Oracle Fusion Cloud Applications

Configuring and Extending Applications

26D

Oracle Fusion Cloud Applications
Configuring and Extending Applications

26D

G59634-02

Copyright © 2011, 2026, Oracle and/or its affiliates.

Author: Barnali Roy, Essan Ni Jirman, Rebecca Nethala, Kalyani Manda

Contents

Get Help i

1 Overview 1
Overview of Application Configuration and Extension 1
Examples of Configurations and Extensions, and the Tools to Use 3
Personalization 6
Configuration Life Cycle 7
Context Layers 10

2 Sandboxes 19
Overview of Sandboxes 19
Create and Activate Sandboxes 20
Add Tools to Existing Sandboxes 21
Delete Sandboxes 22
Best Practices for Using Sandboxes 22
Best Practices for Using Page Composer in Sandboxes 23
How to Resolve Conflicts in Sandboxes 23
How the Refresh and Merge Processes Work in Sandboxes 25
Options to Open Configuration Tools from Sandboxes UI 26
Object Locking in Application Composer 28
Create Backups of Sandbox Configurations 29
Publish Sandboxes 29
FAQs for Sandboxes 30

3 Migration and Troubleshooting 37
Tools for Moving and Troubleshooting Configurations 37
Migration 38
Troubleshooting Configurations 56

4 Page Content and Layout 59
Overview of Page Modification 59

Overview of Using Page Composer 59
Overview of Using Visual Builder Studio 64
Overview of Using Application Composer 66
Differences Between Using Page Composer and Application Composer 66
Modify Page Content and Layout Using Page Composer 67
Modify Global Page Template Using Page Template Composer 79
Saved Searches 81
Infolets 85
Configure Infotiles on a Page 89
Pages Hosting Third-Party Applications 89

5 User Interface Text 95
Tools for Changing Text 95
How do I modify text using the User Interface Text tool? 97
Sample Search Values for User Interface Text Modification 98
Overview of Translating Modified Text 100
Translate Existing Strings Added Using Configuration Tools 100
Translate New Strings Added Using Configuration Tools 101
Translate Strings Offline 102
Translate New Strings Offline From English to German 103
FAQs for User Interface Text 105

6 Themes 107
What themes can I use in Fusion Applications? 107
How do I create Redwood themes for Fusion Applications? 107
How do I configure classic themes for Fusion Applications? 107
FAQs for Themes 108

7 Flexfields 109
Overview of Flexfields 109
Overview of Flexfield Configuration 110
Flexfield Components 111
Flexfields at Runtime 113
Modify Flexfields Using Page Composer 114
How Flexfields Work with Oracle Application Cloud Architecture 117
Flexfield Management 119

Flexfield Deployment 138
Value Sets 143
Descriptive Flexfields 171
Extensible Flexfields 189
Key Flexfields 214

8 Configure Navigation 235
Get Started with Configuring Navigation 235
Create and Edit Groups 235
Create and Edit Page Entries 236
Create and Edit Quick Actions 237
Configure Links for Page Entries and Quick Actions 238
Reorder Navigation Entries 242
Delete Custom Groups and Page Entries 243
What are the supported and unsupported EL expressions? 243
How do I resolve unsupported EL expressions? 248
FAQs for Navigation 248
Deep Links 249

9 Help Content 251
How You Manage Different Types of Help 251
Manage Help for Fields and Other UI Elements 252
Why can't I see the Manage Help Content or Edit Getting Started link? 252
Help Windows 253
Getting Started Work Area 255
All Added Help 257

10 Guided Learning 261
Manage Guided Learning for Onboarding and Training 261

11 Oracle Fusion Cloud Applications Identity Domain 263
Overview of Using Oracle Fusion Cloud Applications Identity Domain to Build Extensions 263
Provision a Oracle Cloud Integration Service into the Fusion Applications Identity Domain 264
Configure Fusion Applications Identity Domain Federated with an External IdP for OCI Extensions 267
Manage Users in the Fusion Applications Identity Domain 268
Extension Use Cases with the Fusion Applications Identity Domain 269

Configure OAuth to Access Oracle Fusion Cloud Applications REST APIs 269
Production to Test Considerations for Extensions with Fusion Applications Identity Domain 286

Get Help

In addition to the resources available in Oracle Help Center, you can explore training, connect with the Oracle community, and share your feedback.

Get Training

Increase your knowledge of Oracle Cloud by taking courses at *Oracle University*.

Join Our Community

Use *Cloud Customer Connect* to get information from industry experts at Oracle and in the partner community. You can join forums to connect with other customers, post questions, suggest *ideas* for product enhancements, and watch events.

Share Your Feedback

We welcome your feedback about Oracle Applications user assistance. If you need clarification, find an error, or just want to tell us what you found helpful, we'd like to hear from you.

You can email your feedback to *oracle_fusion_applications_help_ww_grp@oracle.com*.

Thanks for helping us improve our user assistance!

1 Overview

Overview of Application Configuration and Extension

By default, the application provides robust functionality, tailored to support most of the business requirements an organization could have. But you can still make changes to your application to best fit your specific business or personal needs.

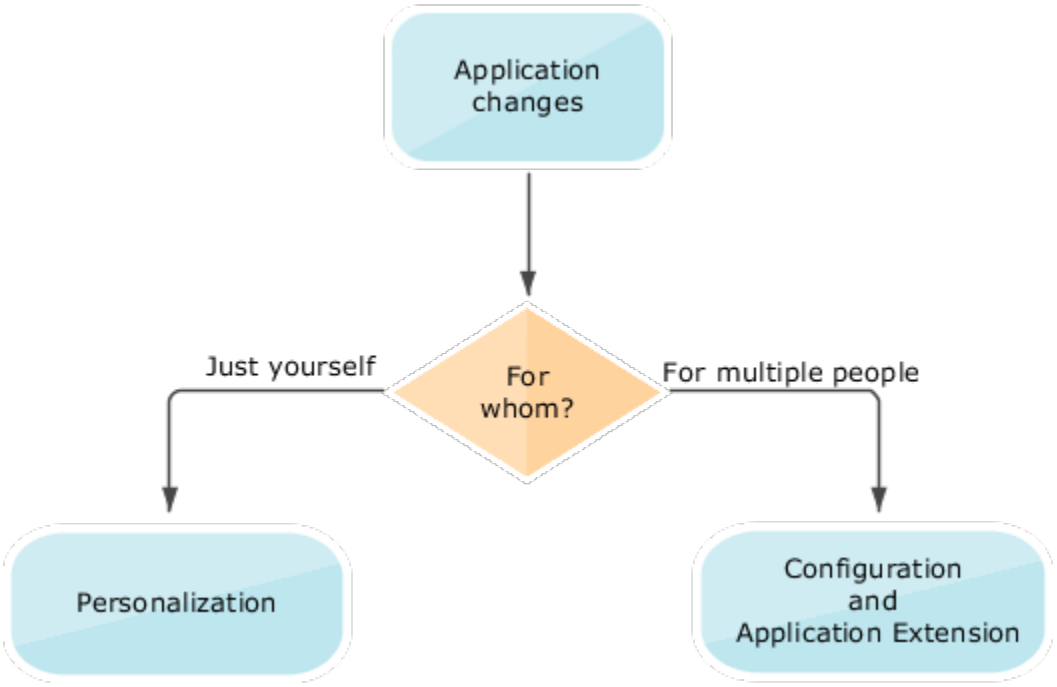
Types of Changes

The types of change you can make depend on whether you're an administrator or a user, the pages you're changing, and who you're making the changes for.

- Configuration: These changes are made by administrators, and such changes affect many users. For example, you might hide some menu items in the Navigator, or create a new page for all users.
- Extensions: These are modifications applied to Oracle Fusion Cloud Applications at runtime using Visual Builder Studio. Extensions are made by one or more application developers and administrators, and such changes affect specific users based on the conditions defined for the extensions. These conditions are evaluated during runtime, determining which users can see the changes. For example, you may want to add a new field to a table or rearrange the fields in a form, but only when the user has a certain role. In most cases, the work for each product family (such as Oracle Fusion Cloud HCM, Oracle Fusion Cloud SCM, and so on) is separated into its own extensions, to which many people contribute.
- Personalization: These changes are made by individual users, and they affect only the users who made the changes. As an administrator, you can't make personalizations for any specific user. Personalizations are also limited to certain types of changes, such as hiding infolets or resizing table columns.

Note: Configurations, extensions, and personalizations are preserved when you move on to a newer release update.

Here's a visual representation of how changes are categorized.



What You Can Change

When you make a change to the data model, that change is available to other aspects of the application. For example, after you add an attribute to an object, you can use that attribute in these related artifacts:

- Web-based view page
- Associated mobile page
- Associated reports

Generally, you use Application Composer to change the data model. You can then reflect those changes in the UI by using Visual Builder Studio, Page Composer, or Application Composer. If you see the **Edit Page in Visual Builder Studio** option in the **Settings and Actions** menu of a page, you know that you must use Visual Builder Studio to edit that page.

Note: Click your user image or name in the global header to open the **Settings and Actions** menu.

For information about configuring business intelligence, see the Creating and Editing Analytics and Reports guides relevant to your products.

Related Topics

- Personalization
- Examples of Configurations and Extensions, and the Tools to Use
- Overview of Context Layers
- Configuration Life Cycle

Examples of Configurations and Extensions, and the Tools to Use

You can configure and extend the application using browser-based composers and other tools. Some configuration tools, such as Application Composer and Visual Builder Studio, are available only for specific product families.

Let's look at some of the key things you can do with these tools. Not all tasks are listed here.

Modify the UI

Use these tools and work areas to modify the UI:

- Page Composer: Configure application page components, such as page content, layout, and more for other users.
- Visual Builder Studio: Extend application page components for certain applications.
- Application Composer: Create fields, objects, and relationships between objects.
- User Interface Text: Edit text that appears on multiple pages.
- Appearance: Change the look and feel of the application pages.
- Page Template Composer: Edit global page template.
- Structure: Configure the Navigator and the icons on the home page for navigation.
- Announcement: Create, edit, and delete announcements displayed on the home page.

Here are some changes that you can make to pages, and the corresponding tools or work areas to use. You can modify certain pages only in either Page Composer or Visual Builder Studio.

Task	Tool or Work Area
Add, move, delete, show, or hide components on a page, including fields	Page Composer or Visual Builder Studio
Add validation to a field	Page Composer
Add external content	Page Composer
Change a page layout	Page Composer or Visual Builder Studio
Add business logic to display certain layouts	Visual Builder Studio
Manage saved searches	Page Composer
Modify dialog box content	Page Composer or Visual Builder Studio

Task	Tool or Work Area
Modify attributes for a flexfield on a page	Page Composer
Change properties for UI components on a standard page	Page Composer or Visual Builder Studio
Edit variables, constants, and events as needed	Visual Builder Studio
Configure the UI Shell template	Page Template Composer
Update a UI text across all pages, such as to change "buyer" to "customer" on all pages where the term "buyer" is used	User Interface Text
Change the look and feel of application pages	Appearance

Tip: In Page Composer, you can make changes using the WYSIWYG view. However, in some cases, you can also use the Source view.

Update the Branding

This table shows examples of where you can change the branding to your own logo, and the corresponding tools or work areas to use.

Task	Tool
Change the logo in the global header	Appearance
Change the logo that appears in report output	Layout editor in the BI application or external applications, such as Microsoft Word

Configure the Navigator and Icons on the Home Page for Navigation

Use the Navigation Configuration page in the Structure work area to configure the Navigator and the work area icons or Apps section on the home page. For example, you can show or hide certain items and reorganize what's there. You can also add a business object to the Navigator.

Configure the Home Page

Here are some things you can change on the home page, and the work area you use to make those changes.

Task	Work Area
Change the layout of the home page	Appearance
Configure the icons for infolet pages in the page control on the home page. You can rename these icons, change their visibility settings, and reorder them.	Home Configuration page in the Structure work area
Create, edit, and delete announcements on the home page.	Announcements

Add Attributes to Business Objects

You can use flexfield tasks in the Setup and Maintenance work area to add user-defined attributes to most business objects, except for those in Oracle Fusion Cloud Sales Automation and Oracle Fusion Cloud B2B Service products. With flexfields, you create your own attributes without coding. A flexfield captures data for a specific purpose, such as information about job positions or inventory items. Each attribute is a segment of a flexfield, and corresponds to a reserved column in the application database.

For Oracle Automation, Oracle B2B Service, and other products, you can use Application Composer to add attributes to business objects or create custom objects so that you can track and store any additional data you might need. For example, you can add new attributes (also referred to as custom fields) to an existing object, or create entirely new custom objects. There's no fixed limit on the number of custom objects that you can create. After you create an object, you can also edit its attributes.

Tip: You use the custom object's security node in Application Composer to secure custom object data.

Modify Reports and Analytics

Predefined analyses, dashboards, and reports help you meet business intelligence (BI) requirements. But you can still modify the reports and analytics to fit specific business needs. You start in the Reports and Analytics work area, and go from there to do more in the BI application. Let's see some examples of the changes you can make, and the corresponding tools or work areas to use.

Task	Tool or Work Area
Create an analysis	Reports and Analytics work area or the BI application
Create or change report layout	Layout editor in the BI application or external applications, such as Microsoft Word
Add more pages, or tabs, to dashboards	The BI application

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

Manage Help Content

Many pages have help icons that you click to open help windows. Here, you can add your own content, such as your company policies or best practices. You can also find help on some UI elements, such as hints that appear when you click in a field.

This table shows a few examples of changes that you can make to help, and the corresponding tools to use.

Task	Tool or Work Area
Add help that users can see in help windows	Help windows
View and edit all of the help that anyone added to help windows	**Manage Help Content** task in the Setup and Maintenance work area
Modify text that's displayed when the user hovers over UI elements, such as a button, link, icon, or tab title	Page Composer
Simultaneously replace multiple occurrences of a word or phrase that appears in the help for UI elements	User Interface Text

Note: You must have the appropriate privileges to add and edit help.

Related Topics

- Guidelines for Using Page Composer
- Overview of Using Visual Builder Studio
- Overview of Flexfields
- How You Manage Different Types of Help

Personalization

Personalization refers to the changes that every user of the application can make to certain artifacts in the UI.

Personalization applies only to the user who made them. They remain every time the user signs out of the application and signs back in.

Here are some examples of personalizations, specific to a user:

- How they use the UI, such as changes to the column width of a table

  > **Note:** Sometimes changes to the column widths of tables remain only while you're on the page. If you navigate away and then return to the page, the column widths are restored to their original sizes. However, the application behavior may vary in some cases.

- What they want to save, such as search parameters
- What they want to see, such as the work area icons to show or hide on the home page

Related Topics

- Examples of Configurations and Extensions, and the Tools to Use

Configuration Life Cycle

You must make all configurations and extensions, including the extensions made using Visual Builder Studio, in a test environment. Then you migrate them to the production environment, which is the one you use for your everyday business.

You must never make these changes directly in the application instance you use for your business needs. This is very important because any incomplete, flawed, or invalidated configuration you make can disrupt your business.

You must create and enter a sandbox before you can start using configuration tools, such as Page Composer and Application Composer, to modify your application. Changes you make with these tools are stored in the sandbox as XML files. These changes are then merged with the mainline metadata when you publish your sandbox.

> **Note:** Changes you make in one sandbox aren't available in the other sandbox. Also, if multiple users work on the same sandbox, then you must follow the prescribed guidelines to avoid conflicts within a sandbox.

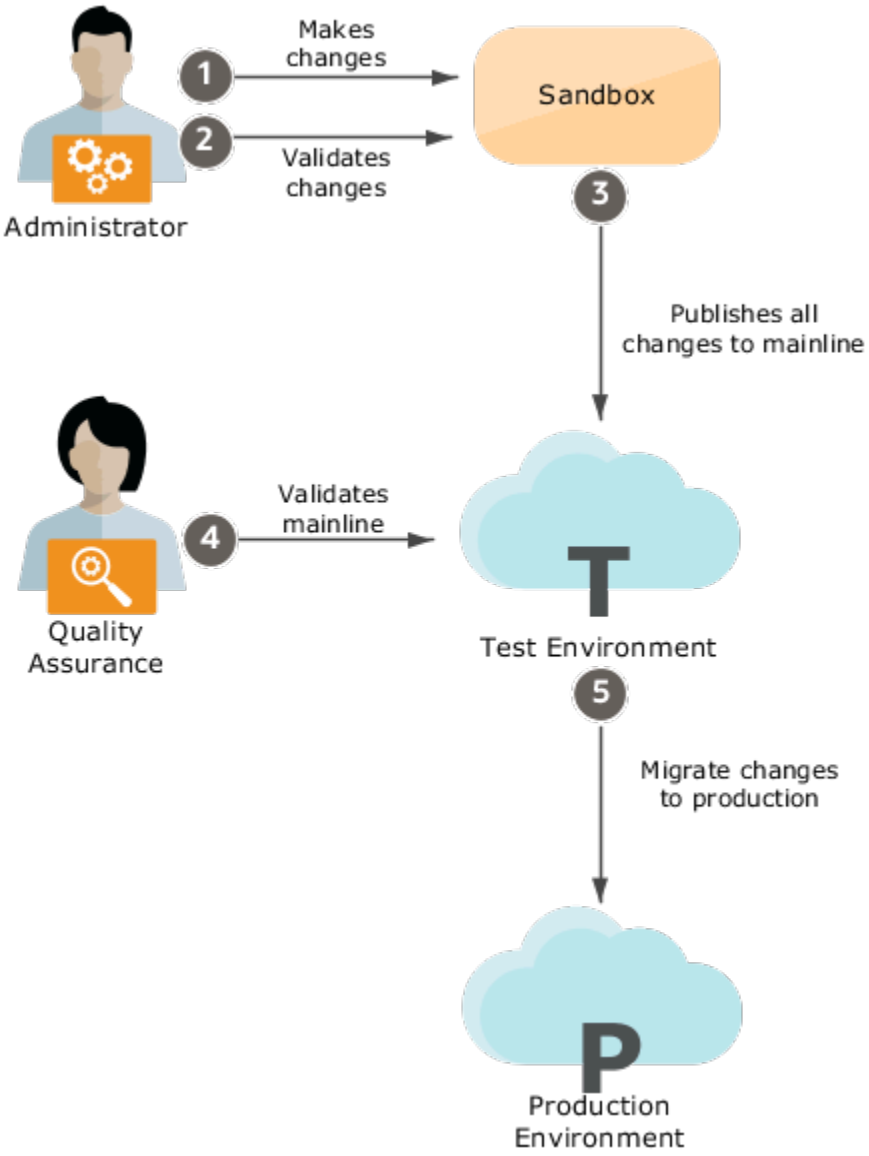
A sandbox is optional if you're using Visual Builder Studio to modify your application. You should use a sandbox only if you require any changes to the underlying data model.

A flexfield sandbox is for testing only and can't be published. Instead, you can deploy a flexfield to the test environment using the flexfield UI. To test a flexfield configuration before you deploy it to the test environment, you must deploy it to a flexfield sandbox. The changes that you deploy to a sandbox are isolated from the test environment. Users who make the flexfield sandbox active in their session can only see your changes. After you're satisfied with the changes in the sandbox, you can deploy the changes to the test environment.

Here are a few typical configuration life cycle examples.

Application Configuration Using Configuration Tools

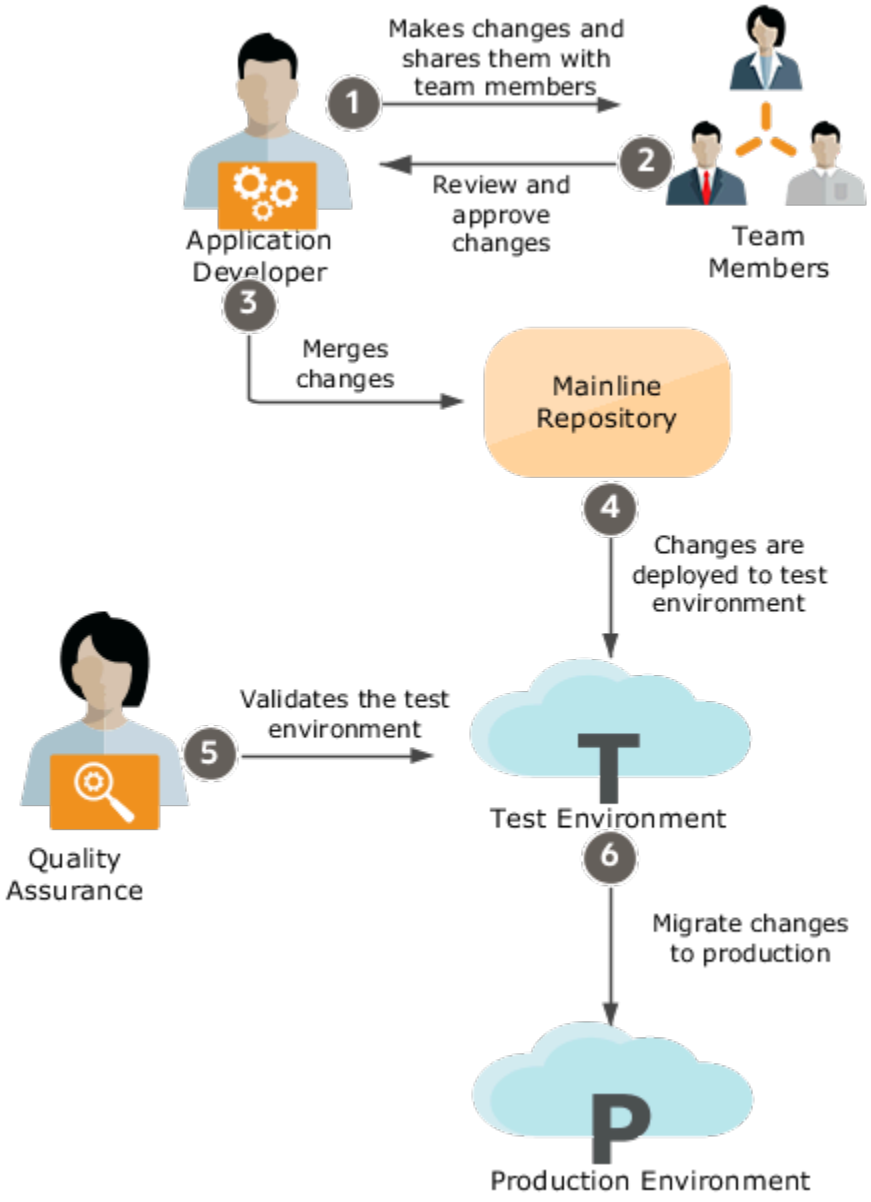
1. You configure the application in a sandbox.
2. You validate the changes in the sandbox.
3. The sandbox is published and deployed to a test environment.
4. Quality Assurance validates the whole environment after all configurations are completed and published.
5. Configurations are migrated from the test environment to the production environment.



