the icon for a category.
 - olf a page entry was created using a different tool, then you can't change its icon using the Structure page.
- For a page entry, change the category under which the page entry is grouped.



- Change the Visible property for the category or page entry.
- · For a custom page entry, change the settings for link configuration.
- Delete page entries that are created using the Structure page.

Editing Page Entries with Tabs

Some pages (for example, Security Console) have tabs. Each tab is a task flow. The tabs are displayed at the bottom of the Edit Page Entry page. On the Edit Page Entry page, you can:

- Click the tab name to rename it
- · Click the tab icon to search and select another icon for the tab
- Click the Visible field for the tab, and change the option to show or hide the tab
- Use the Move Up and Move Down icons to adjust the relative position of the tabs within the page

Reordering Categories and Page Entries

Use the **Move Up** and **Move Down** icons on the Structure page to reorder categories and page entries. For page entries, you can use the **Move To** icon to move page entries to different categories or to the top level.

Related Topics

· Customizing Themes: Overview

Defining Settings for Home and Navigation: Explained

Click **Set System Options** on the Structure page to define settings for the home page and springboard. To open the Structure page, from the Navigator menu, select **Tools - Structure**.

Note: You must be in an active sandbox to define these settings.

Defining Settings

You can define the following settings:

- Home Panel: Specify the content of the side panel that appears to the left of the springboard on the home page.
 Select to display either social networking content or announcements in the panel. By default, social networking content is displayed in the home panel. Also, this panel isn't visible to partners or other external users who sign in to your application, by default. To provide access to such users, contact your security administrator.
- **Enable Infolets**: Select the infolet pages that you want to include on the home page. Based on your selection, these infolet pages will appear as a row of dots on the home page. Users can click a dot to navigate to each included infolet page. If you don't select any infolet page in the settings, no such dots will appear on the home page.

You can use profile options to define settings for the springboard strip that you can find above all simplified pages:

 To enable users to use the springboard strip, set the Springboard Strip Enabled profile option (FND_USE_FILMSTRIP) to Yes.



• If the FND_USE_FILMSTRIP profile option is set to Yes, then you can display the springboard strip as expanded by default. To do so, set the Springboard Strip Expanded profile option (FND_EXPAND_FILMSTRIP) to Yes. A user can still collapse or expand the strip on any page, and once done, this profile option is set by default for subsequent sessions of that user.

Related Topics

Setting Profile Option Values: Procedure

FAQs for Navigator and Springboard Customization

Why can't I edit the Structure page entry or the Tools category?

While customizing the Navigator and springboard, you can't:

- Override the Visible setting for the Structure page entry and Tools category. The default setting is Yes, so the Structure page entry and Tools category always appear on the Navigator and springboard.
- Move the Structure page entry to a different category or to the top level. The default category is Tools, so the Structure page entry always appear under the Tools category.

Why are some springboard icons not displayed on my springboard?

A springboard icon, that is, a group or a page entry may be hidden due to any of the following reasons:

- You may not have security privileges to access or view the group or page entry. To review, check if the group or page entry is available on the Navigator menu. If no, then that means you don't have the appropriate privileges.
- The administrator has hidden the group or page entry from the springboard using the Structure page. To review the visibility settings, from the Navigator menu, select **Tools Structure**.
- The administrator hasn't enabled the offering associated with the group or page entry.
- The accessibility preference for the application is set to the screen reader mode. To review accessibility preferences, in the global area, either click the Accessibility icon or select Settings and Actions - Personalization - Set Preferences.

Related Topics

Enabling Offerings: Explained

Enabling Offerings: Procedure

Accessibility Preferences: Explained

Why are springboard icons appearing at the top level instead of under a folder?



6 Help Customization

Customizing Help: Overview

You can customize help files in Applications Help and help windows, as well as help text that appears on UI elements. The Getting Started work area can also be customized.

Note: To enable help file and Getting Started customization, you or your implementor must select the Help Customization feature choice in your offering configurations.

Help Files

A help file, such as an FAQ or example, provides information about a specific topic. You can find help files in:

- **Help windows:** Which can contain informational text as well as links to help files. Click a help icon next to a page or section title to open the help window and find information about that page or section.
- **Applications Help:** The help site which contains all help files, including help files found in help windows. To open Applications Help at any time, click your user name in the global area and select **Applications Help**.