Application Extension Using Visual Builder Studio

1. An application developer uses Visual Builder Studio to make application changes and shares the changes with other team members.

   Optionally, the application developer may use a sandbox to make changes to the underlying data model, which then need to be published along with the deployment of the application extension changes.
2. Team members review and approve the changes.
3. The reviewed and approved changes are merged with the mainline repository.
4. Once the application extension changes are merged, a build is triggered, and the changes are automatically deployed to a test environment.
5. Quality Assurance validates the whole environment after all extensions are deployed to a test environment.
6. Changes are migrated from the test environment to the production environment.



Related Topics

- View and Diagnose Application Changes
- Tools for Moving and Troubleshooting Configurations
- Download Configurations

Context Layers

Overview of Context Layers

You can use context layers to control which specific set of users your application changes apply to. Make sure you use a context layer to make application changes and set your sandbox to the correct context layer.

A sandbox is a testing environment that sets apart untested changes from the mainline environment so that these changes don't affect the mainline metadata or other users in the environment. You set the context layer while creating your sandbox. If you select a context layer that's not Site, make sure the context layer of your sandbox is supported by the page you want to edit. Otherwise, you won't be able to edit the page. You must also activate the configuration tools you want to use in your sandbox. The context layers for all selected tools are set as Site by default. So the changes you make using these tools affect all users. If you plan to use Page Composer in your sandbox and edit pages at a layer other than Site, you need to create a sandbox just for that layer and activate only Page Composer in it.

> **Note:** Changes made using the User Interface Text tool apply to all layers, not just the Site layer. Suppose you create a sandbox named SandboxA, activate the Page Composer tool, and set the context layer to the Job Role layer. Then you create a new string, **Program Choice** using the Page Composer tool in SandboxA and publish it. After that, suppose you create another sandbox named SandboxB and activate the User Interface Text tool in it. Using this tool, you replace the word, **Program Choice** with **Campaign** in SandboxB, and publish it. In this case, the string, **Program Choice** is changed to **Campaign** in all layers, including the Job Role layer.

Available Layers

Different product families have different context layers. But, every application has these layers:

- Site: Changes made in this layer affect all users of the application.
- User: Changes made in this layer affect just one specific user. But, you can't use this layer to make changes for other users. Personalizations are stored in this layer, and users can make personalizations only for themselves.

Here are the layers you can use when you configure different product families:

- Customer Relationship Management
    - Site
    - External or Internal
    - Job Role
- Human Capital Management
    - Site
    - Country
    - Organization
    - Time Card Layout

- Default Layer
    - Site

How Layers Work

Layers are applied in a specific order.

1. User
2. External or Internal
3. Job Role
4. Site

You won't see changes that were set at a layer that's higher in the hierarchy. The higher the level, the more specific its context is. Layers at higher levels exist within the scope of the layers below them. This means that when a value is set for a context layer, it's set within the scope of the context value for the layers below it.

You can have multiple configurations for an object or a page at the same time, but this is possible only when either of these conditions are met:

- Configurations are in different layers
- Configurations are in the same layer, but the layers have different context values

When a user opens an object or a page, the configurations at the highest layer applicable to an object or a page are given precedence. Let's say you added three columns to a table and then configured these columns for each context layer in this way:

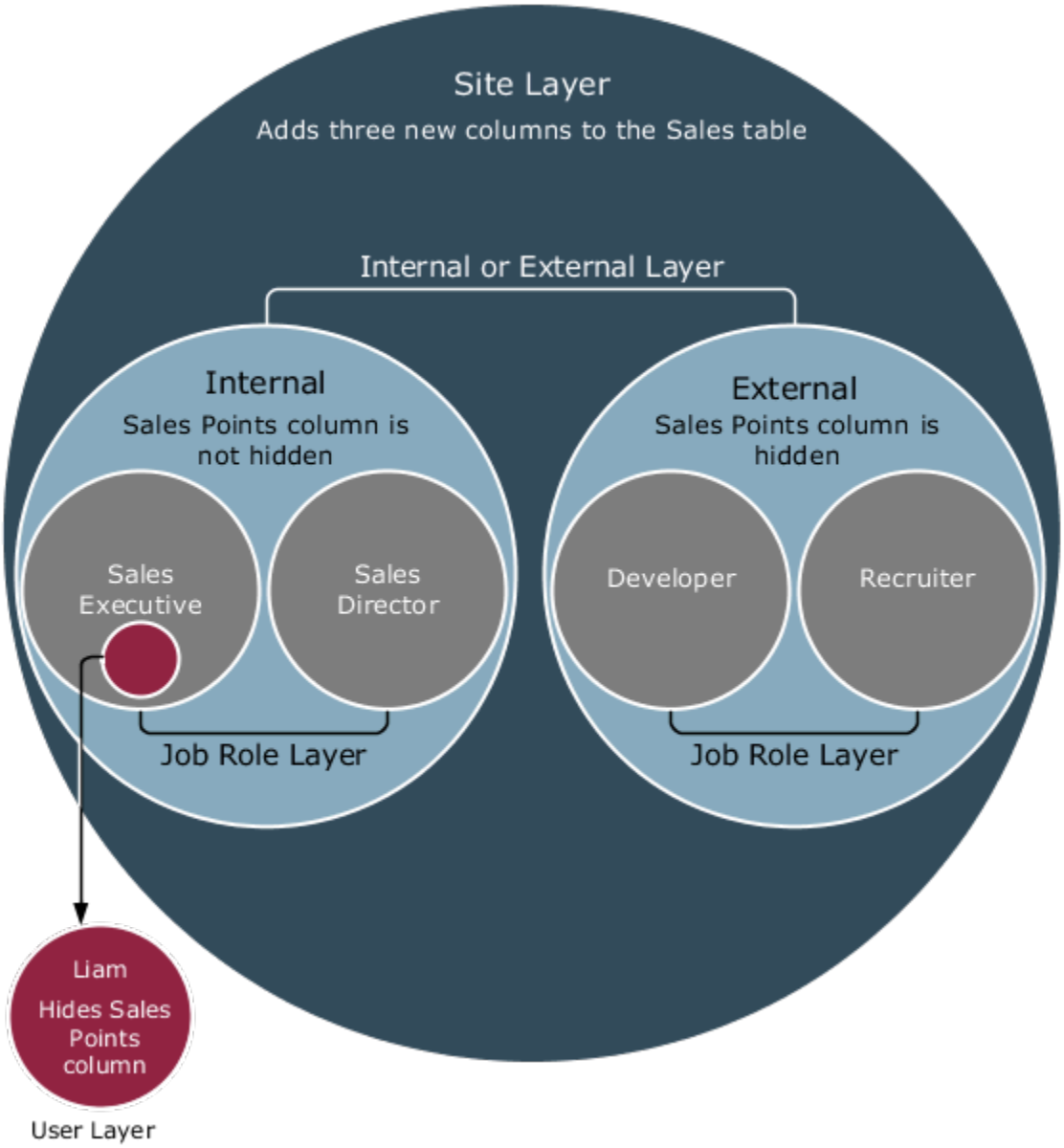
1. At the Site layer, you added three columns to the Sales table, namely, Promotion Name, Sales Points, and Partner Name.
2. For the external developer, the Sales Points column is hidden.
3. For the internal sales role, the Sales Points column isn't hidden.

Liam, who's an internal sales representative, personalizes this page for himself and hides the Sales Points column.

This is how different users see the column:

User	Sees Sales Points Column?
Internal sales role	Yes
External developer	No
External recruiter	No
Liam	No

Here's a visual representation of these configurations in different context layers and values.



Related Topics

- Examples of How the Appropriate Configurations and Personalizations are Applied
- Example of Including Higher Context Layers While You Configure

Examples of How the Appropriate Configurations and Personalizations are Applied

Here are a couple of scenarios that show you what happens behind the scenes with context layers so that the appropriate configurations and personalizations are available to the appropriate users.

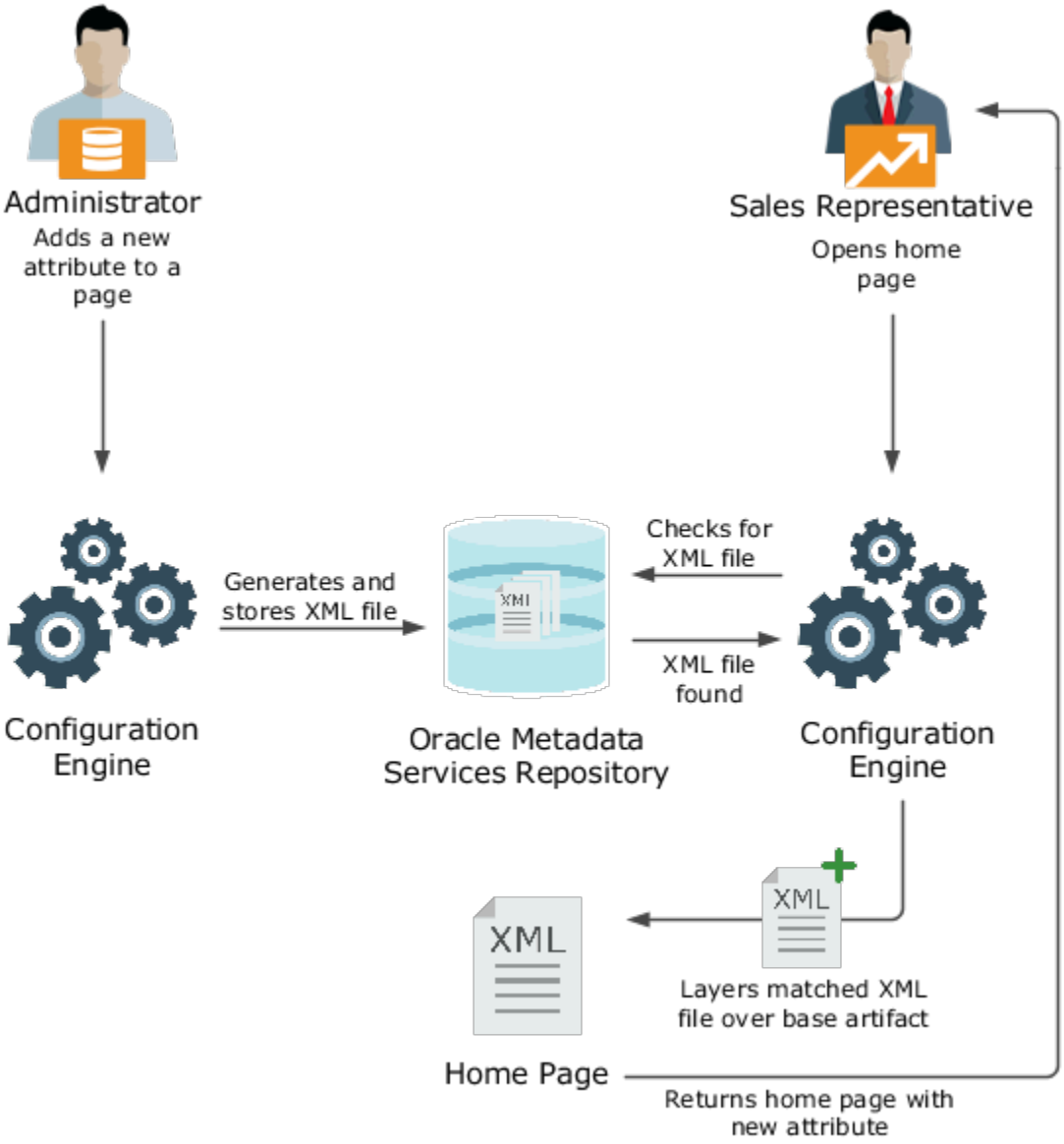
An Administrator Configures a Page for Sales Representatives

Let's say you're an administrator who wants to remove the **Export** button from the home page, but you want to remove it only for sales representatives. You select the Job Role context layer with the value **Sales Representative**, and then remove the button.

This is what happens after you make this change:

1. The configuration engine in Oracle Metadata Services generates an XML file and then stores it in the Oracle Metadata Services repository. So the original file for the page remains untouched.
2. When a sales representative opens the home page, the configuration engine checks the repository for XML files that satisfy these two conditions:
    - Does the file match the requested artifact, in this case, the home page?
    - Does the file match the active context, in this case, the Sales Representative job role?
3. The configuration engine also looks for any XML files with personalizations that this specific sales representative made. But for now, let's assume our sales representative hasn't personalized anything.
4. After the configuration engine finds an XML file that satisfies both conditions, that XML file is then layered over the base artifact, the original Sales page. In this scenario, the XML file removes the **Export** button from your sales representative's page.

Here's an image that shows you how this configuration is done.



A Recruiter Personalizes a Page

Let's say you added three new work area icons A, B, and C to the home page. You added icons A and B at the Site layer for all users and icon C at the Job Role layer just for recruiters. But Liam, who's a recruiter, doesn't use icon B all that much. He can personalize his home page to show only icons A and C. When Liam does this, an XML file is generated for Liam's user layer with this change.

The next time Liam opens his home page, the configuration engine retrieves three XML files:

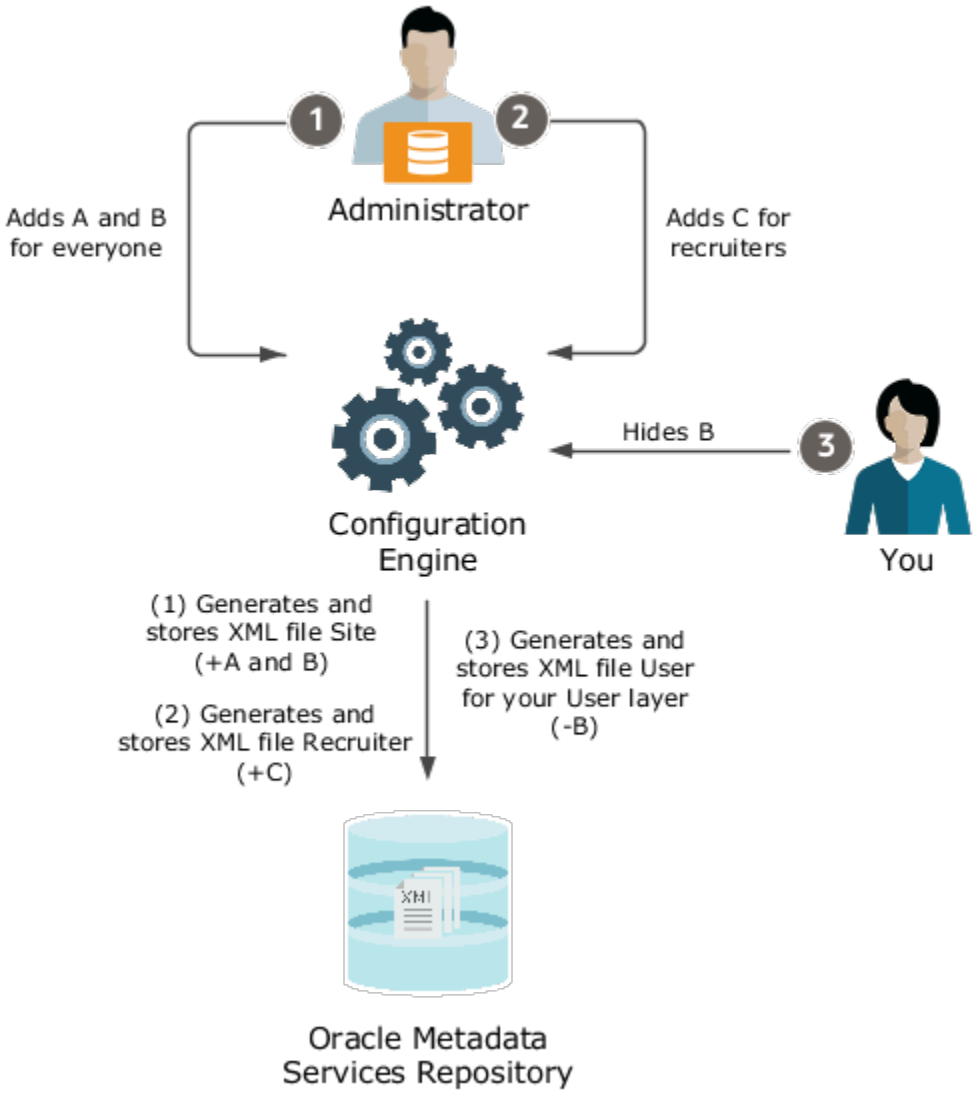
- The file for the changes you made at the **Site** layer
- The file for the changes you made at the **Job Role** layer
- The file for the personalization Liam made

The file for the context with the lowest layer (Site) is applied first, and the file for the context with the highest layer (User) is applied last. In this scenario, this is what happens when each file is applied.

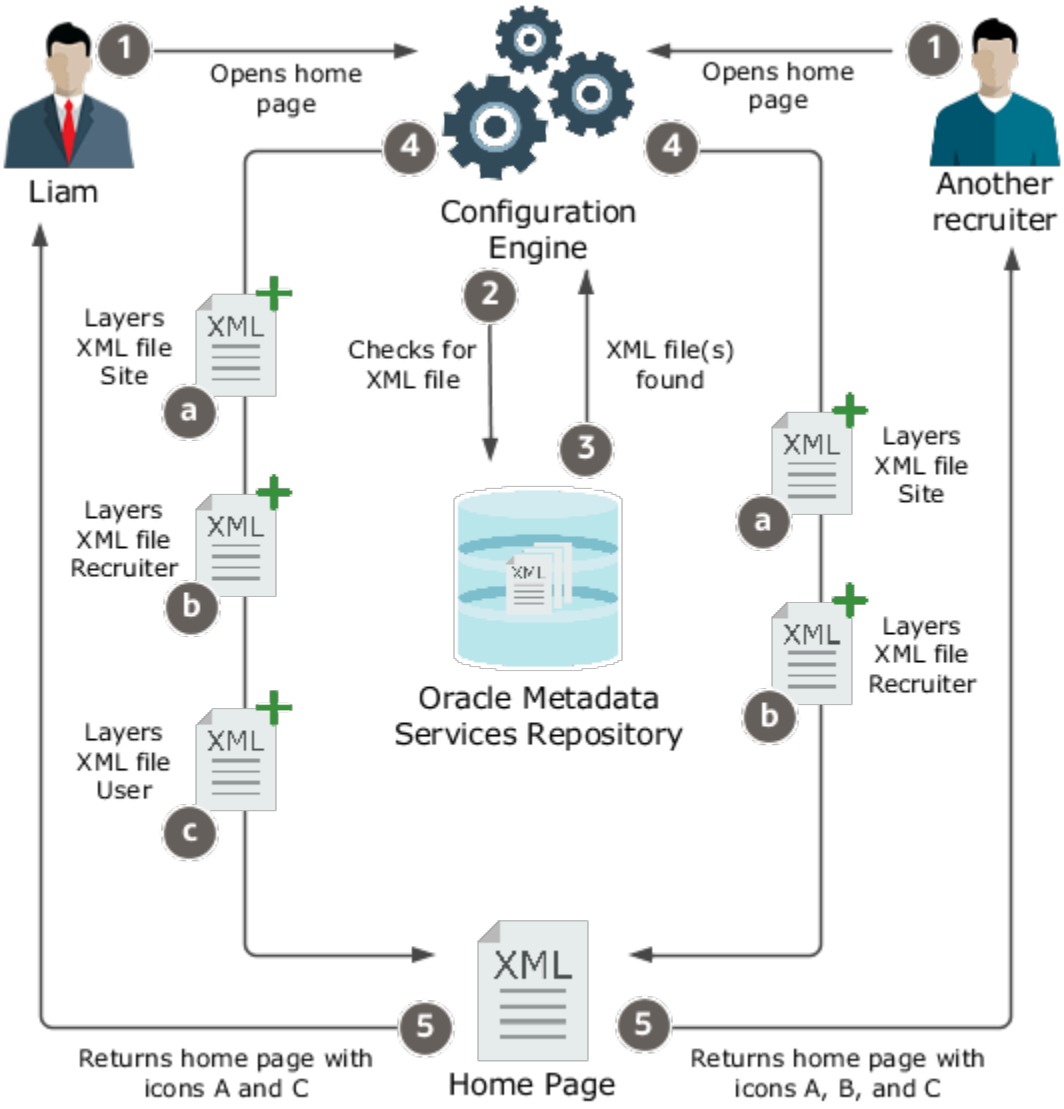
- The **Site** layer adds icons A and B to the home page for everyone.
- The **Job Role** layer adds icon C.
- The **User** layer removes B from the home page because Liam chose to hide it.

When a different recruiter opens the home page, only the XML files for the Site and Job Role layers are applied. This user sees icons A, B, and C.

Here are two diagrams that show you how this personalization flow works. In the first image, you, the administrator, add icons A and B to the home page at the Site layer and icon C at the Job Role layer for recruiters. The image also shows Liam hiding B.



The next image shows what happens when Liam and another recruiter open the home page.



Related Topics

- Overview of Context Layers
- Autoprovisioning
- How do I provision roles to users?
- How can I show or hide work area icons on my home page?
- Personalize Infolets

Example of Including Higher Context Layers While You Configure

When you select a context layer, you can include higher layers to see the application changes from those layers while you configure in your selected layer. Here's an example that explains what happens using the Site, Country, and Organization layers.

What You See While Making Application Changes

Suppose this is what you do when you select context layers:

- Select the **Organization** layer for configuration and select **VMI Manufacturing** as the value for that layer
- Include the **Country** layer and select **France** as the value

Note: The Site layer is automatically included because it applies to everyone.

While you modify pages in Page Composer and make changes for VMI Manufacturing, you also see changes that apply specifically to VMI Manufacturing in France, based on these factors:

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

Let's say you have a field that's hidden in the Site layer, but displayed in the Country layer for France. While modifying pages, you can't see the hidden field because Country is higher than Site.

What Your Application Changes Apply to

No matter what you see while making application changes, your changes apply only to the selected layer, that is, Organization. You now choose to hide the field, and that change applies only to the Organization layer for VMI Manufacturing.

Here's what users see after you make your change:

- VMI Manufacturing in France still can't see the field because Country is higher than Organization.
- Users with other job roles in France can see the field because the Country layer isn't within the scope of the Organization layer, in this case, VMI Manufacturing.
- VMI Manufacturing in any other country can see the field because Organization is lower than Country.

Related Topics

- Overview of Context Layers
- Examples of How the Appropriate Configurations and Personalizations are Applied

2 Sandboxes

Overview of Sandboxes

You use sandboxes to make application changes and test them without impacting other users in the environment. Wherever possible, make changes to the application in a sandbox rather than making direct changes in the mainline environment.

Sandboxes isolate untested configuration changes from the mainline environment. So you can test your changes in the sandbox and then publish it to make your changes available in the mainline metadata or other sandboxes after they're refreshed. After you publish, end users can see your changes after they sign out and sign back in.

Why You Need Sandboxes

Today's business landscape is quite dynamic. Companies are expected to respond quickly to address both customer and market needs. So multiple teams need to make application changes at the same time while sharing the same data model and configuration starting point. But you may get conflicts between teams working that way. To avoid such conflicts, sandboxes come in handy.

Here are a few things you can do using sandboxes:

- Select the configuration tools to enable for your sandboxes while creating them. Since you enable all configuration tools in the same way using the Sandboxes UI, you get a consistent sandbox experience across tools.
- Restrict access to various sandbox activities for users. For example, you can specify these access rights for your sandboxes:
    - Full access
    - Edit and preview access
    - View only access
- View just your application changes without having other context layers hide your content.
- Test your changes in a preview mode that shows you exactly how your application changes would appear in a published sandbox.
- Refresh and merge sandboxes with latest changes in mainline metadata from other published sandboxes. After merging all changes, you can publish your sandbox.
- If you register your target environment in your source environment, you can do these additional migration tasks using the Migration UI:
    - Migrate your changes from the test environment to the target environment without manually downloading and uploading the configuration set file.
    - Move only new changes from the source environment to the target environment.

Sandbox Usage

You typically use sandboxes for either of these purposes:

- Test-Only: You can make application changes using test-only sandboxes, which you don't want to publish to the mainline code.
- Publish: Once satisfied with the application changes made in the test-only sandbox, you can replicate these changes in a sandbox that you want to publish. And then publish your changes to the mainline code. This sandbox type is also known as the integration sandbox, because teams working in parallel use this sandbox as the final staging point before publication to the mainline code.

Note: Before each patch or upgrade, publish or delete your sandboxes. If you haven't yet completed your work, restart with a new sandbox.

Related Topics

- Create and Activate Sandboxes
- Migrate Your Configurations

Create and Activate Sandboxes

To make changes to the application, you must first store the changes in an active sandbox. You can either create a sandbox or select an existing one, and make it active.

Watch video

You must activate the configuration tools you want to use in your sandbox. If you plan to use Page Composer in your sandbox and edit pages at a layer other than Site, you need to create a sandbox just for that layer, and activate only Page Composer in it.

Create and Activate Sandboxes

Follow these steps to create and activate sandboxes for most configuration tools. For flexfields, use the **Manage Descriptive Flexfields** task or the **Manage Extensible Flexfields** task instead.

1. Click **Navigator** > **Configuration** > **Sandboxes**.
2. On the Sandboxes page, click **Create Sandbox**.
3. Enter a name and description for your sandbox.
4. In the **Publishable** field, select **Yes** or **No**. If you set this option as **No**, you can just use your sandbox for testing purposes, but can never publish it.
5. In the All Tools section, select the tools you want to activate for this sandbox. The context layers for all selected tools are set as **Site** by default. So the changes you make using these tools affect all users.
6. If you select Page Composer, you can click the **Edit Sandbox Context** icon and change the context layer from **Site** to another layer, for example **Internal**. You can find the **Edit Sandbox Context** icon in the Support Context column. Make sure you select a context that's supported by the page you want to edit. Otherwise, you won't be able to edit the page.

> **Note:** If you want to use other tools along with Page Composer in your sandbox, don't change the context layer for Page Composer, even though you can. That's because all tools except Page Composer support only a single context layer, Site. So if you change the context layer for Page Composer from Site to any other layer, all other tools that you might have selected earlier will be deselected.

7. Click **Create** to just create the sandbox, or **Create and Enter** to enter or activate the sandbox after creating it.

Here are a few things to know about activating tools in your sandbox.

- If you try to use a configuration tool in a sandbox without activating the tool in it, you get a message prompting you to activate the tool. You can add more tools to your sandbox later also.
- To create and manage saved searches and make UI adjustments (for example, change a table's column width) just for yourself, you must leave your sandbox before making these changes. But if you want to make these changes for others too, then make the changes with Page Composer open, in which case you also must be in a sandbox.

Activate Existing Sandboxes

Follow these steps to activate a sandbox.

1. Click **Navigator** > **Configuration** > **Sandboxes**.
2. From the list of sandboxes, if available, find the one you want to activate, and click the **Enter Sandbox** icon for that sandbox. Your sandbox is activated, and you can see its name on the sandbox bar before the global header. You can use the options available on the sandbox bar to quickly do some activities, such as view sandbox details, publish the sandbox, or leave the sandbox.

Add Tools to Existing Sandboxes

Even though you already selected tools when you created your sandbox, you can still go back and add more tools later.

But if your sandbox has only Page Composer and a layer that's not Site, you can't add any other tool to that sandbox.

1. Click **Navigator** > **Configuration** > **Sandboxes**.
2. Click the name of the sandbox you want to add tools to.
3. In the Active Tools section, click the **Add Tools** icon.

> **Note:** You might not be able to add tools to your sandbox because of various reasons, for example, you just had a release update or upgrade.

4. In the All Tools dialog box, select the tools you want to activate for this sandbox, and click **OK**.

   The context layers for all selected tools are set as **Site** by default. So the changes you make using these tools affect all users.

Related Topics

- Best Practices for Using Page Composer in Sandboxes
- Why can't I add tools to my sandbox?
- Create and Activate Sandboxes

Delete Sandboxes

Before you delete any sandbox, make sure that's not the active sandbox you're currently in. If you're in a sandbox that you want to delete, leave that sandbox first.

On the sandbox bar before the global header, click the sandbox name and select **Leave Sandbox**.

Here's how you delete a sandbox:

1. Click **Navigator** > **Configuration** > **Sandboxes**.
2. Click the name of the sandbox you want to delete.
3. On the Sandbox Detail page, select **Delete** on the **Actions** menu.

Related Topics

- Why is the Delete button disabled for my sandbox?

Best Practices for Using Sandboxes

Several users might be working on your implementation in parallel in different sandboxes. So, follow these best practices to avoid conflicts while using sandboxes.

- To manage sandboxes easily and improve the system performance, reduce the number of open publishable sandboxes at any time.
- Reduce the number of active users using a publishable sandbox.
- Make sure a single user doesn’t open multiple browsers and work in the same sandbox.
- It's recommended that multiple users don't work in the same sandbox at the same time.
- For best performance, make sure you don’t use more than ten sandboxes in parallel.
- Don’t make too many configurations in one sandbox. Instead, make small configurations in separate sandboxes and publish them.
- Sign out and sign back in every time you create, activate, publish, or leave a sandbox. Doing this clears any user-level caching to ensure that you’re working with the latest configurations.
- Publish your sandboxes before patches are applied, or any release updates or upgrades happen. Otherwise, your unpublished sandboxes that existed before the patch was applied, will be marked as unpublishable to avoid invalid or inconsistent content being published.
- Don’t publish your application changes directly in the target environment. To make sure changes aren't published in the target environment, set the **Control Publish Sandbox Action in Production Environment** profile option (FND_ALLOW_PUBLISH_SANDBOX) to **No** in the target environment.
- It’s recommended that you don’t use sandboxes in the production environment.
- Plan to publish your sandboxes and custom subject areas during periods of low system usage or low user activity.

Related Topics

- Best Practices for Using Page Composer in Sandboxes
- Delete Sandboxes
- Publish Sandboxes
- About Application Composer

Best Practices for Using Page Composer in Sandboxes

Here are a few things to keep in mind if you're using Page Composer in your sandbox.

- While creating your sandbox, you can change the context layer for Page Composer from the default layer, Site to another layer, or from another layer back to Site. But for all other tools, you can use only the Site layer. After you create your sandbox, you can't change the context layer for Page Composer.
- If you select a context layer that’s not Site, make sure the page you want to edit supports this context layer. Otherwise, you won’t be able to edit the page. For example, only Customer Relationship Management pages support the Job Role layer. So, if you create a sandbox for a specific job role, you won’t be able to edit Human Capital Management pages in this sandbox.
- If you create configurations for a specific job role using Page Composer, then you must assign yourself that same job role to view and test configurations in the sandbox. For example, if you create your page layout for the Sales Manager job role, you must have the Sales Manager job role to view and test the page layout.
- While creating your sandbox, if you have selected Page Composer and a layer that's not Site, don't add other tools to your sandbox. If you do, you will be prompted to confirm adding other tools and changing the layer for Page Composer to Site.
- After you create a sandbox that has only Page Composer and a layer that's not Site, you can't go back and add any other tool to your sandbox.
- While creating your sandbox, suppose you have selected multiple tools including Page Composer. And then you try to change Page Composer's context layer to a layer that's not Site. That change won't happen unless you confirm that all other tools should be removed from the sandbox.

Related Topics

- Create and Activate Sandboxes
- How the Refresh and Merge Processes Work in Sandboxes

How to Resolve Conflicts in Sandboxes

When you're in a sandbox, if other users publish their sandboxes, you get refresh notifications on the sandbox bar before the global header. At this time, it's a good practice to refresh your sandbox.

When you refresh, all changes published in the mainline environment are brought into your sandbox. You get sandbox merge conflicts in the merge log when different users change a specific file using different sandboxes. If the changes are made to different files, they're automatically merged, and aren't even reported in the merge log.

Note: In Application Composer, an object gets automatically locked when you create it or modify it in a sandbox. So in such cases, an object can only be edited in any one sandbox at a time. If the sandbox is published or deleted, the lock is removed.

You must resolve all conflicts flagged in the merge log so that you can publish your sandbox. To review the merge log, on the Sandbox Detail: <Sandbox Name> page, click the Merge Log tab. The log displays details about the sandbox merge statuses. Let's understand what these statuses mean and how we can resolve the sandbox merge conflicts based on their statuses.

Merge Status	Icon	What It Means	How I Can Resolve Merge Conflicts
Automatically Merged Content	Auto Merge	Different users changed different attributes of the same file using different sandboxes.	These changes are merged automatically.
Resolvable Conflicts	Resolvable	Different users changed the same attribute of the same file using different sandboxes.	These changes can be merged to the mainline environment. On merging, the changes in the mainline environment overwrites your sandbox changes. So review the merge actions and accept or reject them. • If you accept, you can later redo any changes that were overwritten and then publish your sandbox. • If you reject, your sandbox remains untouched, but you still need to accept a merge before you publish your sandbox.
Unresolvable Conflicts	Unresolvable	Different users changed files in different sandboxes, but the merge conflicts can't be automatically resolved.	Do any of these tasks: • Undo your sandbox changes. • In your sandbox, make the same change, which the other user made in the published sandbox, and try to resolve the conflict. • Create another sandbox and make your changes in that one.

Related Topics

- Publish Sandboxes
- How the Refresh and Merge Processes Work in Sandboxes

How the Refresh and Merge Processes Work in Sandboxes

Sandbox changes are refreshed and merged when two different users make changes to the same file using two different sandboxes. Let's look at an example.

Suppose your manager creates a sandbox named Sandbox1 and you create another sandbox named Sandbox2. Your manager then makes a change to a file using Sandbox1 and publishes it to the mainline environment. Now if you make changes to the same file and refresh Sandbox2, all changes published in the mainline environment are merged into Sandbox2. What if both you and your manager entered different values to the same attribute of the file? In that case, the value that your manager entered using Sandbox1 persists because Sandbox1 is published to the mainline metadata. To bring back the changes that you made in Sandbox2, you need to reenter the sandbox and make the changes again.

Application Changes That Can or Can't be Merged

Let's again take the same example of Sandbox1 and Sandbox2. Here's a list of application changes made in Sandbox2 that can merge with the mainline environment, when the same file is changed in both sandboxes.

Application Changes	Tools Used to Make the Changes
Changes in business objects and their related fields	Application Composer and Configure Business Objects
Changes in data security	Data Security
Changes in lookups	Lookups
UI text changes	User Interface Text
Changes to messages, such as warning messages and information messages	Messages
Changes to the appearance of the application	Appearance

Note: Flexfields and setup tasks, apart from the ones listed in the table, aren't supported in sandboxes. So changes in these artifacts aren't merged when Sandbox2 is refreshed.

And here's a list of changes made in Sandbox2 that can't merge with the mainline environment when the same file is changed in both sandboxes. To bring back those changes, you can create another sandbox, make the changes in it, and publish that sandbox.

Application Changes	Tools Used to Make the Changes
Configure the Navigator and the icons on the home page for navigation	Structure
Changes in pricing configuration	Manage Service Mappings
Changes to the page content	Page Composer
Changes to global page template	Page Template Composer
New pages created	Page Integration

Related Topics

- Considerations for Deploying a Descriptive Flexfield to a Sandbox
- How to Resolve Conflicts in Sandboxes

Options to Open Configuration Tools from Sandboxes UI

In the sandboxes UI, after activating configuration tools, you can also use shortcuts to quickly open some of these tools and make your changes.

Note: You can't open all tools from there. In which case, you can get to those tools using regular navigation.

Use Shortcuts in Sandboxes

After you activate a sandbox, all tools activated in it are listed on the sandbox bar and the Sandbox Detail page. You can open these tools from either of these locations:

- The Tools menu on the sandbox bar before the global header
- The Active Tools section of the Sandbox Detail page

In this list, you may notice that some tools are available, while others, for example, Lookups, Messages, Data Security, Global Page Template, and Manage Service Mappings are grayed out. You can click the available tools to open them directly from the Sandboxes UI.

Use Regular Navigation

This table lists the regular navigation options to open all tools that you can activate in your sandboxes. It also indicates whether you can open the tools from the Sandboxes UI.

Tool Name	Can Open Using the Sandboxes UI?	Regular Navigation
Application Composer	Yes	Click **Navigator** > **Configuration** > **Application Composer**.
Configure Business Objects	Yes	Click **Navigator** > **Configuration** > **Business Objects**.
Appearance	Yes	Click **Navigator** > **Configuration** > **Appearance**.
Manage Service Mappings	No	Click **Navigator** > **Order Management** > **Pricing Administration**, and then on the Tasks panel tab, click **Manage Service Mappings**.
Page Integration	Yes	Click **Navigator** > **Configuration** > **Page Integration**.
Structure	Yes	Click **Navigator** > **Configuration** > **Structure**.
Lookups	No	In the Setup and Maintenance work area, use the lookup tasks, such as: • Manage Standard Lookups • Manage Common Lookups • Manage Set Enabled Lookups
User Interface Text	Yes	Click **Navigator** > **Configuration** > **User Interface Text**.
Messages	No	In the Setup and Maintenance work area, use the messages tasks, such as: • Manage Messages • Manage Messages for General Ledger
Data Security	No	Click **Navigator** > **Tools** > **Security Console**, and then click the Administrator tab, and click **Manage Database Resources**.
Page Composer	Yes	Click your user image or name in the global header and select **Edit Pages**.
Global Page Template	No	Click your user image or name in the global header and select **Edit Global Page Template**.

Related Topics

- How to Resolve Conflicts in Sandboxes
- Object Locking in Application Composer
- How the Refresh and Merge Processes Work in Sandboxes
- How do I update existing setup data?

Object Locking in Application Composer

The automatic object locking ability in Application Composer avoids the risk of conflicts between multiple users working on objects in parallel and prevents any sandbox merge conflicts that may arise when different users change a specific object using different sandboxes.

Application Composer automatically places a lock on a business object when you create or modify it in a sandbox. The locked object displays a lock icon next to its object name in Application Composer's object navigation tree. A gold lock indicates that the object is locked in the current sandbox. A gray lock indicates that the object is locked in a different sandbox. Hover over a locked object's name in the navigation tree to display the name of the sandbox that holds the lock.

Application Composer displays and enforces object locks across all sandboxes. You can only edit a locked object in the sandbox that holds the lock. For example if the Service Request object is modified in Sandbox A, only Sandbox A holds the lock and anyone using Sandbox A can modify Service Request. Any other sandbox displays a gray lock on Service Request in Application Composer, and prevents users from modifying it.

The lock status of objects changes in the following cases:

- When a sandbox that had a lock is published or deleted.
- When a new object lock is established.
- When a sandbox becomes not publishable, which happens after a patch, release update, or a migration.

Collapse and then expand Application Composer's object tree to update the lock status of all objects. Any action that causes the object tree to be refreshed will update the lock status, including when you exit and reenter a sandbox, or refresh a sandbox. If an object is locked and that lock isn't yet visible in the current sandbox, a new lock isn't allowed to be placed on that object and an error message appears while saving the changes.

> **Note:** If the sandbox you're working in was created for testing purposes only and was set as not publishable or becomes not publishable, then the business object you create or modify won't be automatically locked. This way, multiple teams can work on the same business object in different test sandboxes.

For more information on how object locking works in sandboxes, see the FAQs on Object Locking.

Best Practices for Preventing Object Locking

Here are some tips on how you can prevent or work around object locking:

- Plan your object model and user interface configurations, and divide the work to prevent two users requiring the same object or objects locked.
- Name sandboxes clearly with appropriate names or initials, so when a lock is in place, it's easy to identify who to contact to release the lock from the sandbox holding the lock.

- Perform configurations in smaller increments. Test and publish more frequently to prevent holding locks.
- Delete test sandboxes created to try out new configurations after they're no longer needed to prevent unnecessary locking.
- If you're using a sandbox just to test changes and don't intend to publish them, then set the **Publishable** field to **No** to prevent locking.

Create Backups of Sandbox Configurations

Before publishing your sandbox, you can download and save your configurations as a .jar file in your local file system to create a backup.

1. Click **Navigator** > **Configuration** > **Sandboxes**.
2. Click the name of the sandbox you want to create a backup for.
3. On the Sandbox Detail page, click the **Download All** button on the Sandbox Changes tab.
4. Save the .jar file.

Publish Sandboxes

After you're done making changes to the application, publish the sandbox to make your changes available to all users. You must have the Administer Sandbox (FND_ADMINISTER_SANDBOX_PRIV) privilege to publish sandboxes.

> **Note:** Remember, you can't make further changes in the sandbox once you publish it.

Before you start, do these tasks:

- Test or validate your changes in the sandbox in preview mode before actually publishing it. If you made changes using Page Composer, don't forget to close it before testing. To preview your changes, on the sandbox bar before the global header, click **Sandbox Mode**, and select **Preview as if Published (Context: All)**.

  > **Note:** You can see the sandbox bar only when you're in an active sandbox.

- Resolve all conflicts flagged in the merge log of your sandbox.

To publish a sandbox:

1. Click **Navigator** > **Configuration** > **Sandboxes**.
2. On the Sandboxes page, click the name of the sandbox you want to publish.
3. Click **Publish**.

   > **Note:** You might not be able to publish your sandbox because of various reasons. For example, you haven't yet made any changes in your sandbox, or the **Control Publish Sandbox Action in Production Environment** profile option (FND_ALLOW_PUBLISH_SANDBOX) is set to **No**.

4. Click **Continue** to Publish. The sandbox is published to the mainline metadata.
5. Click **Done**.

Related Topics

- Create and Activate Sandboxes

FAQs for Sandboxes

Who can preview, edit, and publish sandboxes?

Users with the Administer Sandbox (FND_ADMINISTER_SANDBOX_PRIV) privilege can preview, edit, and publish sandboxes. 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.

> **Note:** With other sandbox privileges, users can only do some tasks. Users having the Manage Sandbox (FND_MANAGE_SANDBOX_PRIV) privilege can edit but not publish sandboxes. While those having the View Sandbox (FND_VIEW_SANDBOX_PRIV) privilege can view sandboxes in read-only mode.

Why can't I add tools to my sandbox?

It might be because of one or more of these reasons.

- Your mainline metadata has changes from other published sandboxes. You must refresh your sandbox and then add tools to it.
- You just had a release update or upgrade, and you can no longer add tools to any of the sandboxes that were available before the update or upgrade. In the future, add tools to your sandboxes before the release updates or upgrades happen.
- A migration set was just applied to your environment, and now you can't add any tools to sandboxes that already have Application Composer. In the future, add tools to sandboxes before applying any migration sets.

Related Topics

- Migrate Your Configurations

Why can’t I open the configuration tool that I added to my sandbox, using the Sandboxes UI?

In the Sandbox UI, shortcut links to some configuration tools, such as Lookups, Messages, Data Security, Global Page Template, and Manage Service Mappings are grayed out. This is because these tools aren't directly accessible from the Sandbox UI. You can open these tools using the regular navigation options.

Related Topics

- Options to Open Configuration Tools from Sandboxes UI

Why is the Delete button disabled for my sandbox?

That's because you're currently in the sandbox that you want to delete. You must first leave that sandbox. On the sandbox bar before the global header, click the sandbox name and select **Leave Sandbox**.

Why can't I publish my sandbox?

It might be because of one or more of these reasons.