You have many options to customize help files, including:

- Creating help files with your own content and adding them to help windows
- Editing the content in predefined help files

Getting Started Pages

The Getting Started work area contains a set of pages that provides information for new users. To open this work area, select **Getting Started** from the Navigator. You can customize predefined pages and create pages with your own custom content.

Embedded Help

Help text might appear when you hover over or click certain UI elements on the page. For example, hint text can appear when you put your cursor in a specific field. The help text you see on the page is called embedded help. To customize embedded help, you can, for example:

- Edit the help text that appears for a specific check box.
- Add help text that appears when users hover over a specific tab.

Related Topics

Features: Explained



Who can add and manage custom help?

Users with the Customize Help Topics (ATK_CUSTOMIZE_HELP_TOPICS_PRIV) privilege can customize:

- Help in Applications Help and help windows
- · Pages in the Getting Started work area

This privilege is assigned by default to the administrators for product families. Your security administrator can define which users have job roles with this privilege.

Help File Customization

Overview

If you have the appropriate roles, then you can customize the help files in Applications Help and help windows (which users open using help icons on the pages they work in). You can also determine which help files appear in which help windows, and which product family tabs a file belongs to in Applications Help.

What you can do to a help file depends on whether it's custom or predefined.

Custom:

- o Create, duplicate, edit, and delete
- Set status (Active or Inactive)

Predefined:

- Duplicate
- Edit (which is really creating a custom version of the predefined file)
- Set status

Navigation

This table describes where to go to customize help.

Help Customization Task	Navigation
Create or edit help for a specific help window.	Click the Manage Custom Help link in the help window.
Edit any help file, including glossary terms.	Open the file in Applications Help and click the Edit link.



Help Customization Task	Navigation
Perform any help file customization task, including managing a set of help (such as all help for a product)	Go to Applications Help and click your user name in the global area to select Manage Custom Help.
Make a copy of all custom help for testing, migration, or other purposes.	Create a configuration package then use the export and import feature in the Setup and Maintenance work area. • The configuration package must use a source implementation project that contains the Define Help Configuration task list. • Select the following objects to export: • Help Configuration • Help Topic

Related Topics

• Implementation Project Based Export and Import: Explained

Source Types for Custom Help: Explained

You have many options to determine the content of custom help, for example by uploading a file, specifying a URL, or typing in the text. When you create help files, you first select a value for the **Source** list, to indicate how to provide your content.

Desktop File

Upload a file of any type from your computer.

Oracle User Productivity Kit

Identify the custom Oracle User Productivity topic to use as help.

Text

Use a rich text editor to enter the content of the help file.

URL

Enter the full URL to a Web site or a file of any type.

Video URL

For a video to play in the help windows assigned to the help file, enter the full **URL** to the video on YouTube or elsewhere.

Help Types: Explained

Applications Help has many types of help content: examples, FAQs, glossary terms, help topics, PDF guides, and videos.



Example

Examples can provide:

- Real use cases to illustrate how and when to do something
- Scenarios to explain abstract concepts

Worked examples show exactly what you do to achieve a specific result. They emphasize decisions that you make and values that you enter.

FAO

FAQs, or frequently asked questions, provide brief answers to questions that you might have about a task or page. For example, they can explain:

- · What a term means
- Why something happened
- How you can perform an action
- What happens if you perform the action

Glossary Term

Glossary terms provide definitions for words or phrases used in help content. When you read help and see terms underlined with dots, you can hover over the term to see its definition. To see the whole glossary, select **Glossary** from the Navigator menu in Applications Help.

Help Topic

Help topics can:

- Explain key concepts
- Tell you the steps to follow to perform tasks
- Help you make decisions by explaining points to consider or describing the options you have
- Show you how application components work together
- Provide reference, overview, and other information

PDF Guide

PDF guides provide information in a book format. The guides in Applications Help contain content that you usually can't find in other help types.

Note: Most of the examples, FAQs, and help topics in Applications Help are also in guides. To see these guides, select **Documentation Library** from the Navigator menu in Applications Help.

Video

Videos, or tutorials, show you how to complete a short task or part of a task. Videos can also give you an overview of complex dashboards and work areas.