- You haven't yet made any changes in your sandbox.
- Another sandbox is currently being published in this environment. Try publishing your sandbox again later.
- Your mainline metadata has changes from other published sandboxes. You must refresh your sandbox and then publish it.
- You don't have the Administer Sandbox (FND_ADMINISTER_SANDBOX_PRIV) privilege, which is required for publishing sandboxes. To get this privilege, contact your security administrator.
- The **Control Publish Sandbox Action in Production Environment** profile option (FND_ALLOW_PUBLISH_SANDBOX) is set to **No**. To enable publishing, set this profile option to **Yes**. In the Setup and Maintenance work area, use the **Manage Applications Core Administrator Profile Values** task in the Application Extensions functional area.
- The application is in maintenance mode. Try publishing your sandbox again later.
- While creating this sandbox, you have set the **Publishable** field as **No**, which means this sandbox can never be published.
- Your sandbox is in an inactive state, which happens if a publish action on it failed. You can never publish such inactive sandboxes.
- Your application was patched, updated, or upgraded, and all sandboxes that were available before the patch, update, or upgrade, can no longer be published. In the future, publish your sandboxes before patches are applied, or any release updates or upgrades happen.
- A migration set was just applied into the environment, and now all sandboxes that have Application Composer activated in them, can no longer be published. In the future, publish or delete sandboxes before applying any migration sets.

Related Topics

- Publish Sandboxes
- Migrate Your Configurations
- Set Profile Option Values

How can I see details for published sandboxes?

On the Sandboxes page, (**Navigator** > **Configuration** > **Sandboxes**), click the Published Sandboxes tab to see the sandboxes that are already published and get details such as name and publishing date.

Why doesn't the page I am editing with Page Composer work with the sandbox context?

Let's see why you might get errors about the page not supporting the context layer of the active sandbox, and how you can resolve them.

- Maybe the context layer of your sandbox isn't a layer that your page supports. Context layers vary based on the category you select for the sandbox. For example, the Customer Relationship Management category supports certain layers that apply to certain pages, and the Human Capital Management category supports other layers and pages.

  To check the context layer of your sandbox, on the sandbox bar before the global header, click the sandbox name and select **Sandbox Detail**. The context is displayed on the Sandbox Details page under the Basic Information section.

  You can activate a sandbox with the appropriate layer and try editing the page again.
- You might not have a role that gives you access to what the sandbox layer covers. For more information, contact your security administrator.

Related Topics

- Create and Activate Sandboxes

Why are the lookup, message, and data security changes appearing in my sandbox even though I haven't yet refreshed my sandbox?

Lookups, messages, and data security records are values stored in tables. These records aren't copied to your sandbox until you make changes in these artifacts in your sandbox.

But what if another user changes these artifacts in a different sandbox and publishes it before you make changes in your sandbox? In that case, the other user's changes show up in your sandbox as soon as you make changes to these artifacts, even before you refresh your sandbox.

Why can't I see application changes made in previously published sandboxes in my current sandbox?

That's because a sandbox is set to only one context layer, and layers are applied in a specific order.

- Applied 4th: User
- Applied 3rd: External or Internal
- Applied 2nd: Job Role
- Applied 1st: Site

When you're in a sandbox set at a specific layer, you can see your own changes plus changes from previously published sandboxes that were set at your context layer and below it. You won't see changes made in previously published sandboxes if those sandboxes were set at a layer that's higher in the hierarchy. So while working in your sandbox, what application changes you see depend on these factors:

- The layer at which changes were made in the previously published sandboxes
- The layer activated in your sandbox

However, you can switch to preview mode to test your changes alongside changes made in previously published sandboxes at all layers. Preview mode puts your sandbox into a read-only mode, where you can see all changes, including those made at a layer higher in the hierarchy than your own. After testing and validating your changes in preview mode, you can publish your sandbox.

Let's look at a few examples to understand how context layers impact what changes you can see in your sandbox:

- When working with Application Composer in a sandbox, your sandbox is always automatically set to the Site layer. In this sandbox, you can see your own changes plus any previously published configurations made at the Site layer. But you won't see changes that were made at higher layers, such as Job Role, External or Internal, or User.
- Let's say you're working with Page Composer in a sandbox at the Job Role layer. You can see your own changes plus changes made in previously published sandboxes at the Site and Job Role layers. But, perhaps in that sandbox at the Job Role layer, you hid a page component and published your changes. Now, if you create a sandbox for Page Composer at the Site layer, you will still see that page component (it's not hidden!) because Site is lower than Job Role in the hierarchy. The changes made at the Job Role layer won't reflect in your sandbox at the Site layer, unless you switch to preview mode.

How can I roll back changes made in a previously published sandbox?

You can create a new sandbox, roll back the unwanted configurations, and publish that sandbox. But if you have made significant configurations that you can't undo manually, ask your help desk to log a service request for a Production-to-Test data refresh.

The data refresh will overwrite all configurations in your environment with the production data.

What happens to my sandboxes after a release update or upgrade?

It depends on whether you published your sandbox before the release update was applied to the environment, or the environment was upgraded.

- If yes, your configurations will remain untouched.
- If not, you will lose the configurations you made in that sandbox. Also, you can no longer publish it. So, before the release update or upgrade happens, you should always publish or delete your sandboxes. If you're not done yet with your work in the sandbox, create a new sandbox in the updated or upgrade environment and make your changes there.

How can I deploy flexfields to a sandbox?

For value sets, key flexfields, and extensible flexfields, you must deploy changes directly to the mainline environment, not to a sandbox. But for descriptive flexfields, you must deploy your changes to a sandbox and test them before publishing your sandbox.

To deploy your descriptive flexfield to a sandbox:

1. Create or activate a sandbox, and select **Flexfields** from the list of tools.
2. In the Setup and Maintenance work area, go to the **Manage Descriptive Flexfields** task
3. Open the flexfield you want to edit, and make your changes.
4. On the **Actions** menu, select **Deploy Flexfield to Sandbox**.

Related Topics

- Considerations for Managing Flexfields
- How do I update existing setup data?

How does object locking impact object workflows?

When you create an object workflow for an object that's not currently locked, a lock is created for that object.

You can create a new object workflow for a locked object in the sandbox that holds the lock. However, an object locked in a different sandbox is disabled for selection in Application Composer's list of available objects while creating an object workflow. Hover over the name of the disabled object in the list to identify which sandbox has a lock on that object.

When are object locks released?

An object lock exists until the sandbox that holds the lock is published or deleted. A lock is not removed when the change that caused the lock is deleted or reverted.

For example, adding a new page layout creates a lock on an object if the object is not already locked, but deleting the layout will not release the lock.

Does importing and exporting configurations impact object locks?

Importing configurations can impact sandboxes. You must delete existing sandboxes before importing new configurations to release all locks and to prevent users from thinking that they can refresh and publish after an import.

How does object locking affect parent and child objects?

An object is locked only when it is directly modified. Modifying a child object will not result in a lock on the parent object.

For example, consider a Custom Context subtab (a subtab that displays content of a different object), which is added to the Premium Accounts object. The Custom Context subtab displays the Accounts object, which is the parent object of Premium Accounts. The object lock is placed on Premium Accounts and not Accounts, because Accounts (the parent object) was not modified.

How does object locking work for dynamic choice lists?

When you create a custom dynamic choice list, a lock is placed on the object that the custom dynamic choice list is created for. The data source object (the object that the dynamic choice list pulls data from) isn't locked.

Can I manually set or remove object locks?

No, you can't manually set or remove object locks. Locks are automatically set when you create or modify objects in a sandbox, and automatically released when you publish or delete the sandbox that holds the locks.

3 Migration and Troubleshooting

Tools for Moving and Troubleshooting Configurations

Configurations are stored in XML files. You can use these XML files to do these tasks.

- Move configurations and extensions to another environment, such as the production environment.
- Diagnose issues noticed in the test environment.
- Send files to your help desk for further diagnosing.

Here's the list of tasks and the corresponding tools to export and move configurations and extensions.

Tasks	Tools to Use
Move all configurations to another application environment.	Migration.
Move only descriptive flexfield configurations to another application environment.	Setup and Maintenance work area. Using this work area, you can move configurations only for a specified module. To move configurations for all modules, use Migration.
Move only extensible flexfield configurations to another application environment.	Setup and Maintenance work area. Using this work area, you can move configurations only for a specified module. To move configurations for all modules, use Migration.
Move only value set configurations to another application environment.	Setup and Maintenance work area. Using this work area, you can move configurations only for a specified module. To move configurations for all modules, use Migration.
Move only lookups to another application environment.	Setup and Maintenance work area. Using this work area, you can move application standard lookups, application common lookups, or both.
Move only data security policies to another application environment.	Setup and Maintenance work area. Using this work area, you can't move Oracle Fusion Cloud Human Capital Management roles.
Export configurations to a file to help diagnose an issue.	Manage Configurations 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.

Related Topics

- Contents of the Migration Set
- Overview of Context Layers
- Guidelines for Moving Related Common Reference Objects
- Export Setup Data Using Offering or Functional Area

Migration

Overview of Migration

You can move your configurations from the test environment, which is the source, to the production environment, which is the target. You can also update your target environment with the extensions made using Visual Builder Studio.

You use the Migration page to create a migration set and export it from the source environment, and then import it into the target environment. A migration set is a JAR file that contains all your configurations across all product families, such as Oracle Metadata Services repository, JEDI, CRM, and BI. Changes you export from the source environment can't be merged with any changes you manually make in the target environment. As such, you must never configure the target environment directly.

When you migrate your configurations, you have the option to migrate only new changes from the source environment to the target environment. For migrating only new changes, you must synchronize both environments by registering the target environment in the source environment. When you register the target environment, your migration set is automatically set to the target environment for import. Otherwise, you have to manually download the migration set from the source environment, and manually upload it into the target.

Your migration set is imported into a sandbox instance before it's applied to the target environment. You can preview your configurations in this sandbox instance before applying them to the mainline.

> **Note:** The topic focuses on moving your configurations from the source environment to the target environment using the Migration page in Oracle Fusion Cloud Applications. For information about moving your extensions made using Visual Builder Studio to the target environment, see *Set Up the Project to Deploy to Production* and *Migrate Oracle Cloud Applications Configurations*.

Related Topics

- Set Up the Project to Deploy to Production
- Migrate Oracle Cloud Applications Configurations

Contents of the Migration Set

You use the Migration page to create a set of configurations and extensions made to an environment, download the set, and then upload it into another environment. This migration set includes configurations across all product families.

What It Includes

The migration set includes, but isn't limited to these changes:

- Application changes done using Application Composer:
    - Object UI extensions
    - Object server scripts
    - Saved searches
    - Workflows
    - Global functions
    - Object model extensions
    - Relationships
    - Role security privileges to access objects
    - E-mail templates
    - Custom subject areas
    - Smart actions

    > **Note:** Saved searches created only at the site layer are migrated, not those created at the user or job role layers. So, you need to manually create the saved searches at the user and job role layers in the target environment.

- Changes made in Oracle Fusion Cloud ERP using Application Composer
- Changes in functional security settings made in Application Composer, including functional privileges that control access to custom objects. However, not all security changes are migrated.

    > **Note:** If you migrate any functional security associated with roles in the source to a target instance, and the corresponding role doesn't exist in the target instance, an error occurs on import. To avoid these errors, you can selectively create these roles in the target environment when you import your migration set.

- Changes made to application artifacts using these tools:
    - Page Composer
    - Appearance
    - Structure
    - User Interface Text
    - Page Integration

- Changes in these artifacts of the Applications Core Setup application:
  - Messages
  - Lookups
  - All data security policies for standard and custom objects, including standard and custom policies.

    Note: Data security policies that are created using these tasks in the Setup and Maintenance work area aren't migrated:
    - Manage Data Role and Security Profiles
    - Candidate Security Profiles
    - Job Requisition Security Profiles
    - Manage Country Security Profile
    - Manage Document Type Security Profile
    - Manage Legislative Data Group Security Profile
    - Manage Organization Security Profile
    - Manage Person Security Profile
    - Manage Position Security Profile
    - Manage Transaction Security Profile
    - Payroll Flow Security Profile
    - Payroll Security Profile
  - Descriptive, extensible, and key flexfields, and value sets that have been deployed
  - Deep links
- All published configurations within sandboxes
- Changes in Reports and Analytics, such as regeneration of SOAP services, including user-defined attributes

What It Might Include

Depending on what modules you select during export, your migration set might include these changes:

- Changes in CRM email templates created in Application Composer
- Changes in flexfields and value sets
- Changes in the Enterprise Scheduler Service (ESS) module:
  - Job definitions
  - Job sets
  - Job schedules
  - Incompatibilities
  - Work shifts
  - Work assignments
  - Work assignment schedules
  - Job request parameters
- Changes in Service Oriented Architecture (SOA) artifacts, such as configurations done using SOA Composer

- Changes done using Oracle Transactional Business Intelligence, including but not limited to these features:
    - Oracle Analytics Answers
    - Oracle Analytics Delivers
    - Business Intelligence Composer
    - Dashboard Builder
    - Oracle Analytics Publisher

> **Note:** You can move configurations done using business intelligence tools only if the **Business Intelligence in Configuration Set Migration Disabled** profile option is set to **No**.

What It Doesn't Include

Your migration set doesn't include these items:

- The following Application Composer changes:
    - Object artifacts that were generated to make extensions available for importing and exporting (generated either automatically when you publish a sandbox or manually from the Import and Export page in Application Composer)
    - User names and passwords for secured SOAP web service connections
    - The enabled attachment feature for custom objects
- Personalizations
- Extensions created using Visual Builder Studio. You must deploy extensions to the production environment instead of migrating them.
- Changes in activity types and subtypes

> **Note:** Activity types and subtypes are performance metrics that managers use to track the performance of their team members. For example, email is an activity type that sales managers can use to track the performance of their salespeople. The subtypes of this activity type can be outbound and inbound. Managers can use activity subtypes to track more details about an activity type.

- Deletions, for example, the migration set doesn't include the removal of a custom object. After you import a configuration set into the target environment, you must examine the environment for any deletions that you must make manually.
- Profile options
- Roles or role hierarchy changes
- Custom roles that aren't assigned to custom objects in Application composer

- Security Console changes, including these changes made directly in the Security Console:
  - Enterprise roles
  - New duty roles
  - Role hierarchy changes

  You must migrate your custom role and role hierarchy changes from the source environment to the target environment using a .csv file.

Related Topics

- Import and Export Custom Objects
- Export and Import HCM Security Setup
- Overview of Smart Actions

Migrate Your Configurations

Use a migration set to move your configurations from the source environment to the target environment. You can import your configurations into a sandbox in the target environment before you apply them to the mainline.

If you register your target environment in your source environment, you can do these additional migration tasks:

- Migrate your changes from the source environment to the target environment without manually downloading and uploading the migration set file.
- Move only new changes from the source environment to the target environment. This type of migration is called delta migration. However, only sandbox-aware modules support delta migration. All other modules move all changes every time they're migrated, even in delta migrations.

> **Note:** The topic focuses on moving your configurations from the source environment to the target environment using the Migration page in Oracle Fusion Cloud Applications. For information about moving your extensions made using Visual Builder Studio to the target environment, see *Set Up the Project to Deploy to Production* and *Migrate Oracle Cloud Applications Configurations*.

When You Can Do Delta Migration

A delta migration moves only those changes that were made to an environment after the last migration was done. On the other hand, a full migration moves all changes made to an environment.

You can do a delta migration if you meet these two conditions:

- You have registered your target environment in your source environment.
- Both of your environments are synchronized.

Here are some scenarios when you can perform delta migration:

- Let's say your source environment is Alpha, and your target environment is Beta. You create object X in the Alpha environment, and do a migration. Then, you create object Y in the Alpha environment and do another migration. If the second migration is a delta migration, only the object Y is moved. However, if it's a full migration, both the objects X and Y are moved.

- Let's consider a setup of three environments, namely, development (source), test (target), and production (target).

  Between development and test environments:

  - Application change C1 is made in a sandbox in the development environment, and then published and moved using delta migration to the test environment.
  - Application change C2 is made in a sandbox in the development environment, and then published and moved using delta migration to the test environment.

  Between development and production environments:

  - Previously published application changes C1 and C2 in the development environment get moved using delta migration to the production environment.

Related Topics

- What themes can I use in Fusion Applications?
- Set Up the Project to Deploy to Production
- Migrate Oracle Cloud Applications Configurations
- How do I update existing setup data?

Before You Start

Before creating a migration set, consider these points:

- Make sure that the source and target environments are of the same release, with the same standard and one-off patches applied to both environments.
- You should delete or publish any sandboxes in the target environment that have Application Composer enabled before you begin your migration.
- Make sure that all Page Composer configurations made in sandboxes are complete and published.
- Make sure that all configurations and extensions made using the Structure page, the **Manage Standard Lookups** task, and the Security Console are complete.
- If you don't need to migrate flexfield configurations related to business intelligence, set the **Business Intelligence Extender in Configuration Set Migration Disabled** (FND_CSM_BI_EXTENDER_DISABLE) profile option to **Yes**. Setting this profile option to **Yes** makes the migration process faster. The default value for this profile option is **No**. You can find this profile option in the **Manage Administrator Profile Values** task in the Setup and Maintenance work area.

  > **Note:** If you don't enable this profile option in the target environment, flexfield configurations related to business intelligence are included in your migration set. In such cases, the migration process takes longer to complete.

- To move content created using Oracle Transactional Business Intelligence, set the **Business Intelligence in Configuration Set Migration Disabled** profile option to **No** in both the source and target environments. You can find this profile option in the **Manage Administrator Profile Values** task in the Setup and Maintenance work area.
- You must have the following privileges to access the Migration page:
  - Manage Outgoing Configuration Set

    - Manage Incoming Configuration Set

  Contact your security administrator for details.

- You should never make changes in the target or production environment while applying configurations.

  > **Note:** If you make changes to the production environment in emergencies, you must make the same changes to the test environment. Making the changes to the test environment ensures that these changes are included in the next migration. However, in Application Composer, you should not create an object in the production environment because if you do so, you need to ask your help desk to log a service request for a production-to-test data refresh to resolve any migration issues.

- Don't make any changes in the source environment during the export process.
- Delete any temporary files on the server from previous migrations. If there are temporary files on the server, click the **Delete** button next to your previous import and export records.
- If you have made any security changes in the source environment outside of Application Composer, make sure that you manually re-enter these security changes in the target environment before importing your migration set. For example, if you have set up a custom security role in the source environment, you must manually add this role in the target environment.

> **Note:** For recommended best practices, see *Best Practices for Migration*.

Register the Target Environment

1. In your source environment, in the Setup and Maintenance work area, go to the **Manage Configuration Set Migration Target Security Policy** task.
2. Enter the URL of your target environment. Use the full URL, including the host and protocol information.
3. Enter the application administrator role user credentials. Single sign-on isn't supported. So, enter your local user name and password.
4. Click **Save and Close** to apply the changes.

Verify the Target Environment

1. In the source environment, click **Navigator** > **Configuration** > **Migration**.
2. Click the Environment Info infotile.
3. Verify that the page displays the same URL as the target instance.

If you don't see the correct target instance, try registering it again.

Create Migration Set

1. In the source environment, click **Navigator** > **Configuration** > **Migration**.
2. Click **Create Migration Set** from the Outgoing infotile. If you want to migrate all configurations, click **Move All Changes** for a full migration.If you want to migrate only new configurations instead of all changes, click **Move New Changes** for a delta migration.
3. Provide a name for the migration set.
4. Provide the email ID of the person you want to notify after the migration set is created. You can provide multiple email IDs separated by commas. When the migration set is created, email notifications are sent to these specified people.
5. Optionally, type a description of the set.
6. Select the content you want to migrate.

> **Note:** The Industry solution extensions module is for Oracle's internal use only and has no impact on migration.

7. Click **Save and Close**.
8. Click the **Refresh** icon periodically to see the current status of the set creation. You can click the **Log** icon to review the process log, which provides more details about the configurations that are being compressed. If an error or exception occurs during this process, the log gives you information about the configurations that failed to compress. The process runs asynchronously, so you can close the page and return to it later.
9. Eventually, the status changes to Ready for Download, which means that the migration set is complete. You can click **Content Read Me** to download the Readme file listing all the configurations you exported.
10. If you have registered the target environment, you don't need to download your migration set. In such cases, you can directly apply the migration set.
11. If you didn't register the target environment, do these tasks:
    - Click the **Download** icon to download your migration set. Ensure that the downloaded file is a JAR file.
    - After you download the file on your local file system, click **Delete** to remove the temporary files from the server.

Apply Migration Set

After you apply configurations, end users must sign out and sign back in to see the changes. So, apply configurations when fewer people are signed in to the environment.

To apply configurations to the target environment, follow these steps:

1. Open the Migration page in the target environment.
2. Click the Incoming infotile.
3. Click the **Upload Migration Set** link. Then, browse for your migration set file, and click **OK**.

> **Note:** If you have registered the target environment, you don't need to upload the migration set to the target environment. That's because your migration set is automatically sent to the target environment for import. But, if you didn't register the target environment, or if the target was unreachable during export, you must upload your migration set.

4. Provide the email ID of the person you want to notify after the migration set is imported. You can provide multiple email IDs separated by commas. When the migration set is imported, email notifications are sent to these specified people.
5. Review the roles in your migration set that are missing from your target environment, and create them if required. Remember, you don't have to manually remove the same excluded roles from your migration set every time you import data from your source to target environments. That's because while importing if you exclude any roles from your migration set, those roles are automatically excluded from subsequent imports.

   To review and create roles, do these steps:

   a. Click the **Details** icon.
   b. In the Review Configuration work area, click the Missing Roles tab. New roles are displayed by default.
   c. From the **Show Roles** list, select new, excluded, or all roles to display them:
      - **New Roles**: Displays roles that are available in the source environment but are missing in your target environment.
      - **Excluded Roles**: Displays roles that are neither created in the target environment nor imported.
      - **All Roles**: Displays roles that are a combination of new roles and excluded roles.
   d. Select the roles you want to create in the target environment.

e. Click **Create Role**. You get a warning message. Click **Yes** to proceed.
f. After the roles are created, click **OK** to confirm.
g. Select the **Allow import process to continue even if new roles don't exist in the environment** check box to add the remaining new roles to the excluded list.
h. Click **Save and Close**.