Page or Section Values: Explained

The Page or Section value represents where users can click a help icon to open a help window that can contain links to help files. Use this value on the Manage Custom Help page in Applications Help to:

- Search for help that appears in a specific help window.
- Add help files to the help window.

Where Help Windows Are Found

In most cases, the page or section value corresponds to a page or section header that has a help window. Help windows are also available:

- On specific tabs, dialog boxes (windows), or dashboard regions
- In the Setup and Maintenance work area, next to names of task lists or tasks in tables

How to Read the Values

The page or section value reflects the logical navigation to the help window. For example, **Process Details window, Output tab** doesn't mean that the help window is in two different places. It's in the Output tab within the Process Details dialog box.

When the Same Help Window Is On Multiple Pages

In some cases, a page or section value does represent a single help window that appears on multiple pages. If the value is:

- For example Create and Edit Message pages, then the same help window is on both the Create Message and Edit Message pages.
- Just a region name without a page or window name, then the same help window is in a section or dashboard region that appears on multiple pages.

Customizing Help

Watch: This video tutorial shows you how to add and edit help in help windows.

Customizing Help in Help Windows: Procedure

When users click help icons on a page, they open help windows that have informational text, links to help files, or both. Use the Manage Custom Help dialog box to edit those help files, create new files to appear in the help window, change the order of links in the window, or add links to existing custom help. To restrict access to any of those help files, use the Manage Custom Help page in Applications Help to assign a help security group to the help file.

- Tip: You can also use:
 - The Customer User Interface Text tool to edit the informational text that appears in a help window
 - Page Composer to hide the More Help link at the bottom of a help window



Opening the Manage Custom Help Dialog Box

Use the Manage Custom Help dialog box to create and edit help files, or to reorder links:

- Click the help icon to open the help window you want to customize. If you don't see any help icons, click **Show** Help in the global area.
- 2. Click Manage Custom Help.

Editing Help Files

To edit help files in the help window:

- 1. Open the Manage Custom Help dialog box for the help window.
- 2. Click the title of the help file.
 - Note: If you're updating a predefined file, then you see the Create Help dialog box because you're really creating a custom version of the predefined file.
- 3. Change the title or content of the help file as needed, as well as the status.
 - Setting the status to Inactive hides the file in the help window and in Applications Help.
- **4.** Open the More Details section if you need to update more aspects of the help file, including fields that correspond to the following filters in Applications Help:
 - Help Type
 - Role
 - Language
 - Country
- 5. Save your work.

Creating Help Files

To add a custom help file that appears in the help window (and also Applications Help):

- 1. Open the Manage Custom Help dialog box for the help window.
- 2. Click Create.
- 3. Select a source and enter the content of the help file.
- 4. Select the help type, which corresponds to the Help Type filter in Applications Help.
- 5. Enter the title, keywords, and description for your file.
- 6. Optionally select a language, country, or role, which correspond to filters in Applications Help.
- 7. Save your work.

Adding Links to Help Windows

To add existing custom help files as links in any help window:

- 1. In Applications Help, click your user name in the global area and select Manage Custom Help.
- 2. In the search, select the page or section that corresponds to the help window.
- 3. Click Search.
- 4. In the search results toolbar, click Select and Add to Help Location.
- 5. Search for the existing custom help file, select it, and click **Apply**.
- 6. Repeat the previous step for all the help files you want to add.
- 7. Click Done.



Assigning Help Locations: Procedure

To determine which help windows your help file appears in, assign the appropriate help locations to the file. You also use help locations to define where help files appear in the Task and Product filters in Applications Help, as well as which product family tabs the files belong to. Help locations include:

- Task hierarchy for the Task filter and product family tabs
- Product hierarchy for the Product filter and product family tabs
- Page or section values for help windows

Help locations are available only on the Manage Custom Help page in Applications Help, not the Manage Custom Help dialog box from help windows. When you create help from the help window, the help files are automatically assigned to the help window and task hierarchies associated with the window.

Determining Where Help Appears in the Task Filter, Product Filter, and Product Family Tabs

Enter at least one hierarchy, as many as you need:

- 1. In Applications Help, click your user name in the global area and select **Manage Custom Help**.
- 2. Find the help file to edit or duplicate, or click **Create**.
- 3. Enter or update the general information for the help file.
- 4. In the Help Location section, add a row if there isn't already a blank one, or edit an existing row.
- 5. Select **Task** or **Product** for the hierarchy type.
- **6.** Select nodes for as many levels of the hierarchy as you need, starting with level 1. If the Task hierarchy nodes you're assigning are at level 4 or lower, then click the **Details** icon to assign nodes.
- 7. Add more rows as needed.
- 8. Save your work.

Selecting Help Windows

Every page or section value is associated with a specific node in the Task hierarchy. When users click **More Help** from a help window, they get all the help files that are assigned to the same Task node as the page or section value.

To determine the help windows that a help file appears in:

- 1. Enter the Task hierarchy that's associated with the help window (as described above) to narrow down the list of available page or section values.
 - Note: For any help window for task lists or tasks in the Setup and Maintenance work area, select this hierarchy:
 - Hierarchy: Task
 - Level 1: Functional Setup
- 2. Select the page or section in the same row.

If you know the exact page or section you want, then you can select the value without entering a Task hierarchy. The associated hierarchy automatically fills in the rest of the row.



Adding Help to the Getting Started Region in Applications Help: Procedure

In Applications Help, users can use the tabs right below the global area to browse help by product family. Every product family tab has a Getting Started region, which contains guides and videos to help new users. Users can also find help topics for new users in specific roles. You can add your own custom help to the Getting Started region for any product family.

Adding Help to the Region for a Product Family

Follow these steps:

- 1. Click your user name in the global area in Applications Help, and select Manage Custom Help.
- 2. Create or edit the help file you want to add to the Getting Started region.
- 3. In the General Information section of the Create Help or Edit Help page, click the **Getting started** check box.
- 4. Make sure that the help file has either or both of the following:
 - Help type of PDF guide or Video
 - o One or more roles assigned
- 5. In the Help Location section, enter at least one row with **Product** in the Hierarchy column.
- 6. In the Level 1 column, select the product family that you want the file to appear under. Optionally select a product in the Level 2 column.
- 7. Save your work.

Links in Custom Help: Points to Consider

When you create or edit custom help, follow best practices when you include links to help files or other content. If you're duplicating a predefined help file, then you may see existing links. The types of links that you can work with include:

- Related help links
- Standard hypertext links
- Links to documentation library content
- Glossary term links

For all link types, except the standard hypertext links, you must create or edit custom help with a Text or Desktop File source type. For standard hypertext links, the source type can also be URL.

Related Help Links

Related help is the section at the end of help files that contains links to other help files. The syntax for related help contains a comma-separated list of title IDs that represent help files.

This figure provides an example of related links code.

OfaRelatedTopics(CREATE_AUTOMATIC_POSTING_CRITERIA_S_0000, JOURNAL ENTRIES HOW THEY RE RECORDE 0000)



In this example, the help file has two links to related help.

- To remove all related help, delete this code.
- To remove individual links, delete only title IDs (for example, create_automatic_posting_criteria_s_0000).
- To replace existing links or add new links, retain the code syntax and enter the right title IDs. To find title IDs, search for the help files on the Manage Custom Help page. Show the **Title ID** column in the search results if the column is hidden.

Standard Hypertext Links

You can create standard hypertext links to any file or Web site as long as you make sure that the links are valid and stable. These links can appear anywhere in the body of your help file as long as they come before any related help links.

If you're working on a help file with the Text source type:

- 1. In the Help Content section of the Create or Edit Help page, highlight what you want to use as link text.
- 2. Click the Add Link icon.
- 3. Enter the full URL, for example http://www.oracle.com.
- Tip: To find the URL for a help file that you want to link to, open that help file in Applications Help, and click the **Bookmark** link.

Links to Documentation Library Content

The syntax for links to HTML files in documentation libraries is:

<?ofa linkToSTDoc(WCSUG4636) ?>Understanding Tags<?ofa endLink ?>.

wcsug4636 is the anchor ID and understanding Tags is the link text. You can:

- Change the link by replacing the existing anchor ID, editing the link text, or both.
- Remove the link by deleting all the code for it.
- Create links to documentation library content by following the same syntax. These links can appear anywhere in the body of your help file as long as they come before any related help links.
- Note: To ensure that you're linking to a supported documentation library, enter anchor IDs only from documentation libraries that are linked from predefined help.

Glossary Term Links

Glossary term links provide definitions in a note box when users hover over the term in help files.