6. To compare the configurations made using Application Composer in the source and the target environments, download and review the configuration reports. You can then troubleshoot issues that occur in one environment but not in another.

   To download and review configuration reports, follow these steps:

   a. Click the **Details** icon.
   b. In the Review Configuration work area, click the Reports tab. The configuration Delta Summary is displayed by default.
   c. From the **Download Report** list, select the report you want to download:
      - **Configuration Delta Report**: Displays the configuration differences between the source and the target environments.

        > **Note:** You can retain up to 10 configuration delta reports at a time.

      - **Source Configuration Report**: Displays the configuration made to artifacts in the source environment.
      - **Target Configuration Report**: Displays the configuration made to artifacts in the target environment.
   d. Click **Save and Close**.

7. You can validate your migration sets and check for missing roles before importing them to the target environment. Based on the validation results, you can choose to proceed with the import as is or take corrective actions before importing.

   To validate your migration sets, do these steps:

   a. Click **Validate**.
   b. Click **Yes** to confirm.

8. Click **Import** to import your migration set into a sandbox instance in the target environment.

   > **Note:** Make sure that you have the Application Administrator (ORA_FND_APPLICATION_ADMINISTRATOR_JOB) role to successfully import the migration set to the target environment.

9. Wait for the status to change to Successfully Imported.
10. Click **Preview** to view your configurations in the sandbox preview mode. You can also edit web service connections and mashup URLs in preview mode.

    > **Note:** You can preview your extensions using Visual Builder Studio. To move these changes to the production environment of Oracle Fusion Cloud Applications, you need to first create an environment in Visual Builder Studio. Then you need to package and deploy the extensions for the production sandbox of Oracle Fusion Cloud Applications. While creating the production sandbox packaging job, make sure that the parameter name matches with the sandbox name in Oracle Fusion Cloud Applications.

11. Provide the email ID of the person you want to notify after the migration set is applied. You can provide multiple email IDs separated by commas. When the migration set is applied, email notifications are sent to these specified people.
12. Click **Apply** when you're ready to apply your configurations to the target environment.

13. Periodically, click the **Refresh** icon to view the current status of the apply action. The migration set is successfully applied to the target environment when the status changes to Applied and Deleted. You can review the process log, if required.

    The process runs asynchronously, so you can close the page and return to it. If problems occur during an Apply action, ask your help desk to log a service request.

Post Migration Tasks

Do these tasks after you apply your migration set to the target environment:

1. Access the target environment and look for any deletions that you must make manually.
2. Delete and recreate any web service connections in the target environment, using the target environment URL and credentials.
3. Deploy all flexfields that display a Patched status.
4. Manually migrate all business processes created in the source environment to the target environment.
5. If a new theme was created and applied in the source environment, and you want to use that theme in the target environment, then go to the Appearance work area and manually apply that theme.
6. After applying configurations, do functional testing to verify the changes. Suppose testing exposes problems with the configurations, such as importing more than what you intended, or the changes weren't what you expected. In such cases, restore your environment to its state before the migration set was applied.
    a. Open the migration set in the Incoming infotile of the Migration page.
    b. Provide the email ID of the person you want to notify after the environment is restored to its state before the migration set was applied. You can provide multiple email IDs separated by commas. When the configuration is restored, email notifications are sent to these specified people.
    c. Click **Restore** to revert to the state before the migration set was applied.
7. Finally, broadcast information to the users that they must sign out and sign in to view the most recent changes.

Preview Mode in Migration

After importing your configurations into the target environment, you can preview them in a sandbox instance before applying them. But you can preview only those configurations that are sandbox aware and have no dependency on non-sandbox artifacts.

You can also edit web service connections and mashup URLs in preview mode before applying the migration set. You can use Application Composer to delete and recreate web service connections and reconfigure mashup URL definitions in the target environment.

> **Note:** When migrating configurations, make sure you have the appropriate administrator roles to test all your changes in preview mode. Also, in Application Composer, you can view only changes related to web service connections and mashup URLs.

What You Can See

The Oracle ADF module is the only module that's sandbox aware. As such, changes to these MDS artifacts from ADF are available for preview:

- Custom objects
- Custom attributes
- Pages

- Workflows
- Groovy scripts
- Triggers
- Lookups
- Attachments
- Data security
- Strings
- Web services

> **Note:** Though visible, these web services won't be functional because their connections aren't migrated.

However, if any of these artifacts have any dependency on non-sandbox aware artifacts, they don't show up in your preview.

What You Can't See

Artifacts that aren't part of the Oracle ADF module aren't visible in preview mode. Here are a few of those non-sandbox aware artifacts:

- BI reports
- BI custom subject areas
- Functional security
- New custom objects
- Email templates
- Workspace configurations
- Notification preferences
- ESS artifacts
- SOA artifacts
- Industry extensions
- Analytics

Related Topics

- SOAP Web Service References for Groovy Scripts
- Register Your Web Application

Best Practices for Migration

Here are some of the recommended best practices that you must follow to migrate your configuration sets.

General Recommendations

Recommendation	Details
Don't cancel or retry the migration process.	The migration process can take some time. So, even if the process seems to be slow, don't cancel or retry. Allow the process to complete without any interruptions.
Test all application changes in the source environment.	You must test all the application changes made in the source environment before starting with the migration process.
Check before you restore your environment after migration.	You can't restore your environment to an earlier configuration if the environment is upgraded after your previous migration. But if a new import is submitted in the upgraded instance, then the most recent import can be reversed.
Run the Metadata Manager in the source environment using Application Composer.	After an upgrade to the source and target environments, if you want to migrate your application changes, run the Metadata Manager in Application Composer on only the source environment to support delta migration.
Check data security privileges as they aren't automatically revoked in the target environment.	During migration, 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 isn't 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 configuration instance during the migration import process and applied to the target environment.
Initiate migration export and import tasks only from the mainline metadata.	You can initiate migration export and import tasks only from the mainline metadata. You can't initiate a migration if you're in an active sandbox.
You can't access certain reports or dashboard pages during an upload or restore activity.	While an upload or restore activity processes Presentation Services changes, these things can happen: • Reports that were submitted by Oracle Enterprise Scheduler to Oracle Analytics Publisher and were scheduled to execute during the process might fail. • The Reports and Analytics pane might not be displayed. • Oracle Analytics Publisher reports might not be displayed on Oracle Business Intelligence Presentation Services analyses or dashboard pages. • Users might not be able to access these Oracle Transactional Business Intelligence features: o Oracle Business Intelligence Answers o Oracle Business Intelligence Delivers o Business Intelligence Composer o Oracle Business Intelligence Interactive Dashboards
Don't restore your environment to an earlier configuration.	When you restore your environment to an earlier configuration, you lose all personalizations you made after the previous migration.
Never modify the metadata extract.	You can't modify the metadata extract. You can migrate metadata from one instance to another.
Do a full migration, if your source and target environments aren't synchronized.	If your source and target environments aren't synchronized, you must do a full migration to synchronize them. When you do a full migration, you move all your changes from the source to the target environment. However, if you made changes to the target environment, then these changes might be overwritten when you do a full migration.

Recommendation	Details
Never make changes in the target or production environment while applying configurations.	You must never make changes in the target or production environment while applying configurations. **Note:** If you make changes in the production environment in emergencies, you must make the same changes in the test environment. Making the changes in the test environment ensures that these changes are included in the subsequent configuration migration. But remember, in Application Composer, you must not create, edit, or delete an object in the production environment. If you do so, you need to ask your help desk to log a service request for refreshing the production-to-test data so that you can avoid any migration issues.
Never create custom objects directly in the target environment.	To avoid metadata inconsistencies while migrating security changes using the Functional Setup Manager, you must not create custom objects directly in the target environment.
Make application changes at the user level or the job role level during the migration process.	During the migration process, you can make application changes at the user level or the job role level because personalizations made at these levels aren't migrated.
Avoid changing security policies directly in the target environment.	After migrating security changes from the source to the target environment, avoid changing security policies directly in the target environment. For example, if you want to change the end date for a security grant, don't change it in the target environment. Instead, make such changes in the source environment and migrate them to the target environment.
Make sure the source and the target environments are on the same release and same patch level and have the same database version.	For secure import, make sure the source and the target environments are on the same release and same patch level and have the same database version. Avoid making any changes directly in the target environment. Instead, make changes in the source environment and migrate these changes to the target environment. However, if unavoidable, you can make only these application changes in the target environment manually: • Server scripts • Field changes made using Page Composer, such as adding and removing fields, and setting a required field. • Label changes You must not make any other application changes manually in the target environment. **Note:** The application changes that you create manually are overwritten by the changes made during subsequent migration from source to a target environment.

Recommendations for Source and Target Environments

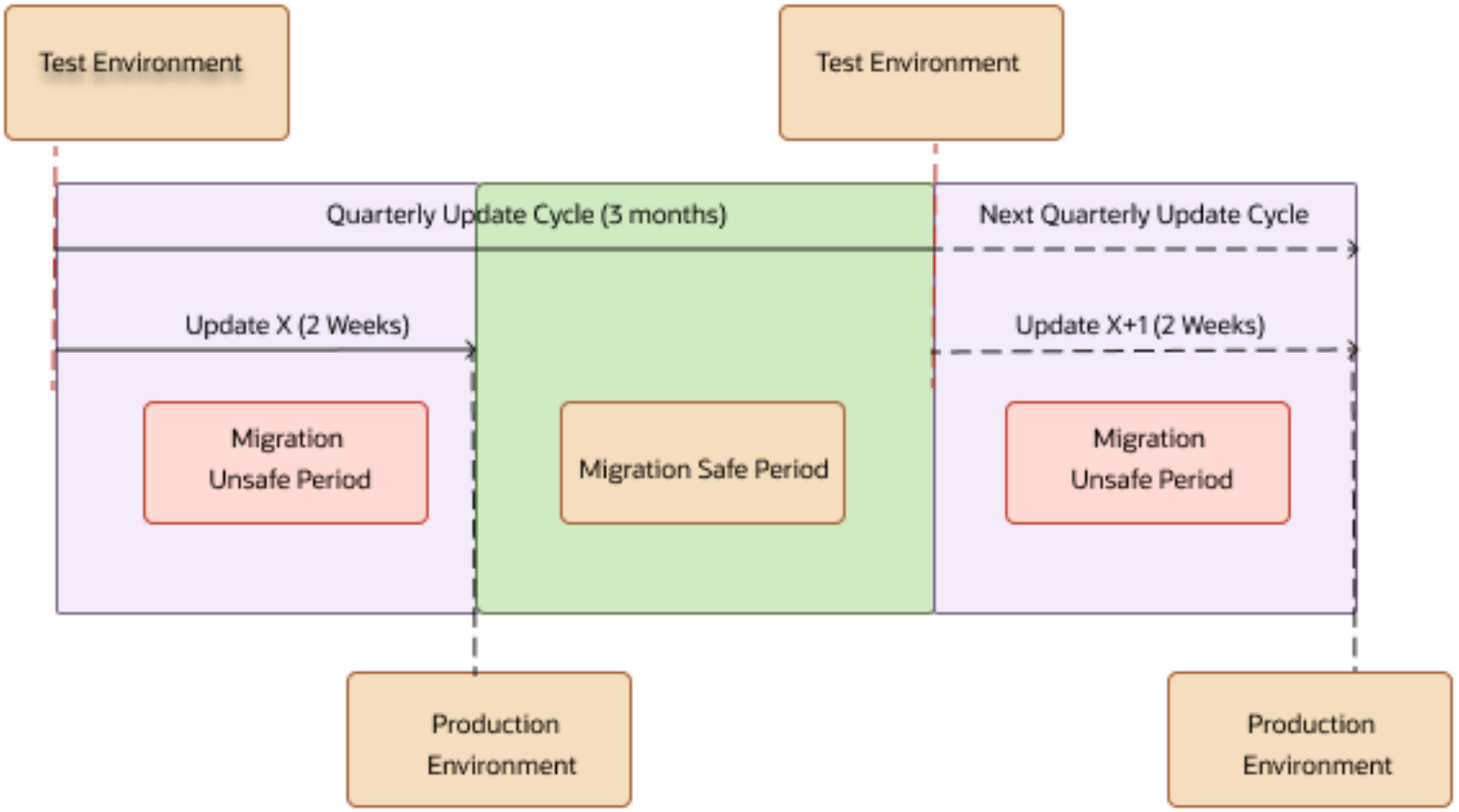
Recommendation	Details
Plan your migration.	Plan to import your migration sets during periods of low system usage or low user activity.
Alert users about your migration.	Alert users about the planned import by adding a banner on the home page using the Page Composer tool.

Recommendation	Details
Avoid making any application changes in the Site layer.	Don't make any application changes using application composer or page composer in the Site layer.
Avoid making any changes in the source and the target environments during export or import.	During export or import, you must not make any changes in the source or target instance.
Avoid publishing your sandbox during the import.	During import, don't publish your sandbox. If you publish your sandbox during this period, the import process might fail.
Make sure you create the reports and reference them to subject areas that were created in the source environment.	You can create reports directly in the target environment. However, make sure you create the reports and reference them to the subject areas that were created in the source environment. Don't create subject areas directly in the target environment.
Migrating your application changes might result in changes to fields that are marked as required in the application.	As a result of the migration, there might be changes to fields that are marked as required in the application. If you get errors while entering data for business objects, make sure to sign out and sign back in after the import.
From the target environment, make sure you delete the flexfield metadata that were deleted from the source environment.	The flexfield metadata that are deleted from the source aren't automatically deleted from the target during the migration process. So, to make sure the source and the target have the same flexfield metadata, you must manually delete the flexfield metadata from the target. Otherwise, you get merge conflicts in the merge log. For example, suppose you created a flexfield segment with the segA segment code in the source environment and assigned it to a **Character** column. This setup is then migrated to a target environment. Now, in the source environment, you change the column assigned to the segment to a **Number** column. As the change is not allowed directly on the setup UI, you delete the segA segment code and recreate it with the same segA segment code using a **Number** column. When this data is migrated, you get merge conflicts in the merge log as the same segA segment with a **Character** column already exists in the target environment.
Lookup values for lookup fields aren't overwritten during the migration import.	The lookup values for lookup fields that exist in both source and target aren't overwritten during the migration import. The values from the source are added to the target and they 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**. **Note:** The lookup values that are deleted from the source aren't automatically deleted from the target during the migration process. So to make sure the source and the target have the same lookup values, you must manually delete the lookup values from the target.
Make sure you use either migration or setup data migration for all lookup migrations.	You can migrate lookups with both migration and setup data migration. However, once you use one of these methods, you must use the same method for all subsequent lookup migrations. Don't use both methods to migrate lookups between the same set of environments.
Don't publish your application changes directly in the target environment.	To make sure changes aren't published in the target environment, set the **Control Publish Sandbox Action in Production Environment** profile option (FND_ALLOW_PUBLISH_SANDBOX) to **No** in the target environment.

Recommendations About When to Not Migrate

Recommendation	Details
While a quarterly update is in progress in the production environment, don't migrate your application changes that were done using the Application Composer.	Quarterly updates take two weeks to complete on your provisioned (non-production and production) environments, during which time the environments are on different updates and patch levels.

The following figure represents the timeline and the recommended safe period for performing migrations using Configuration Set.



Recommendations for Faster Migration

Recommendation	Details
Always perform a delta migration.	This reduces the size of the migration set that must be migrated from source to target and can shave off a significant amount of the time it takes to perform a full migration.
Complete any preparatory work and prerequisites.	To expedite your configuration set migration, perform all preparatory work ahead of time, such as exports and imports.

Recommendation	Details
	Note: Migrating your configurations and Setup and Maintenance capabilities, such as exports and imports are two separate and independent processes and should not be confused as one full migration.
Select only the required optional modules to reduce migration failures.	For faster migration, if there are no application changes in specific components, exclude them from migration. For example, if there are no application changes related to BI or Analytics, exclude the BI component for migration.
Exclude custom Business Intelligence reports from migration by backing them up outside the **Custom** folder.	This creates a smaller-sized archive file for an easy and successful migration to the target environment.
Don't migrate flexfield configurations related to business intelligence, if you don't need them.	If you don't need to migrate flexfield configurations related to business intelligence, set the **Business Intelligence Extender in Configuration Set Migration Disabled** (FND_CSM_BI_EXTENDER_DISABLE) profile option to **Yes**. Setting this profile option to Yes makes the migration process faster. The default value for this profile option is **No**. You can find this profile option in the Manage Administrator Profile Values task in the Setup and Maintenance work area.

Related Topics

- Import and Export Custom Objects
- Export Setup Data Using Offering or Functional Area
- Import Setup Data Using Offering or Functional Area

Best Practices for Managing Web Service Connections

Here are some best practices for managing web service connections during migration.

- Create any SOAP or REST Web Service connections in the target environment with the same name as defined in the source environment.
- Wherever possible, use REST Web Services instead of SOAP Web Services. Using REST Web Services reduces the web service connections' chances of getting changed or deleted after you import your migration set to the target environment.
- Review and recreate web service references in the target environment during migration in preview mode.
- Delete and create web service connections in the target environment using target environment URL and credentials during migration in preview mode.

  Here are the scenarios when you might need to delete and recreate web service connections in the target environment:

Scenario	After Migration	Action Required
New web services in the source environment to be migrated	New web services are partially migrated to the target environment and you get a warning message	Delete and create web service references

Scenario	After Migration	Action Required
Web services exist in both environments (with the same names, relative URLs, and security settings)	Web services are migrated to the target environment	No action required
Web services exist in both environments (with same names, but different relative URLs and security settings)	Web services are migrated to the target environment, but details don't match	For REST web services, delete and recreate web service references For SOAP web services, refresh web service references
Additional web services exist only in the target environment	Web services are deleted from the target environment after migration	Create web service references again, if required

Guidelines for Migrating Global Search Configurations

The migration set that you create using the Migration page doesn't include global search configurations. After importing a migration set, if the global search is configured in the source environment, you must manually set the configurations in the target environment.

Manually Resetting Global Search Configurations

You must adhere to these key guidelines for manually resetting global search configurations:

- If a search view object was activated in the source environment, then after migration, you must activate the object in the target environment. Use the **Manage Search View Objects** task to activate the object.
- If a search view object was deactivated in the source environment, then after migration, you must deactivate the object in the target environment. Use the **Manage Search View Objects** task to deactivate the object.
- The process of activating and deactivating a search view object affects the categories that are shown in the global search. So, after activating or deactivating any search view object, you must adjust the categories to reflect the state of the source environment. Use the **Manage User Interface** task to adjust the categories to be shown in the global search.
- If the category name for a search view object changes, then after migration, you must update the category name. Use the **Manage Search View Objects** task to edit the search view object and then update the category name.
- If the schedule for a search view object changes, then after migration, you must update the schedule. Use the **Manage Search View Objects** task to edit the search view object and then update the schedule.
- If any search view objects were reconfigured in the source environment, then the source environment automatically triggers a reindex. After migration, you need not reconfigure the search view objects. However, you must trigger a full reindex in the target environment for any search view objects that were reconfigured

in the source environment. To trigger a full reindex, select the search view object in the **Manage Search View Objects** task, and click the **Full Re-Index** button.

FAQs for Migration

What's the difference between a delta migration and full migration?

A delta migration moves only those changes that were made to an environment after the last migration was done.

On the other hand, a full migration moves all changes made to an environment.

For instance, let's say Alpha is your source environment, and Beta is your target environment. You create object X in Alpha environment and do a migration. Then, you create object Y in Alpha environment and do another migration. If the second migration is a delta migration, only the object Y is moved. However, if it's a full migration, both the objects X and Y are moved.

When can I do a delta migration?

You can do a delta migration if you meet these two conditions.

- You have registered your target environment in your source environment, without any issues.
- Both of your environments are still synchronized.

Why aren't my environments synchronized?

Your environments might not be synchronized if you did either of these things.

- Directly configured the target environment.
- Upgraded one of the environments or both of them.

Related Topics

- How can I synchronize my environments?

How can I synchronize my environments?

If your source and target environments aren't synchronized, you must do a full migration to synchronize them again.

Related Topics

- Migrate Your Configurations

Can I make changes in my sandbox in preview mode while migrating configurations?

It's not recommended to make any changes, apart from editing web service connections and mashup URLs, in your sandbox in preview mode while migrating configurations.

Troubleshooting Configurations

View and Diagnose Application Changes

You can use the Manage Configurations dialog box to view and diagnose changes made to the application pages. Application changes are role-dependent and by default, the Manage Configurations dialog box displays the changes that the signed-in user had performed.

Before you view application changes, make sure that you have administrative privileges to access the Manage Configurations dialog box. To view the page that contains the changes, do these tasks:

1. Click your user image or name in the global header, and on the **Settings and Actions** menu, select **Manage Configurations**.
2. On the Manage Configurations dialog box, use the Search text field to search for the page, page fragment, or task flow.

You can also use the Manage Configurations dialog box to do these tasks:

- View the application changes for a user in the Current Context column.
- View all application changes made at the site level, and for any user, in the All Layers column.
- View application changes made by more than one user.

To diagnose issues pertaining to application changes, determine whether the changes have been applied to a page. You can use the Manage Configurations dialog box to determine if page-level changes exist. If a page modification causes problems, such as a user interface component disappears from a page, you can export the application changes and examine the document file.

Related Topics

- Tools for Moving and Troubleshooting Configurations
- Overview of Context Layers

Download Configurations

You can download all configurations of a page for all layers using the Download Configuration for All Layers link in the Manage Configurations dialog box. You can also download and view configuration set reports after exporting configurations.

Download Configuration Files From a Page

Let's say the configurations you made to a page resulted in an error. You need to download the configurations you made to that page to diagnose what went wrong. You can use the Manage Configurations dialog box to download all the configuration XML files for that page. The downloaded files contain the changes made to all the layers of that page, and you can use them to diagnose configuration issues.

If you know the file path to the page you want to download configurations from, then follow these steps:

1. Click your user image or name in the global header, and select **Manage Configurations**.
2. In the **Search** field, enter the file path to your page and click the **Search** icon.
3. Click **Download Configuration for All Layers**.