This figure shows an example of code for a glossary term link.

OfaGlossaryTerm("accounting period", ACCOUNTING_PERIOD_0001)

In this example, accounting period is the link text, or glossary term, and ACCOUNTING_PERIOD_001 is the identifier, or title ID.

To remove the link but retain the text, delete all the code except the term itself.



• To add glossary term links, you must follow the link syntax and use the correct title ID for the glossary term. To find title IDs, search for the glossary terms on the Manage Custom Help page. Show the **Title ID** column in the search results if the column is hidden.

If your help file has the Desktop File source type, then make sure before uploading that the quotes around glossary terms are actual quotation marks in raw HTML, not **LQUOT**. Otherwise, quotation marks will appear when users view the help file.

Customizing PDF Guides: Worked Example

This example demonstrates how to customize a PDF guide that came with Applications Help. This guide is currently not available from any help window.

The following table summarizes key decisions for this scenario.

Decisions to Consider	In This Example
What changes do you need to make to the guide?	Change the title of the guide and hide all the content that's not about a particular subject
Should the customized guide appear in any help window?	Yes, the help window for the Manage Administrator Profile Values page
Which products and tasks should the customized guide be assigned to?	Same as the original guide, plus the task associated with the help window
Do you want to limit access to the customized guide?	No, same as the original guide
Do you want to tag the customized guide with a role for the Role filter?	Yes, the Application Administrator or Implementor role

For this scenario:

- 1. Edit a copy of the original PDF guide.
- 2. Create a custom version of the original help file, with your new PDF as the help content.

Copying and Editing the PDF Guide

- 1. Open the original PDF guide in Applications Help and save a copy to your desktop. Leave the help file for the guide open.
- 2. Using a PDF editor application, change the title of the guide wherever it appears. Delete the content you want to hide from users.

Replacing the Original PDF Guide

- 1. In the help file that you still have open for the original PDF guide, click the **Edit** link.
- 2. On the Create Help page, use the default values except where indicated.
- 3. Update the title to the name that you want to display to users.
- 4. In the **File Name** field, browse for and select your customized guide.
- 5. Delete any keywords or parts of the description relevant to the content you removed from the PDF guide.
- 6. From the Roles list, select Application Administrator or Implementor.
- 7. Add a row in the Help Location table.
- 8. Click the icon in the **Details** column for the new row, and enter the following values.



Field	Value
Hierarchy	Task
Level 1	Functional Setup
Level 2	Perform Functional Setup
Level 3	Set Application Options
Level 4	Define Profiles

- 9. Click OK.
- **10.** Select **Manage Profile Option Values page** in the Page or Section column.
- 11. Click Save and Close. The help file for the original PDF guide is automatically set to inactive.

Adding Custom User Productivity Kit Content to Help: Worked Example

This example demonstrates how to add a custom Oracle User Productivity Kit topic as a video help file in Applications Help.

Note: Your topic must be made with User Productivity Kit 3.6.1 or later to be added as help.

The following table summarizes key decisions for this scenario.

Decisions to Consider	In This Example
What User Productivity Kit content do you want to add to help?	One topic from a module with five topics
Should the topic appear in any help window?	Yes, the one next to the Overview page title in the Scheduled Processes work area

For this scenario:

- 1. Generate a report of User Productivity Kit document IDs, to identify the User Productivity Kit topic when you create your help file.
- 2. Publish the module as a player package.
- 3. Create a custom help file for the User Productivity Kit topic.

Generating a User Productivity Kit Document ID Report

- 1. In the User Productivity Kit Developer, select **Details View**.
- 2. Right-click any column header, for example Name, and select Column Chooser.
- 3. In the Column Chooser dialog box, click and drag the Document ID column header and drop it after the Name column.
- 4. Close the Column Chooser dialog box.
- 5. From the File menu, select to print, and save the output as a Microsoft Excel file to your desktop.



Publishing the Module as a Player Package

- 1. From the Developer, make sure that the topic that you want to add as a help file has the See It play mode. The topic can also have other modes, but only the See It mode is included in the custom help file.
- 2. Publish the module, specifying any location for the output and selecting to publish the selection only.
- 3. In the Formats section of the Publish Content window, select the **Player** check box under the **Deployment** check box group.
- **4.** In the Player section, select the **Include HTML Web Site** check box. Your custom help file includes a text-only version of the User Productivity Kit topic.
- 5. Finish the publishing process, after optionally setting other options.
- **6.** Go to where you specified the output to be generated.
- 7. In the Publishing Content folder, copy the PlayerPackage folder and add it to the Web server where you store User Productivity Kit content.

Creating a Custom Help File for the Topic

- 1. Open the Scheduled Processes work area.
- 2. Open the help window for the Overview page, and click Manage Custom Help.
- 3. Click Create.
- **4.** In the Create Help dialog box, complete the fields as shown in this table.

Field	Value
Source	Oracle User Productivity Kit
File Location	The full URL of the player package folder on the Web server, for example, http:// <your domain="">. com/ MyContent/ PlayerPackage</your>
Document ID	The document ID of the User Productivity Kit topic to add to the help window on the Scheduled Processes Overview page. Copy and paste this ID from the Microsoft Excel file that you generated earlier.
Help Type	Video
Topic Title	The name of the User Productivity Kit topic.
Keywords	Terms relevant to the topic.
Description	Summary of the topic.

5. Click Save and Close.

Editing Glossary Terms: Procedure

You can edit the glossary terms in the Applications Help glossary. These are the same terms that users might find as links in help topics. The links provide definitions when users hover over the terms.



Procedure

Follow these steps:

- 1. In the global area of Applications Help, select **Glossary** from the Navigator.
- 2. Find the glossary term.
- 3. Click Edit.
- 4. Update the topic title (the term) or definition as needed.
- 5. Optionally change the assigned hierarchies.
 - o The last node of each hierarchy appears after the term's definition in the glossary.
 - o Glossary terms don't appear in the Task or Product filter.
- 6. Save your work.

Predefined Glossary Terms

When you edit a predefined glossary term:

- You're actually creating a copy, and the original term becomes inactive.
- The glossary in Applications Help shows your custom version.
- In help files, existing links to the glossary term automatically point to your custom version.
- Note: If you later inactivate your custom glossary term, make sure to activate the original term so that the links still work.

FAQs for Help File Customization

What happens when I edit predefined help?

You're actually creating a new custom help file based on the predefined file. The custom version replaces the original, which becomes inactive and hidden from users. You can display both versions by reactivating the original file.

How can I add a Youtube video to custom help?

Create a custom help file using the URL to the Youtube video.

- 1. Find the video in Youtube.
- 2. Click the Share button.
- 3. Click the **Embed** button.
- 4. Copy the URL within the embed code, for example http://www.youtube.com/embed/cunique ID>.
- 5. Open the Create Help page from a help window or from the help site.
- 6. Select **URL** as the source, or **Video URL** if you want the video to play within help windows.
- 7. Paste your copied URL in the URL field.
 - Tip: If you selected Video URL, change your pasted URL to start with https so that users don't need to adjust browser settings to play the video in the help window.
- 8. Enter other information, and click Save and Close.



How can I restrict access to specific help files?

You must create or edit help from the Manage Custom Help page.

- 1. In Applications Help, click your user name in the global area and select **Manage Custom Help**.
- 2. As you create or edit a help file, select a help security group, which represents a set of roles that have access to the help.

The predefined groups are:

- Unsecured: Anyone can view the help.
- Secured: All internal employees and contingent workers have access (unless this group was edited).

If you don't see the **Security Group** field, then you or an administrator can enable this feature choice in the Setup and Maintenance work area.

3. Save your work.

Related Topics

- Creating Help Security Groups: Worked Example
- Setting Up Help Customization: Procedure

What's the difference between assigning a role and a security group to a help file?

When users filter or browse for help files by role in Applications Help, they get the help files tagged with the role. The help security group hides the help file completely from users who don't have the roles defined in the group.

Why can't users find my custom help in their search results?

If you customized those help files recently, they might not be indexed yet for the search in Applications Help. The indexing process runs on a defined schedule. Users can still find your help files by browsing, for example using the Task or Product filter.

What happens to my custom help after an upgrade for Applications Help?

Nothing happens to your custom help files. Upgrades affect only predefined help files, active or inactive.

Take a look at any inactive file that's updated to see if you want to:

- Activate the updated version.
- Make similar edits to the custom version of that file, if any.

How can I change the background image on the Applications Help home page?