If you don't know the file path, then follow these steps:

1. Create and activate a sandbox.
2. Go to the page you want to download configurations from.
3. Click your user image or name in the global header, and select **Edit Pages**.
4. Click the **Structure** tab.
5. Click any component on the page.
6. In the confirmation window, click **Edit**.
7. Go back to the **Add Content** tab.
8. Click **Manage Configurations**.
9. Scroll down and click **Download Configuration for All Layers**.

Download Configuration Set Reports

After exporting configurations, you can view and download a configuration set report that contains a list of all configurations available in a configuration set. This report includes all new or updated artifacts:

- Objects
- Fields
- Pages
- Business intelligence (BI) changes

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

To download Configuration Set Reports, follow these steps:

1. Click **Navigator** > **Configuration** > **Migration**.
2. From the Outgoing tab, click **Content Read Me** to download the Readme file listing all the configurations you exported.

Related Topics

- Overview of Sandboxes

4 Page Content and Layout

Overview of Page Modification

You can activate a sandbox and use Page Composer to change page content, layout, and more. You use Page Template Composer to configure the global page template if, for example, you want to modify the global header.

Similarly, you can use other tools to create pages and change UI text, themes, infolets, and so on.

Here are some of the things that you can do using various tools:

- Change the UI text using tools, such as User Interface Text.
- Change the look and feel of the application using the Appearance page.
- Open an infolet page and modify it using Page Composer.
- Create and manage pages for hosting third-party applications using Page Integration page.
- Change the global page template using Page Template Composer.

Note: You can't use Page Composer to configure your home page. Instead, you can use other tools such as, Structure and Appearance for home page configurations.

Related Topics

- Guidelines for Using Page Composer
- Get Started with Configuring Navigation
- How can I configure how infolets are displayed in Fusion Applications?
- How do I define the display settings of the home page?

Overview of Using Page Composer

Guidelines for Using Page Composer

For certain applications, use Page Composer to configure application page components, such as page content and layout for other users. Here are the changes you can make using this tool.

- Add, move, delete, show, and hide fields
- Add an EL expression to a field
- Rearrange fields
- Change a page layout
- Manage saved searches
- Modify attributes for a flexfield on a page

Before modifying pages, you must do these tasks:

- Understand the typical workflows for working with application changes.
- Verify that the page can be modified. To do so, you can check if either the **Edit Pages** or the **Edit <Page Name> Pages** menu item is available in the **Settings and Actions** menu. If not, then that means you can't modify the page. Click your user image or name in the global header to open the **Settings and Actions** menu.
- Confirm that your privileges are sufficient for modifying the page.
- Activate a sandbox that has the Page Composer tool in it. Make sure the context layer of your sandbox is supported by the page you want to edit. Otherwise, you won't be able to edit the page.

Related Topics

- Overview of Sandboxes
- Create and Activate Sandboxes
- Why doesn't the page I am editing with Page Composer work with the sandbox context?
- Overview of Context Layers
- Best Practices for Using Page Composer in Sandboxes

How to Open Page Composer

To open Page composer, create and enter a sandbox with Page Composer as one of the tools and select **Page Composer** from the **Tools** menu. Make sure the context layer that you want is selected.

If you're planning to use Page Composer to make changes for specific job roles, then you must create a separate sandbox just for Page Composer. You can't use other tools to make changes for specific job roles.

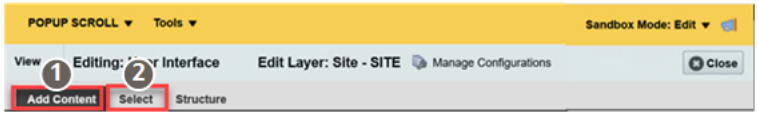
1. Activate a sandbox. The active sandbox name appears at the top of the page.
2. Click the user image or name in the global header, and on the **Settings and Actions** menu, select **Edit Pages**.

Page Composer Modes

When configuring UI pages, you can use Page Composer in either of these modes:

- Add Content (callout 1): Use this mode to move, add, and edit regions on dashboards and external-facing partner pages. You also use this mode to create saved searches, specify search result content, and to navigate in the UI when you're in Page Composer. By default, Page Composer opens in the Add Content mode. You can also click the Add Content tab on the top left corner of a page to select this mode.
- Select (callout 2): Use this mode to configure the properties of different UI elements, such as fields and table columns. You can click the Select tab to activate this mode.

Here's the page composer work area, where you use the Add Content (1) or Select (2) mode to edit page components.



Add Content Mode

Use the Add Content mode to do these tasks:

- Configure all of the external-facing landing and transactional partner management pages.

- Configure the table display of search results.
- Create saved searches.
- Navigate to other pages while working in Page Composer.

When you open Page Composer, the tool opens in the Add Content mode automatically. On pages that support configuration in this mode, different regions are highlighted with borders and display configuration controls.

- A bar appears at the top of the page indicating that you're editing the page in Page Composer.
- You can click **Change Layout** to change the column layout of the page.
- You can move a region, by grabbing its toolbar and dragging it to a different position.
- You can add a new tab.

You can also make these changes in each region:

- Click **Close** to remove a region.
- Click the Add Content tab to add content.
- Click one of the icons to the right of the Add Content button to add a new region. These icons specify where the new region is created.

Select Mode

In the Select mode, a border appears around individual fields and other UI elements as you move your cursor over them. When you click within the border of the UI elements that support configuration, these options are displayed:

- Edit Component
- Edit Parent Component

You use the Component Properties dialog box to make your configurations. All configurations aren't available on all the UI elements, so the content of this dialog box varies.

- To make a field read-only, you select the **Read Only** option.
- To make a field required, you select the **Required** option.

  > **Note:** Before you can make a field required, the field must display a value. If it doesn't, then close the Component Properties dialog box and enter a value first.

- To hide the field, you deselect the **Show Component** option.

> **Note:** Some UI elements are protected from updates to preserve the product business logic. For example, if a field is read-only because of a security privilege defined in the application, you can't edit it in Page Composer.

Page Composer Views

You can use either Design view or Source view for viewing and changing page content and layout in Page Composer.

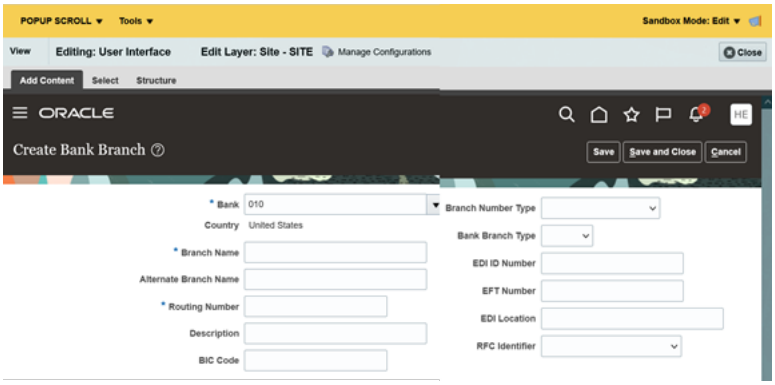
To open a view option, select it from the **View** menu at the top left corner of the page. Although both views share many common page modification features, you can use some unique features in each view.

Note: By default, you get the Design view to modify pages. But for some pages, you must use the Source view. Set the **Source View for Page Composer Enabled** profile option (FND_PAGE_COMPOSER_SOURCE_VIEW) to **Yes** so that the **View** menu is available for you to switch between the two views. You can find the profile option in the **Manage Administrator Profile Values** task in the Setup and Maintenance work area.

Design View

In the Design view, you see one region that shows a WYSIWYG rendering of the page and its content. In this view, you can add content, edit the component properties and the parent component properties, change the page layout, and delete a component.

Here's the design view, where you can add content and edit page components.

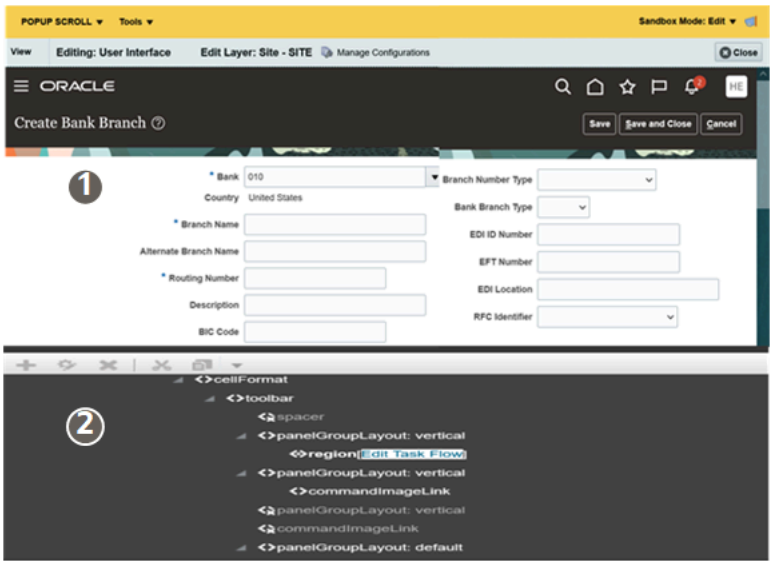


Source View

In the Source view, you see two regions:

- The selection pane (callout 1) shows a WYSIWYG rendering of the page and its content.
- The source pane (callout 2) shows 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.

Here's the source view, where you use the selection pane (1) and the source pane (2) to edit page components.



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 do these tasks:

- 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 Components** on the view header to work with components and access their properties. You can also right-click the object in the hierarchy and click **Edit**.

Page Component Properties

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. You can open this dialog box by selecting the component that you want to edit and clicking **Edit Component**. You can see properties of similar functions in 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 might 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 might have a parameter that provides a choice between units of measurement.
Display Options	Includes these settings that affect the chrome of a component: • 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 inside the parent component. The parent component controls the child components.

Tab	Description
Style Content Style	Settings that affect the look and feel of the component chrome or the component contents. For example, font, color, and dimension. These settings override corresponding values from a parent object, such as a component, page, and application, providing an opportunity to fine-tune appearance. Some style properties might 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 associated with all components on the current page. For example, when you select a check box within the current context, it's an event. The code that drives the result of an event, such as making another component visible is an event handler.

Considerations for Modifying Objects That Appear on Multiple Pages

Use Page Composer to modify objects that appear on multiple pages. Whether your application changes 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 modify objects that appear on multiple pages.

Appearance of Object Modifications Based on Shared and Non-Shared Task Flows

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

If the object is...	Then the modifications...
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 application changes that you make in Page Composer.

Overview of Using Visual Builder Studio

For certain applications, use Visual Builder Studio to make changes to various page components. Changes made using Visual Builder Studio are called extensions. Here are the changes you can make using this tool.

- Display fields from your data source to the UI
- Add business logic to display certain layouts

- Edit variables, constants, and events as needed
- Change the default content
- Rearrange fields
- Hide fields
- Create your own App UIs and deploy them alongside other applications in your Oracle Fusion Cloud Applications instance. In other words, you can now build robust applications that are managed as part of the same Fusion Applications ecosystem as the application UIs produced by Oracle. For more information, see *Extending Oracle Cloud Applications with Visual Builder Studio* guide. You might want to start with *The Basics*, just to get oriented.

Your test environment is automatically populated with an instance of Visual Builder Studio. Depending on what tasks you want to do with Visual Builder Studio, here are a few recommendations:

- If you're creating an extension, you can deploy it to any of your environments (test or production), not just the test environment. For instructions, see *Set Up VB Studio to Extend Oracle Cloud Applications*.
- If you want to use this instance of Visual Builder Studio to create visual applications (new custom applications) rather than extensions, you must deploy visual applications to a standalone instance of Visual Builder. This is an entirely separate product that may require a separate license. For more details, contact your sales account representative.

Note: After you install an instance of Visual Builder, you need to set up Visual Builder Studio using the instructions in *Create and Set Up a Project for Development (Different Identity Domain)*. Only after setting it up, you can start developing visual applications.

How to Verify Whether a Page Can Be Extended Using Visual Builder Studio

If you see the **Edit Page in Visual Builder Studio** option in **Settings and Actions** menu of a page, that means you must use Visual Builder Studio to make changes to that page. If you don't see the option, that means the page can't be modified using Visual Builder Studio.

Note: Click your user image or name in the global header to open the **Settings and Actions** menu.

Here are a few things to know about using Visual Builder Studio:

- If you need to make changes to the data model, you must first use other configuration tools, such as Application Composer, and then use Visual Builder Studio to make the changes in the UI.
- You can make additional changes to the data model later using a sandbox and associate that sandbox with the workspace (your private work area) in Visual Builder Studio.
- When you're ready to merge your application extension changes with the mainline repository, you should publish the sandbox. Publishing the sandbox before merging your application extension changes can help avoid potential problems resulting from using two different data models. The application extension changes are then automatically deployed to a test environment.

Related Topics

- Administering Visual Builder Studio
- Extending Oracle Cloud Applications with Visual Builder Studio

Overview of Using Application Composer

Use Application Composer to extend certain applications by creating fields, objects, and relationships between objects. Here are some of the changes you can make using this tool.

- Modify objects by adding new fields, or create entirely new objects.
- Create foreign key-based relationships between two objects.
- Modify user interface pages by exposing your newly created fields for an object, or create an entirely new work area for your custom objects.
- Expose object relationships on pages in the form of subtabs.
- Write application logic, such as triggers and validation rules, for an object or for use across multiple objects.
- Implement functional and data-level security for custom objects.
- Enable custom objects for reporting by creating custom subject areas.

Related Topics

- Configuring Applications Using Application Composer

Differences Between Using Page Composer and Application Composer

You can use multiple tools to configure the UI. This topic highlights the differences between the types of configurations you can do using Application Composer and Page Composer.

Key Differences

Basis of Comparison	Page Composer	Application Composer
Configuring the page you're viewing	Configure a UI page you're viewing and see your changes reflected immediately on that page. You can use Page Composer to make changes to one page at a time. For example, if you make a field required on the Edit Infolets page, the field isn't automatically required on the Create Infolets page.	You make your configurations in Application Composer. You can then test your work by navigating to the appropriate page. Because you use Application Composer to configure application objects, if you make a field required, it affects all interfaces where that field appears.
Supported interfaces	Application UI	• Application UI • Mobile • Microsoft Outlook

Basis of Comparison	Page Composer	Application Composer
Creating your own fields and application objects	Not supported.	You can add custom fields and entirely new objects.
Creating saved searches	Create and edit saved searches for users. The saved searches provide saved search criteria to speed up common searches. You can also use them as filters in actionable infolets.	Not supported.
Tailoring your configurations for users with different job roles	Supports different levels of configurations for application pages.	Supports configurations by job role or for all users.
Configuring external-facing partner pages	You must use Page Composer to configure all external-facing partner pages, such as the partner registration page and the partner portal landing page.	You must use Application Composer to configure only for partner pages internal to your organization, such as the partner Overview page.

What You Can't Do in Page Composer

Using Page Composer, you can't do these tasks:

- Change either the data or functional security for a page. While you can hide some components, you must change the security setup to control access.
- Filter or reorder lists of values.
- Add any code, validation logic, or scripting to the buttons or fields.
- Add call outs to other APIs to populate fields with values.
- Change the selections that appear in the **Navigator** menu.
- Configure the Search region inside Search and Select dialog box.

Modify Page Content and Layout Using Page Composer

Add and Hide Components Using Page Composer

You can select components like task flows, portlets, and layout components from the resource catalog and add them to your page. You can use the **Show Component** property on the Component Properties dialog box to show or hide page components.

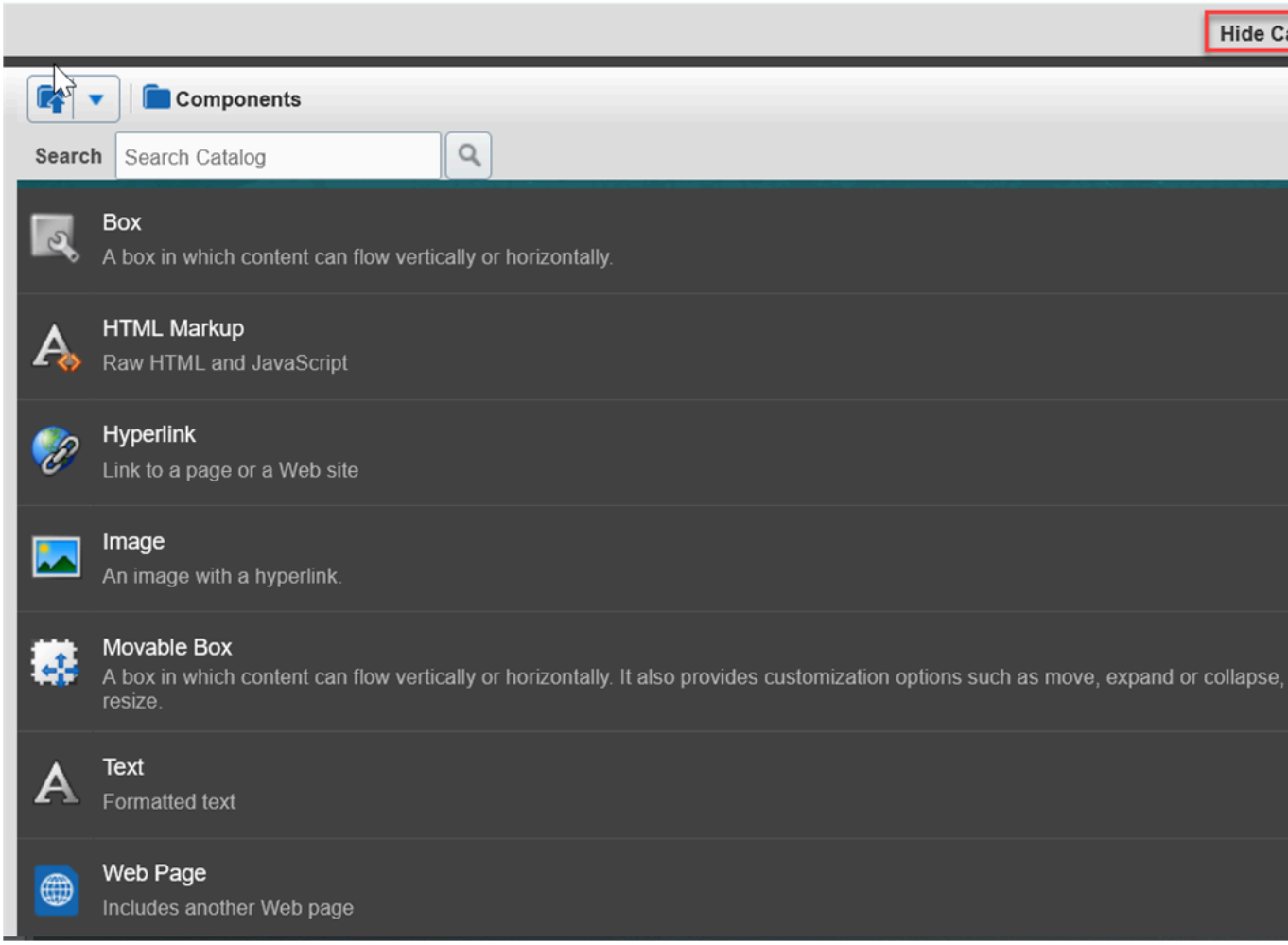
Before You Start

In Page Composer, you get the Design view by default to edit pages. But on some pages, you must use the Source view. If you don't see the Source view, set the **Page Composer Enabled** (FND_PAGE_COMPOSER_SOURCE_VIEW) profile option to **Yes**. In the Setup and Maintenance work area, go to the **Manage Applications Core Administrator Profile Values** task in the Application Extensions functional area.

Add Components in Design View

Here's what you do:

1. Activate a sandbox that has the Page Composer tool in it. Make sure the context layer of your sandbox is supported by the page you want to edit. Otherwise, you won't be able to edit the page.
2. Click your user image or name in the global header and select **Edit Page** from the **Settings and Actions** menu.
3. In the Design View, you can add components using the Select tab.
    a. Click **Show Catalog** on the top right corner of the page to view the resource catalog pane.
    b. In the resource catalog pane, open the **Components** folder.



    c. Click the **Add** button associated with the component you want to add to your page and click **Close**.
    d. Click **Hide Catalog** on the top right corner of the page to hide the resource catalog pane.
4. Select the component you just added.
5. In the Source View pane, click the **Show the properties of <component>** icon.

6. Change the component properties, as appropriate. For example, if you added the Text component, enter the text you want to display.
   If you have added an HTML markup with references to an external CSS, here are a few things to keep in mind:
   - The CSS has an `Access-Control-Allow-Origin` header on its server.
   - Your link tag has a `crossorigin` attribute specified.
7. Click **Close** in the Page Composer toolbar to save your changes.

Add Components in Source View

Here's what you do:

1. Activate a sandbox that has the Page Composer tool in it.
2. Click your user image or name in the global header and select **Edit Page** from the **Settings and Actions** menu.
3. From the **View** menu, select **Source**.
4. Hover over the page until a blue border appears around it.
5. Click anywhere in the page component as long as the border is visible.
6. In the Source View pane, click the **Add content into the selected component** icon.
7. In the Add Content dialog box, open the **Components** folder.
8. Click the **Add** button associated with the component you want to add to your page and click **Close**.
9. Select the component you just added.
10. In the Source View pane, click the **Show the properties of <component>** icon.
11. Change the component properties, as appropriate. For example, if you added the Text component, enter the text you want to display.
    If you have added an HTML markup with references to an external CSS, here are a few things to keep in mind:
    - The CSS has an `Access-Control-Allow-Origin` header on its server.
    - Your link tag has a `crossorigin` attribute specified.
12. Click **Close** in the Page Composer toolbar to save your changes.

Here's the list of components you can add to your page:

Component	Description
Box	You can use this component to place content on a page.
HTML Markup	You can use this simple editor to enter raw text and HTML tags, including Javascript embedded in HTML <script> tags.
Hyperlink	You can use this component to add a link to a page or a website. It can point to a location in the application that's either internal or external.
Image	You can use this component to add a picture, a logo, or a linked image to a page.
Movable Box	You can use this component to place content on a page. You can also move, expand or collapse, and resize your content.
Text	You can use this component to add UI text or any other kind of informative content to a page.

Component	Description
Web Page	You can use this component to provide URLs of other web pages within the context of a WebCenter application page.

Hide Components Manually

Use the **Show Component** property to specify whether the component appears to users. 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.

If the component is a 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**.

Hide 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? What action or event must happen?	Check box 2 is selected	Determines what the occurrence, or event, is. Determines the component that triggers the event. Determines what expression to write.
What happens when the condition is met? What happens when the event happens?	Button "B" appears. Show the component: Button B (The implication is that button B is hidden until the event occurs.)	Determines the effect of the action.
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**.
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.

Related Topics

- How can I move page components?
- How can I delete components from a page?
- Best Practices for Using Page Composer in Sandboxes
- Modify Tabs on Pages
- How do I update existing setup data?

Modify UI Components Using Page Composer

You can modify your UI components by updating their properties. For example, you can change the field labels, or make a check box required.

Any changes you make apply only to the page that you're on. And they apply to all or specific groups of users, depending on the context layer that you select. For example, you can make changes only for users with a specific job role. In which case, you must also have that job role to test your changes in the sandbox.

Before You Start

In Page Composer, you get the Design view by default to edit pages. But on some pages, you must use the Source view. Set the Source View for **Page Composer Enabled** profile option (FND_PAGE_COMPOSER_SOURCE_VIEW) to **Yes** so that the View menu is available for you to switch between the two views. In the Setup and Maintenance work area, use the **Manage Applications Core Administrator Profile Values** task in the Application Extensions functional area.

Modify UI Components

1. Activate a sandbox that has the Page Composer tool in it. Make sure the context layer of your sandbox is supported by the page you want to edit. Otherwise, you won't be able to edit the page.
2. Click your user image or name in the global header and select **Edit Pages** from the **Settings and Actions** menu.
3. Either stay in the Design view to get the main properties, or select Source from the View menu to get all the properties.
    - If you stay in the Design view, do these tasks:
        i. Find the UI component you want to modify.
        ii. Click the Select tab that's right after the Page Composer bar.
        iii. Hover over the UI component until a blue border appears around it.
        iv. Click anywhere in the component as long as the border is visible and select **Edit Component**.
    - If you select Source from the View menu, do these tasks:
        i. Hover over the UI component until a blue border appears around it and select the component.
        ii. Click the **Show the properties of <component>** icon in the Source View pane.
4. Update the component's properties, which may include some of the properties listed here

Property	Description
Label	The display text for the component. For example, the field prompt or the single prompt for a group of check boxes.
Read only	Indicates whether the component is rendered as read-only. You can set the value to **True**, **False**, or **EL expression**.
Rendered	Indicates whether the component is available on the page. You can set the value to **True**, **False**, or **EL expression**.

Property	Description
Required	Indicates whether users must enter some data for the component before they save. You can set the value to **True**, **False**, or **EL expression**.
ShortDesc	Text that appears when users hover over the component.
Show Component	Indicates whether the component is visible or hidden to users.
Show Required	Indicates whether an asterisk is displayed to tell users that the component is required.

5. Click **OK** to save your changes.
6. If available, click **Save and Label** to label your changes so that you can revert to this configuration at any point in the future. Labels are saved with a composer_ prefix. For example, if you enter myLabel, the label is composer_myLabel.
7. Click **Close** on the Page Composer bar.

Related Topics

- Examples of How the Appropriate Configurations and Personalizations are Applied
- Overview of Sandboxes
- Create and Activate Sandboxes
- Best Practices for Using Page Composer in Sandboxes
- Set Profile Option Values

Modify Tabs on Pages

This example demonstrates how to modify 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 changes for?	All users
What changes are you going to make?	Hide the Export Configuration tab from the Manage Exports page

Before You Start

Activate a sandbox that has the Page Composer tool in it. Make sure the context layer of your sandbox is supported by the page you want to edit. Otherwise, you won't be able to edit the page.

Modify Tabs

Do the following:

1. From the Navigator menu, select **Tools** > **Export Management**.
2. Click your user image or name in the global header, and select **Edit Pages** in the **Settings and Actions** menu.
3. Make sure you've selected the **Site** layer as the context 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 modify.
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 Component**.
8. In the Component Properties dialog box, either stay on the Tabs tab or click the Children tab.
    a. If you stay on the Tabs tab, do these tasks:
        i. Select **Export Configuration** and select the **Hide this tab** check box.
        ii. Click **Apply** and then **OK**.
    b. If you select the Child Components tab, do these tasks:
        i. Deselect **Incentive Compensation**.
        ii. Click **Apply** and then **OK**.
9. Close Page Composer.

Related Topics

- Overview of Sandboxes
- Create and Activate Sandboxes
- Add and Hide Components Using Page Composer

Change Page Layout

You can configure your page layout to define the number, placement, and orientation of content regions. You can set the layout style while creating a page or 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 that has the Page Composer tool in it. Make sure the context layer of your sandbox is supported by the page you want to edit. Otherwise, you won't be able to edit the page.

Change Page Layout

1. Open the page you want to edit.

2. Click your user name or image and select **Edit Pages** to open Page Composer.
3. From the View menu, select **Design**.
4. Click **Change Layout**.
5. Select the new layout.

Related Topics

- Overview of Sandboxes
- Create and Activate Sandboxes

Modify Dialog Box Content

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

Before You Start

Activate a sandbox that has the Page Composer tool in it. Make sure the context layer of your sandbox is supported by the page you want to edit. Otherwise, you won't be able to edit the page.

Modify Dialog Boxes

To modify 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 modify 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

- Overview of Sandboxes
- Create and Activate Sandboxes

Make Application Changes Visible Based on User Roles

This example demonstrates how you can make application changes 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 changes visible?	Hiring managers, Sherry Callaway and Terrance Whitaker **Tip:** Create a role just for testing application changes. Call it `DEVCUST_TEST_ROLE`. When you're sure that the application change works, change the security to the appropriate role.
Which expression to add for verifying whether a user has the appropriate privilege?	`#{securityContext.userGrantedPermission['MANAGER_REPORTS_LINK_PRIV']}`

As a prerequisite, activate a sandbox that has the Page Composer tool in it. Make sure the context layer of your sandbox is supported by the page you want to edit. Otherwise, you won't be able to edit the page.

Make Application Changes Visible to a User Role

Follow these steps:

1. Create a privilege.
2. Add the '`MANAGER_REPORTS_LINK_PRIV`' object to '`DEVCUST_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 modify, 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 permissions that render UI components based on the user's access rights. (Oracle JDeveloper is not available in Oracle Cloud implementations.)

Related Topics

- Overview of Sandboxes
- Create and Activate Sandboxes

FAQs for Page Content and Layout

How can I move page components?

To move page components using Page Composer, do these tasks.

1. In Design view, drag and drop the component.
2. In Source view, do any of the following tasks:
    a. Cut and paste the component.
    b. Drag and drop the component.
    c. Open the Component Properties for the container component and rearrange the components on the Child Components tab.

Related Topics

- Add and Hide Components Using Page Composer

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.

Related Topics

- Add and Hide Components Using Page Composer

What happens if my configurations make the page inaccessible?

If your configurations make a page inaccessible, ask your help desk to log a service request.

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 ready-to-use state, click Reset Page in Page Composer (Design or Source view).

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

What do I do if Page Composer closes while I am navigating in the UI?

If you're using Page Composer in Design mode and trying to navigate between different objects in the application, the Page Composer sometimes closes. That's because it's designed to configure one object at a time.

When you no longer see the Page Composer bar at the top of your screen, simply restart Page Composer. Your previous work is preserved in the active sandbox, so you don't have any data loss.

Can I use Page Composer to configure all elements on a page?

No, you can't use Page Composer to configure all UI elements on a page.

Some UI elements are protected from updates to preserve the product business logic. For example, if a field is read-only because of a security privilege defined in the application, you can't edit the field.

You can't select some page elements in the Select mode of Page Composer. For others, the **Edit Component** option is disabled or only a subset of the properties are available in the Component Properties dialog box.

> **CAUTION:** Don't edit the **Taskflow Id** parameter in Page Composer Task List properties. This may damage page metadata.

In general, you can configure at least some of the properties of the following page components:

- af:activeOutputText
- af:inputComboboxListOfValues
- af:inputDate
- af:inputFile
- af:inputListOfValues
- af:inputNumberSlider
- af:inputNumberSpinbox
- af:inputRangeSlider
- af:goLink
- af:query
- af:inputText
- af:outputFormatted
- af:outputText
- af:panelLabelAndMessage
- af:panelFormLayout
- af:panelHeader
- af:richTextEditor
- af:selectBooleanCheckbox
- af:selectBooleanRadio
- af:selectManyCheckbox
- af:selectManyChoice
- af:selectManyListbox
- af:selectManyShuttle
- af:selectOneChoice
- af:selectOneListbox
- af:selectOneRadio
- af:selectOrderShuttle
- af:showDetailHeader

- af:table
- af:column
- fnd:applicationsTable
- af:commandMenuItem
- af:menu
- af:panelGroupLayout
- af:panelTabbed
- af:group
- af:commandButton

Why can't I see the site-level changes in the application after publishing my sandbox?

It might be because you have user-level changes that are hiding or overriding the site-level changes. To see the site level changes, you need to remove or reset your personalization.

Related Topics

- How can I reset all personalization changes made by a specific end user?

Modify Global Page Template Using Page Template Composer

Configure the Global Header

The global header is the uppermost region in the UI, and it's available on all pages. It's where you find, for example, the Navigator. To modify the global header, use Page Template Composer to configure the global page template.

- Add components, such as add HTML markup
- Edit components, such as add expression language to hide the link tag for HTML markup
- Delete components, such as remove the link tag for HTML markup

Before You Start

In Page Template Composer, you get the Design view by default to edit pages. But on some pages, you must use the Source view. Set the Source View for **Page Composer Enabled** profile option (FND_PAGE_COMPOSER_SOURCE_VIEW) to **Yes** so that the View menu is available for you to switch between the two views. In the Setup and Maintenance work area, use the **Manage Applications Core Administrator Profile Values** task in the Application Extensions functional area.

Add Components to the Global Header

Here's what you do:

1. Activate a sandbox that has the Page Template Composer tool in it, and make sure the context layer you want is selected.
2. Click your user image or name in the global header and select **Edit Global Page Template** from the **Settings and Actions** menu.
3. Select **Source** from the **View** menu.
4. Hover over the global header until a blue border appears around it.
5. Click anywhere in the global header component as long as the border is visible.
6. Click the **Add content into the selected component** icon in the Source View pane.
7. In the Add Content dialog box, click the **Components** folder.
8. Click the **Add** button associated with the component you want to add and click **Close**.
9. Select the component you just added and click the **Show the properties of <component>** icon in the Source View pane.
10. Change the component properties, as appropriate. For example, if you added the Text component, enter the text that you want to display.

    Note: Here's what you need to ensure if the component you added is HTML markup with references to an external CSS:
    - The CSS has `Access-Control-Allow-Origin` header in its server.
    - Your `link` tag has `crossorigin` attribute specified.

11. Click **Close** in the Page Template Composer toolbar to save your changes.

Edit Components in the Global Header

Here's what you do:

1. Activate a sandbox that has the Page Template Composer tool in it, and make sure the context layer you want is selected.
2. Click your user image or name in the global header and select **Edit Global Page Template** from the **Settings and Actions** menu.
3. Select **Source** from the **View** menu.
4. Select the component that you want to edit.
5. Click the **Show the properties of <component>** icon in the Source View pane.
6. Edit the component properties, and click **OK** to save your changes.
7. Click **Close** in the Page Template Composer toolbar to save your changes.

Delete Components from the Global Header

Here's what you do:

1. Activate a sandbox that has the Page Template Composer tool in it, and make sure the context layer you want is selected.
2. Click your user image or name in the global header and select **Edit Global Page Template** from the **Settings and Actions** menu.
3. Select **Source** from the **View** menu.
4. Select the component that you want to delete.
5. Click the **Delete <component>** icon in the Source View pane.

6. Click **Close** in the Page Template Composer toolbar to save your changes.

Related Topics

- Set a Banner Message in the Global Header

FAQs for Page Template Composer

Can I configure the look and feel of my application using the Page Template Composer tool?

No, you should always use the Themes page of the Appearance work area to configure the look and feel of your application.

Saved Searches

Overview of Saved Searches

Users can create and edit saved searches for themselves on pages where the Saved Search option is available. Users with appropriate roles can make their saved searches available for all users, using the Page Composer tool at the site layer.

The saved search captures search criteria that are used for getting the desired search results. You can create saved searches for keyword searches with filters as well as searches with multiple criteria.

Note: You must create or edit saved searches only at the site layer to make them available for all users.

Here are the steps to create and edit saved searches for all users:

1. Activate a sandbox that has the Page Composer tool in it, and make sure the context layer you want is selected.
2. On the search page that has a **Save** button, click your user image or name in the global header, and select **Edit <Page Name> Pages** in the **Settings and Actions** menu.
3. From the View menu, select **Design**.

   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.

4. Create and edit saved searches.
5. Save your changes and close Page Composer.
6. After testing your changes, publish the sandbox to make your changes available to all users.

Related Topics

- Overview of Sandboxes
- Create and Activate Sandboxes
- Save Keyword Searches with Filters
- What gets saved when I create a saved search for searches with multiple criteria?

Save Searches with Multiple Criteria

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 after 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 appropriate roles, you can also create and edit saved searches for other users using Page Composer.

Create 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.

Change 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.

Change Settings or Rename and Delete 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.
   - **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 editing the saved search within 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

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

Save Keyword Searches with Filters

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 appropriate roles, you can also create and edit saved searches for other users using Page Composer.

Create and Edit 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 organize filters as follows, depending on what's available to you:
   - Use the **Add** or **Reorder** buttons.
   - Click the **Personalize** icon to show, hide, or reorder filters, and click **OK**.

4. Set filter values or select filters, and click the **Save** button.
5. Name your saved search.
    - To create a saved search, enter a new, unique name.
    - 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.

Change Settings or Rename and Delete 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 user-defined, 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 editing the saved search within 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 user-defined saved searches only, you can also rename or delete the saved search.
6. Click **OK**.

Related Topics

- Use Filters with Keyword Searches

How can I create saved searches for all users or specific users in Fusion Applications?

As an admin, you can create saved searches with the Page Composer tool opened in a sandbox. The context level of the sandbox makes your saved searches available to all or select users.

1. In Sandboxes, create a sandbox or enter an existing one. Make sure the sandbox has the Page Composer tool.
2. From the sandbox Tools menu, select **Page Composer**.
    - The default context level for the tool is **Site**, which applies to all users.
    - To apply to specific users, click the **Edit** icon and select a level other than **Site**.
3. Open the page where you want to create saved searches. For example, from the **Navigator**, select **Tools**, and then select **Transaction Console**.

4. Click your user image or name in the global header and select **Edit Page** from the **Settings and Actions** menu.
5. Set the filters you want for the saved search (in this example, on the Transaction Manager: Transactions page).
6. Click **Save** and optionally set this saved search as **Default** for all affected users.
7. Test your changes and publish the sandbox.

Infolets

Overview of Configuring Infolets

You can configure infolets that aggregate key information for a specific area, for example, personal profile.

Your users use infolets arranged in tabs in the Analytics section of the home page.

If your users don't find infolet pages on the home page, you can enable them by using the Home Configuration tab in the Structure work area. Let's look at some tasks you can do to configure infolets.

- Create infolets.
- Add content to infolets. For example, you can add a task flow or a performance tile report, and 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, a report is an analysis, not an Analytics Publisher report.

- Edit infolets. For example, edit infolet content and add, change, or remove link to detailed report.
- Delete infolets.

But before you start creating and editing infolets, here are a few things to keep in mind:

- You can add analyses from the catalog to an infolet, but not Analytics Publisher reports, or other objects like filters or prompts.
- To create or edit infolets, you must first either create and activate a sandbox, or activate an existing one. But make sure the sandbox has the Page Composer tool selected. If you want to make changes in a context layer that isn't the default layer, Site, you must create a separate sandbox just to use Page Composer in it. You can then change the context layer from Site to the other layer. For example, to create or edit infolets for a user with a specific job role, you must select the **Job Role** context layer.
- You can validate your changes in the sandbox in preview mode before you publish it.

Related Topics

- Best Practices for Using Page Composer in Sandboxes
- How can I configure how infolets are displayed in Fusion Applications?

Create Infolets

Watch video

Use infolet pages to create infolets. You can create an infolet, add content to it, and link detailed reports. For some product-specific infolet pages, you can't create infolets.

1. Activate a sandbox that has the Page Composer tool in it.
2. Open an infolet page.
3. Click your user image or name in the global header, and on the **Settings and Actions** menu, select **Edit Pages**.
4. Click the **Infolet Repository** icon, and select **Create Infolet**.

   > **Note:** If you can't find the **Create Infolet** menu item, it either means you can't create infolets for this page, or the theme may be using a global text and icon color that's too light, impacting the visibility of the UI element.

5. Enter a title for the infolet and set its views.
   - Specify the dimensions for the front view.
   - Enable or disable the back view.
   - Enable or disable the expanded view, and specify its dimensions. The dimensions of the front and the back views must be the same, but the expanded view must be bigger because it displays more details.
6. Click **Save and Close**. You can now add content to infolets.

Add Content to Infolets

You can add content to the infolet's front, back, or expanded view. You can't add a business intelligence dashboard to an infolet because a dashboard report is generally bigger than an infolet.

1. Select the infolet's view that you want to add content to.

   > **Note:** To open infolet's back view, click the **Back View** icon on the bottom right corner of the infolet. And to open expanded view, click the **Expanded View** icon on the bottom right corner of the infolet's back view.

2. Click **Add Content** on the infolet. You can find this button on any of the infolet's views, but only if you don't have any existing content in that view.
3. Search and 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. You can now add detailed reports to the infolet.

Link Detailed Reports to Infolets

To provide detailed information about a subject matter on an infolet, you can add a link to a detailed report. After you add the link, your 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.

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. Search and select a report, and click **Add** to add it to the infolet.
4. Close the Add Content dialog box.
5. Click **Done**. You can now publish the sandbox.

Publish the Sandbox

1. Preview the infolet's front view, drill down to the detailed report, and then preview the back and the expanded views.

2. After you made changes, click **Close** to leave Page Composer.
3. Test your changes and publish the sandbox to make the new infolet available to your users.

Related Topics

- Overview of Sandboxes
- Create and Activate Sandboxes
- Best Practices for Using Page Composer in Sandboxes
- Why can't I see application changes made in previously published sandboxes in my current sandbox?

Edit and Delete Infolets

On the infolet page, use the options available on each infolet to edit infolet content and delete infolets.

1. Activate a sandbox that has the Page Composer tool in it.
2. Open an infolet page.
3. Click your user image or name in the global header, and on the **Settings and Actions** menu, select **Edit Pages**.

> **Note:** After you're done making changes, click **Close** to leave Page Composer, test your changes, and publish the sandbox.

Edit Infolet Content

1. Click the **Actions** icon on the top right corner of the infolet, and select **Edit Content**.

> **Note:** To edit the infolet content of the back or expanded view, open the view and use the **Actions** icon on that view.

2. Click **Add Content** to replace the existing content of the infolet.
3. Search and 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.

Edit Title and Views

To edit title and views of an infolet, click the **Actions** icon on the top right corner of the infolet, and select **Edit Title and Views**.

Edit or Remove Detailed Report

You can edit or remove the detailed report.

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

Edit Infolet Visibility

You can show or hide an infolet on the infolet page. To edit the visibility settings of an infolet, do these steps:

1. Click the **Actions** icon on the top right corner of the infolet, and select **Edit Visibility**.
2. Select one of these options:
    - **Yes**: The infolet appears on the infolet page.
    - **No**: The infolet doesn't appear on the infolet page.
    - **EL Expression**: The evaluation of the EL expression decides whether the infolet appears on the infolet page.

Delete Infolets

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

> **CAUTION:** You can't easily retrieve an infolet once you delete it. But if you have accidentally deleted any infolet, don't worry. Contact My Oracle Support at https://support.oracle.com.

Related Topics

- Overview of Sandboxes
- Create and Activate Sandboxes

FAQs for Infolets

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.

Why can’t I find the Create Infolet menu item on the Infolets page?

If you can't find the **Create Infolet** menu item, it either means you can't create infolets for this page, or the theme may be using a global text and icon color that's too light, impacting the visibility of the UI element.

Related Topics

- Appearance Settings for Classic Themes

Configure Infotiles on a Page

You can use Page Composer to decide how your infotiles appear on your page. You can show, hide, or reorder them any way you like.

1. Activate a sandbox that has the Page Composer tool in it, and make sure the context layer you want is selected.
2. Go to the page with your infotiles.
3. Click your user image or name in the global header, and select **Edit Pages**.
4. Click the **Configure Layout** icon.
5. Make changes to your page.
    - Select the infotiles you want to show on your page, and deselect the ones you want to hide.
    - Hover over the infotile you want to reorder, and click **Move Up** or **Move Down**.
6. Click **OK**.
7. Click **Close**.
8. Test your changes and publish your sandbox.

Related Topics

- Overview of Sandboxes
- Publish Sandboxes

Pages Hosting Third-Party Applications

Create Pages for Hosting Third-Party Applications

Use the Page Integration tool to create pages for hosting third-party applications to address needs specific to your organization.

All the new pages are initially placed in a single category or group, and then you can move them to other groups using the Structure work area.

Create the First Page

Follow these steps to create the first page.

1. Activate a sandbox that has the Page Integration tool in it.
2. From the Navigator menu, select **Configuration** > **Page Integration**.
3. Click **New Page**.
4. On the Create Page page, enter a page name. Make sure the page name has no spaces or special characters.
5. In the **Category Name** field, enter a name for the group to place your page in.

> **Note:** Your new page is placed in the group that you create here. You can later move your new page from this group to any other group using the Structure work area. But the Page Integration tool shows the moved pages in the group that you create here.