In the Setup and Maintenance work area, open the Set Help Options task, and upload your own image in the Help Site Customization section. Use an image that's white along the entire left border, like you see in the default image.

Note: If you don't see the Help Site Customization section on the Set Help Options page, check with your implementor about selecting the Help Customization feature choice.

Customization of Getting Started Pages



Overview

The Getting Started work area provides pages of information to introduce new users to the application. If you have multiple cloud services, you get a separate set of pages for each service. You can customize the predefined pages or add your own, so that users get content specific to your organization.

Access to Customization

Open the Getting Started work area, and, if you have multiple sets of pages, select the set to customize. You then see the **Edit Getting Started** link if you have the appropriate roles.

Key Customization Tasks

You can:

- Create and edit pages using a rich text or HTML source code editor
- Reorder the pages
- Activate or inactivate any page
 - ▼ Tip: To hide the link to a set of Getting Started pages, inactivate all pages within that set. Users with access to edit Getting Started pages can still see the link, but everyone else can't. If you inactivate all sets except one, then users land on the first page of the active set when they open the Getting Started work area.
- Delete custom pages (not predefined ones)

How can I add a YouTube video to a Getting Started page?

Create or edit a page in the Getting Started work area, and include a piece of code from YouTube.

- 1. Find the video in YouTube.
- 2. Click the Share button.
- 3. Click the Embed button.
- 4. Copy everything in the text box.
- 5. Back in the application, open the Getting Started work area.
- 6. If you have multiple sets of Getting Started pages, select the set you want to add the video to.
- 7. Click the Edit Getting Started link.
- 8. Open an existing Getting Started page or create a new one.
- 9. Click Source Code Editing Mode in the toolbar.
- 10. Paste in the code you copied from YouTube.
- 11. Click Save and Close.

Embedded Help Customization



Customizing Help That Appears on the Page: Highlights

You can customize help that you see on the page, for example hints for check boxes or text in help windows. There are different types of such embedded help. Embedded help doesn't include help that you open using links in help windows, or help that you find in Applications Help.

Creating, Editing, or Deleting Embedded Help

- Use Page Composer to edit, create, or delete hint text that appears on hover over buttons, links, icons, or tab titles. Open the properties of the UI element to define the help text in the shortDesc field.
- Use the User Interface Text tool to edit the text for any type of embedded help, including informational text in help windows. You usually use this tool to make bulk changes, for example to change a phrase wherever it appears in any UI label, embedded help, messages, and so on.
- Edit, create, or delete most types of embedded help using design time tools (not available in Oracle Cloud implementations). Refer to the Customizing or Adding Static Instructions, In-Field Notes, and Terminology Definitions section.

See: Oracle Fusion Applications Extensibility Guide for Developers

Related Topics

· Customizing Simplified Pages Using Page Composer: Procedure

• Page Component Properties: Explained

Bulk Text Customizations: Explained



Glossary

accounting flexfield

The structure that determines the chart of accounts, including the number and order of the individual segments, as well as assigning the value sets to the segments.

action

The kind of access, such as view or edit, named in a security policy.

ADF

Acronym for Application Developer Framework. A set of programming principles and rules for developing software applications.

analysis

A selection of data displayed in one or more views, such as a table or chart, to provide answers to business questions.

analytics

Business intelligence objects such as analyses and dashboards that provide meaningful data to help with decision making.

application identity

Predefined application level user with elevated privileges. An application identity authorizes jobs and transactions for which other users are not authorized, such as a payroll run authorized to access a taxpayer ID while the user who initiated the job is not authorized to access such personally identifiable information.

business object

A resource in an enterprise database, such as an invoice or purchase order.

chrome

The set of visual elements (for example, header; expand and edit icons) around the perimeter of a component or task flow that enables users to act directly on the object.

context

A grouping of flexfield segments to store related information.

context segment

The flexfield segment used to store the context value. Each context value can be associated with a different set of context-sensitive segments.



context-sensitive segment

A flexfield segment that may or may not appear depending upon a context. Context-sensitive segments are custom attributes that apply to certain entity rows based on the value of the context segment.

customization

A change to the predefined artifacts of the application. Customizations impact multiple users.

customization layer

A level that represents the scope of users that a customization impacts. For example, all users or only those who meet specific criteria.

dashboard

A page that provides quick access to key tasks and summary information for various objects within a functional area of interest.

dashboard

A collection of analyses and other content, presented on one or more pages to help users achieve specific business goals. Each page is a separate tab within the dashboard.

data security

The control of access and action a user can take against which data.

descriptive flexfield

Customizable expansion space, such as fields used to capture additional descriptive information or attributes about an entity, such as a customer case. You may configure information collection and storage based on the context.

design time

The type of activities that developers perform at the code or data model level.

desktop page

A page that's optimized for extended periods of use with monitors.

desktop user interface

A user interface that's optimized for extended periods of use with monitors.

detailed report

A comprehensive report that provides detailed information about the subject matter. When you link a detailed report to an infolet, users can click anywhere in the infolet area to drill down to that detailed report.



enterprise

An organization with one or more legal entities under common control.

extensible flexfield

Customizable expansion space used to capture multiple sets of information within a context or multiple contexts. Some extensible flexfields let you group contexts into categories.

extension

A new artifact in addition to what's predefined in the application, for example a new business object or page.

feature choice

A selection you make when configuring offerings that modifies a setup task list, or a setup page, or both.

flexfield

A flexible data field that you can customize to contain one or more segments or store additional information. Each segment has a value and a meaning.

flexfield segment

An extensible data field that represents an attribute and captures a value corresponding to a predefined, single extension column in the database. A segment appears globally or based on a context of other captured information.

global area

The region at the very top of the user interface that remains the same no matter which page you're on.

infolet

A small, interactive widget on the home page that provides key information and actions for a specific area, for example social networking or your personal profile. Each infolet can have multiple views.

job role

A role, such as an accounts payable manager or application implementation consultant, that usually identifies and aggregates the duties or responsibilities that make up the job.

key flexfield

Configurable flexfield comprising multiple parts or segments, each of which has a meaning either individually or in combination with other segments. Examples of key flexfields are part numbers, asset category, and accounts in the chart of accounts.

key flexfield segment instance

A single occurrence of a key flexfield segment in a key flexfield structure instance.

key flexfield structure

The arrangement of segments in a key flexfield. In some cases, you can define multiple structures for a single key flexfield.



key flexfield structure instance

An occurrence of a key flexfield structure that shares the same order of segments as other instances of the key flexfield structure. However, each instance uses different value sets to validate the segments.

mainline metadata

The primary branch of metadata that a sandbox is published to. Once published, changes made in the sandbox become available to all users.

Navigator

The menu in the global area that you can use to open the work areas and dashboards that you have access to.

offering

A comprehensive grouping of business functions, such as Sales or Product Management, that is delivered as a unit to support one or more business processes.

performance tile report

A summary report that shows data in the small infolet format. When you add a performance tile report to an infolet, users can see summary information about the subject matter.

personalization

A change that users make to control the look or behavior of the application. Personalizations impact only the user making the change.

privilege

A grant of access to functions and data; a single, real world action on a single business object.

report

An output of select data in a predefined format that's optimized for printing.

role

Controls access to application functions and data.

run time

The type of activities that users perform while they are in a running application.

sandbox

A testing environment that isolates untested code changes from the mainline environment so that these changes don't affect the mainline metadata or other sandboxes.



scheduled process

A program that you run to process data and, in some cases, generate output as a report.

segment

A segment is a single field within a flexfield and maps to a single table column in your database. When customizing a flexfield, you define the appearance and meaning of individual segments.

simplified page

A page that's optimized for performing quick and frequent tasks on any device.

simplified user interface

A user interface that's optimized for performing quick and frequent tasks on any device.

site layer

Customizations made in this layer affect all users.

springboard

The grid of icons on the home page or the strip of icons above all simplified pages. Use the icons to open pages.

task flow infolet

An infolet that displays summary information about a task.

tree

Information or data organized into a hierarchy with one or more root nodes connected to branches of nodes. A tree must have a structure where each node corresponds to data from one or more data sources.

tree structure

A set of guidelines or a framework applied to create a tree, include data, version a tree, or access a tree.

user layer

Customizations made in this layer affect only the user making the change.

value set

A set of valid values against which values entered by an end user are validated. The set may be tree structured (hierarchical).

work area

A set of pages containing the tasks, searches, and other content you need to accomplish a business goal.