6. Search and select an icon for the page.
7. From the list of application roles, select the role to whom you want to grant access to the page. This list contains only custom roles because you can't modify predefined roles.
8. 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 modified access, a partner can directly perform an action or display information to the specific user without any additional authentication.
9. Click **Save and Close**.

    The 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 in a group, then that page icon appears at the top level (not in any group) on the springboard. However, such page icons appear in their respective categories on the Navigator menu.

Create Subsequent Pages

After you created the first page, you can create the subsequent pages.

1. On the Page Integration Wizard: New Pages page, click **New Page**.
2. On the Create Page page, enter a page name. Make sure the page name has no spaces or special characters.
3. Search and select an icon for the page.
4. From the list of application roles, select the role to whom you want to grant access to the page.
5. 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.
6. Click **Save and Close**.

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

Related Topics

- Manage Pages Hosting Third-Party Applications
- Overview of Sandboxes
- Reorder Navigation Entries
- Create and Activate Sandboxes

Manage Pages Hosting Third-Party Applications

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.

Edit Page Settings

Follow these steps:

1. Activate a sandbox that has the Page Integration tool in it.
2. On the Page Integration Wizard: New Pages page, click the name link for the page that you want to edit.
3. On the Edit Page page, make the required changes.
4. 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.

Add 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. From the list of application roles, select the role to whom you want to grant access to the page. This list contains only custom roles because you can't modify predefined roles.
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.

Edit 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.

Rename Categories

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

Navigate 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

- Overview of Sandboxes
- Create Pages for Hosting Third-Party Applications
- Configure Links for Page Entries and Quick Actions
- Change Page Layout

FAQs for Pages Hosting Third-Party Applications

Why can't my users find the page I created?

That's probably because your new pages are secure, and the security administrator hasn't assigned the page access role to these users.

Related Topics

- Create Pages for Hosting Third-Party Applications

5 User Interface Text

Tools for Changing Text

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

Use these tools to make text changes:

- Application Composer
- User Interface Text
- Page Composer
- Structure

Multiple factors influence the option you select to modify UI text. Here are a few factors:

- Your product family
- The extent and scope of your changes
- The components that you modify

This table lists the tools that you can use to modify UI text, navigation steps to use these tools, and the corresponding product families associated with these tools.

Tool or Task	Navigation	Product Family
Application Composer	Click **Navigator** > **Configuration** > **Application Composer**.	• Oracle Fusion Cloud Sales Automation • Oracle Fusion Cloud B2B Service • Oracle Fusion Cloud Supply Chain & Manufacturing • Oracle Fusion Cloud Project Management
User Interface Text	Click **Navigator** > **Configuration** > **User Interface Text**.	All applications
Page Composer	Click your user image or name in the global header, and select **Edit <Page Name> Pages** in the **Settings and Actions** menu. If the Edit <Page Name> Pages option isn't available in the **Settings and Actions** menu, then select **Edit Pages** instead.	All applications
Structure	Click **Navigator** > **Configuration** > **Structure**.	All applications

Text Modification Scenarios

This table includes possible scenarios for modifying user interface text. Compare your situation to the scenario in the table to determine the most appropriate tool for modifying 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 change affects multiple pages throughout your application. You can edit 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 change 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 change 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 change affects 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 change 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 specific object wherever the object appears	Targeted: A business object The change affects a specific component of a specific message in the message dictionary.	User Interface Text	Change the label of the opportunity business object, from "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".
Replace words or phrases that appear in the browser tab title on a specific page	Targeted: A page The change affects browser tab title on a specific page or page fragment.	User Interface Text	Change the browser tab title from "Audit Reports" to "Audit Summary".

No matter which tool you use to make changes, all application changes are written in a single override bundle. So, the latest application changes overwrite the previous ones.

> **Note:** If you replace text using plain text as an input value, it supersedes any changes that use the override bundle. For example, if you enter a direct string in Page Composer, then Page Composer writes these changes in a file (not string resource) containing page modifications. So, such changes overwrite the changes in the override bundle.

Related Topics

- Guidelines for Using Page Composer
- Examples of Configurations and Extensions, and the Tools to Use
- How do I modify text using the User Interface Text tool?

How do I modify text using the User Interface Text tool?

Use the **User Interface Text** tool to change text in various application components. You can modify and translate multiple strings at once, and also export them for offline modifications. To use the User Interface Text tool, make sure you have the Manage User Interface Text (FND_MANAGE_USER_INTERFACE_TEXT_PRIV) privilege.

1. In Sandboxes, create a sandbox or enter an existing one. Make sure the sandbox has the **User Interface Text**.
2. From the sandbox Tools menu select **User Interface Text**.
3. Enter the term you want to find and its replacement. You can select the types of application components you want to display in the search results.
    - User Interface Text: Field Name, page titles, help text (not links) in help windows
    - Global Menu Label Text: Menu labels on the navigator and home page
    - Multipart Validation Messages: Error, validation, and warning messages defined in message dictionary
    - Enterprise Scheduler Text: Names and descriptions of scheduled processes
4. Click **Search**.

5. Preview the search results in separate tabs, organized by the types of application components. Before replacing the strings, make sure you review the changes in each tab of the search results.
   a. You can do these tasks across all the tabs.
      - Edit the replacement text.
      - Exclude the rows containing strings you don't want to change.
   b. You can also do additional tasks for some tabs.

      User Interface Text

      - Click **Export All** to export all the the search results and view them in a .csv file.
      - From the **Exclude** drop-down list, you can also select one of these options:
        - None: Doesn't exclude any search result.
        - All: Excludes all results.
        - None (Page): Doesn't exclude any result on the page.
        - All (Page): Excludes all results on the page.
      - Click **Query By Example** and query your search results in the fields above the table column headers headers and find specific strings. To revert to previous search results, delete any text you've entered in these fields.

      Messages

      - In the Other Matches column, click to view matching text found in the message details.
      - You can expand each row to view additional information.
6. When you're satisfied with the search results and your adjustments, click **Replace Strings**. Otherwise, click **New Search** to start afresh.
7. Test your changes and publish the sandbox.

Sample Search Values for User Interface Text Modification

You search and replace text using the User Interface Text tool. You can enter singular and plural forms of whole words or phrases. You can either do case-sensitive searches or search for an exact match of your search text value.

This table lists the sample values that you can use while entering search text.

Note: You can't perform partial word searches, nor can you use wildcard characters as part of the search text.

Search Text	Expected Match	Match?	Reason

`Flex`	`flex`
	
	`Flex`
		Yes	The application searches for any occurrence of your search string, no matter what its position is, in the strings it searches.
			
			Unless you select **Match Case**, all matches are considered exact.

Search Text	Expected Match	Match?	Reason
`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 **Find** 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** to search for an exact match of your search text value.

`^Flex$`	`flex`
	
	`Flex`
		Yes	Use `^string$` in the **Find** 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.
			
			Use just `^string` in the **Find** field to say this phrase must appear at the beginning of a field.

`flex credits`	`Flex Credits Configuration`
	
	`Allow rollover of flex credits`
	
	`Flex Credits`
	
	`Manage Flex Credits Configuration`
		Yes	The application searches for any occurrence of your search string, no matter what its position is, in the strings it searches.
			
			Unless you select **Match Case**, all matches are considered exact.

`flex credits`	`flex credit shell plan`
	
	`Allow rollover of flex credit for...`
		No	The application searches for any occurrence of your search string, no matter what its position is, in the strings it searches.

`credits configuration$`	`Flex Credits Configuration`
	
	`Manage Flex Credits Configuration`
		Yes	Use `string$` in the **Find** field to say this phrase must appear at the end of a field.

Search Text	Expected Match	Match?	Reason
			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.

Overview of Translating Modified Text

If you install and use multiple languages in your application and modify the text, enter translations of the modified text for all languages. You can enter translations for existing and newly added strings manually. You can also translate strings offline.

You can use many configuration tools to update or add strings. For example, you can use lookups to add translations, or you can use the User Interface Text tool to update all strings and enter their translations.

Related Topics

- Translation of Flexfield and Value Set Configurations
- Translate Existing Strings Added Using Configuration Tools
- Translate New Strings Added Using Configuration Tools

Translate Existing Strings Added Using Configuration Tools

In this example, you translate existing strings manually.

This 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 modify?	Page

Decisions to Consider	In this Example
What's the replacement text that you want to replace the existing text with?	Work area

Enter Text Translations for Existing Strings

1. Install English language in your application.
2. Create and activate Sandbox1.
3. Click **Navigator** > **Configuration** > **User Interface Text**.
4. Click the Search and Replace tab, which is the default tab you see.
5. In the **Find** field, enter the text, "page".
6. In the **Replace** field, enter the text, "work area".
7. Click **Search**.
8. Preview and adjust changes in your search results.
9. Click **Replace Strings**.
10. Test and verify all messages and pages affected by the text changes. Be sure to test across all applications.
11. Publish Sandbox1.

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

Related Topics

- Create and Activate Sandboxes
- Overview of Translating Modified Text
- Translate New Strings Added Using Configuration Tools

Translate New Strings Added Using Configuration Tools

In this example, you translate new strings that were added using configuration tools. Remember, while creating strings using configuration tools, such as the Structure page, always use the same language, that is, your base language.

This 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

Decisions to Consider	In this Example
What's the French replacement text that you want to replace the newly created English text with?	Ordinateur

Enter Text Translations for Newly Added Strings

1. Install both English and French languages in your application.
2. Create and activate Sandbox2.
3. Click **Navigator** > **Configuration** > **User Interface Text**.
4. Click the Search and Replace tab, which is the default tab you see.
5. In the **Search** field, enter the newly created English string. You must search in English because the French equivalent has not yet been created.
6. In the **Replace** field, enter the French string as the replacement text.
7. Select French as the language you want to translate your new English string to.
8. Click **Search**.
9. Click **Replace Strings**.
10. 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: You can select any active language instead of French to translate your new English string.
11. Publish Sandbox2.
    Your replacement text for the newly added string is now available to all users.

Related Topics

- Create and Activate Sandboxes
- Overview of Translating Modified Text
- Translate Existing Strings Added Using Configuration Tools

Translate Strings Offline

You can translate your modified strings offline. To translate your strings offline, export them, add the necessary translations, and then import them back into the application.

Export Strings

1. Create and activate a sandbox.
2. Click **Navigator** > **Configuration** > **User Interface Text**.
3. Click the Export tab.
4. Select the source language for the text you want to export for translation.
5. Click **Export** and download the file with all the modified strings that have been published as well as those that are in the current sandbox.

Translate Strings

1. Extract the .zip file you downloaded, and open the .xlf file inside it using a text editor of your choice.
2. Locate the strings you want to edit. You can find these strings inside `<source>` and `</source>`.
3. Replace the `<target/>` tag following the `</source>` tag with the translation you need. The translated string should be inside `<target>` and `</target>`.

   Let's look at an example of an English to Spanish translation.

   ```
   <source>All Receipts</Source>
   <target>Todos los Recibos</target>
   ```

4. Save the file when you're done.
5. Compress the translated .xlf file into a .zip file.

Import Translated Strings

1. Click **Navigator** > **Configuration** > **User Interface Text**.
2. Click the Import tab.
3. Select the target language you want to import the translated strings to.
4. Click **Browse** and select the .zip file you want to import.
5. Click **Import**.

> **Note:** After importing your strings, test your text changes inside the sandbox. Publish the sandbox to apply your changes to the application.

Related Topics

- Create and Activate Sandboxes

Translate New Strings Offline From English to German

In this example, you translate new strings that were added using configuration tools, from English to German. Remember, while creating strings using configuration tools, always use the same language, that is, your base language.

This 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?	Sandbox3
What's your base language?	English
What's the language you want to translate your newly added English string to?	German

Decisions to Consider	In this Example
What's the newly created English text that you want to translate in German?	Network
What's the German replacement text that you want to replace the newly created English text with?	Netzwerk

Export Strings

1. Install both English and German languages in your application.
2. Create and activate Sandbox3.
3. Click **Navigator** > **Configuration** > **User Interface Text**
4. Click the Export tab.
5. Select **American English** as the source language for the text you want to export for translation.
6. Click **Export**.

Translate Strings

1. Extract the .zip file you downloaded. The .zip file name includes "en", which is the code for English Language.
2. Rename the .zip file to change the code from "en" to "de", which is the code for German language.
3. Open the .xlf file inside the .zip file using a text editor of your choice.
4. Locate the strings you want to edit. You can find these strings inside `<source>` and `</source>`.
5. Replace the `<target/>` tag following the `</source>` tag with the translation you need. The translated string should be inside `<target>` and `</target>`.

   Here is how it should look:

   ```
   <source>Network</Source>
   <target>Netzwerk</target>
   ```
6. Save the file when you're done.
7. Compress the translated .xlf file into a .zip file.

Import Translated Strings

1. Click **Navigator** > **Configuration** > **User Interface Text**.
2. Click the Import tab.
3. Select **German** as the target language you want to import the translated strings to.
4. Click **Browse**, and select the .zip file you want to import.
5. Click **Import**.

> **Note:** After importing your strings, test your text changes in Sandbox3. Publish Sandbox3 to apply your changes to the application.

Related Topics

- Create and Activate Sandboxes

FAQs for User Interface Text

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

It depends on types of text changes in the sandbox. If you want to undo all text changes done in the user interface and global menu label, delete your sandbox before publishing it.

Note: However, you can't undo the text changes done in messages and enterprise scheduler.

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

No, but you can use the Manage Configurations dialog box to download all your configurations to a .zip file.

To export all your configurations:

1. Click your user image or name in the global header, and on the **Settings and Actions** menu, select **Manage Configurations**.
2. On the Manage Configurations dialog box, you can search for the files ending with **.xlf** or **.xml**.
3. Click **Download Configuration for All Layers**.

These files list all text changes done in your application using browser-based tools, such as Application Composer, Page Composer, and User Interface Text. You can use the contents of these files for diagnosis and troubleshooting purposes. However, these files can't be uploaded to the sandbox.

Related Topics

- View and Diagnose Application Changes

Why didn't text in my BI reports and SOA processes change when I used the User Interface Text tool for bulk updates?

The bulk updates that you do using the User Interface Text tool affect only the text that appears on application pages, in messages, in global menu label, and in enterprise scheduler.

Why didn't text in my lookup change when I used the User Interface Text tool to modify multiple occurrences of text?

The updates you do using the User Interface Text tool affect only the text that appears on application pages.

Within a given lookup type, each entry correlates a lookup code to a meaning. The meaning is the text that actually appears on the application page. To edit the meaning for a predefined lookup, do these steps:

1. In the Setup and Maintenance work area, go to the **Manage Standard Lookup** task.
2. Search for the lookup you want to edit the meaning for.
3. Edit the lookup meaning.
4. Click **Save and Close**.

To edit the meaning for a predefined lookup in a specific language, use the **Translation Editor** icon on the table toolbar or in the form.

Related Topics

- Overview of Lookups
- How can I edit lookups?
- How can I enter or edit translated text in Fusion Applications?

Why is the Restore button disabled on the Migration page?

It might be because of one or more of these reasons.

- A sandbox is currently being published in your environment.
- Your metadata is being imported from the source to the target environment.
- Flexfields are being deployed to a sandbox.
- The environment is upgraded after your previous migration.

6 Themes

What themes can I use in Fusion Applications?

There are three options:

- Apply the Default theme, which is a Redwood theme and has News feed as the home page layout.
- Create a Redwood theme either from scratch or using the Default theme.
- Use your previously saved classic themes, which you can edit using the classic themes editor. You can’t create new classic themes, and predefined classic themes aren't available anymore.

How do I create Redwood themes for Fusion Applications?

1. In Sandboxes, create a sandbox or enter an existing one. Make sure the sandbox has the Appearance tool. Don't include spaces or special characters in the sandbox name.
2. From the sandbox Tools menu, select **Appearance**.
3. Create your own theme from scratch. Or change settings in the Default theme and save that configuration as a new theme.

   Tip:
   - For cover image, the recommended image size is 2600x290 px.
   - For images, the files you upload can be of PNG, JPEG, and GIF types, and can have spaces in their names.
4. Apply the theme and publish the sandbox.

How do I configure classic themes for Fusion Applications?

1. In Sandboxes, create a sandbox or enter an existing one. Make sure the sandbox has the Appearance tool. Don't include spaces or special characters in the sandbox name.
2. From the sandbox Tools menu, select **Appearance**.
3. From the banner on the Appearance Themes page, open the classic themes editor.

   If you don't see the banner, that means you don't have any classic themes.

4. From the list of themes, select any classic theme you've previously saved and edit the appearance settings.

   Tip:

   - For heading color, choose a dark color because in some cases, lighter page and section headings won't be visible against the page background.
   - For cover image, the recommended image size is 2600x290 px.
   - For images, the files you upload can be of PNG, JPEG, and GIF types, and can have spaces in their names.

5. Apply the theme and publish the sandbox.

FAQs for Themes

Why does the application logo look different on some pages?

Changes made to the logo using tools like Page Composer or Page Template Composer supersede the changes made using the Appearance work area. So we recommend you only use the Appearance work area to make changes to the logo. But before that, make sure you undo any changes made to it using Page Composer or Page Template Composer.

7 Flexfields

Overview of Flexfields

A flexfield is a set of placeholder fields associated with business objects and placed on the application pages to contain additional data.

You can use flexfields to modify the business objects and meet enterprise data management requirements without changing the data model or performing any database programming. Using flexfields, you can capture different data on the same database table and modify the application features. For example, airline manufacturers may require specific attributes for orders that aren't predefined. Using a flexfield for the order business object, they can create and configure the required attribute.

When you first define your flexfields, you choose how many segments you want to use and what order you want them to appear in. You also choose how you want to validate each of your segments. The decisions you make affect how you define your value sets. When your user enters a value, your value sets validate that value. The value sets can be one of these types:

- List of valid values that aren't stored in any application table.
- Table-validated value sets that use a predefined list of values stored in an application table.
- Format only value sets that use any value as long as it conforms to your specified formatting rules. For example, the value must not exceed the maximum length you define for your value set.

You can assign a value set to a flexfield segment to define the valid values in that segment. While defining valid values in a segment, either reuse an existing value set or define a new value set. You can reuse a value set as often as you want across multiple flexfields or different segments within the same flexfield.

Flexfield Types

Here are the types of flexfields:

- Descriptive flexfields
- Extensible Flexfields
- Key Flexfields

Both extensible and descriptive flexfields provide an expansion space that you can use to configure additional attributes without additional programming. Descriptive flexfields are stored in additional columns on the same table as that of the object, whereas extensible flexfields are stored in a separate extension table.

Key flexfields consist of one or more segments, where each segment can have a meaning. You can use key flexfields to enter multipart values, such as a part number, a job code, or an account code. Users see the flexfields as fields or attributes on UI pages. To manage the flexfields, use any of these tasks in the Setup and Maintenance work area:

- **Manage Descriptive Flexfields**: To configure descriptive flexfield segments if you have enabled descriptive flexfields on a page.
- **Manage Extensible Flexfields**: To configure extensible flexfield segments if you have enabled extensible flexfields on a page.
- **Manage Key Flexfields**: To configure key flexfields, which is a configurable multipart key.

- **Manage Value Sets**: To create reusable value sets to validate values in flexfield segments.

> **Note:** You can manage value sets within the **Manage Descriptive Flexfields** or **Manage Extensible Flexfields** tasks.

Related Topics

- How can I access predefined flexfields?
- Overview of Descriptive Flexfields
- Overview of Extensible Flexfields
- Overview of Key Flexfields
- Overview of Value Sets

Overview of Flexfield Configuration

Configuring a flexfield involves identifying the need for enhancing a business object with user-defined attributes and then integrating the attributes into deployment. In the case of key flexfields, configuring the flexfield involves identifying value set assignments and determining segment structures.

Overall Process for Configuring User-Defined Attributes

Before using flexfields to create attributes, familiarize yourself with the context layers and the configuration life cycle of the application. You can add attributes to a business object using a flexfield, if developers have registered that object to a flexfield. 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 extensible flexfields changes to the mainline metadata, to display the attributes on the application pages and to make them available for integration with other tools such as Oracle Transactional Business Intelligence. But for descriptive flexfields, you must deploy your changes to a sandbox and test the changes before publishing your sandbox. To deploy your descriptive flexfield to a sandbox:
    a. Create or activate a sandbox, and select **Flexfields** from the list of tools.
    b. In the Setup and Maintenance work area, go to the **Manage Descriptive Flexfields** task.
    c. Open the flexfield you want to edit, and make your changes.
    d. On the **Actions** menu, select **Deploy Flexfield to Sandbox**.

    e. Click **Yes** to continue with the deployment.
6. Perform the necessary steps to integrate the 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 User-Defined Keys

Using key flexfields, you can configure intelligent key codes containing 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 key flexfield changes to the mainline metadata to display it on the application pages and to make it available for integration with other tools such as Oracle Transactional Business Intelligence.

    > **Note:** You can't deploy your changes in value sets, key flexfields, and extensible flexfields to your sandbox. You must deploy them directly to the mainline environment. However, if you're making changes to any descriptive flexfields, you must deploy your flexfield to the sandbox and test your changes before publishing your sandbox.

7. Perform the necessary steps to integrate the flexfield into the technology stack.

Related Topics

- Overview of Extensible Flexfields
- Considerations for Managing Flexfields
- Overview of Key Flexfields
- Overview of Context Layers
- Configuration Life Cycle

Flexfield Components

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.

> **Note:** You can translate all these flexfield components to the preferred languages without changing the language session of the application. To specify the translations in all the enabled language rows, use the Translation Editor option on the respective edit pages. Once the updates are made, users can view the translated text for the specific flexfield components.

Related Topics

- How Flexfields Work with Oracle Application Cloud Architecture
- Overview of Value Sets
- How can I enter or edit translated text in Fusion Applications?

Flexfields at Runtime

Business objects have an associated descriptive or extensible flexfield. Using these, you can create 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 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.

You can locate flexfields on a page using the **Settings and Actions** menu. Click your user image or name in the global header to open this menu, and select **Highlight Flexfields** to locate flexfields on a page. The page renders in a special mode, displaying the location of flexfields, if any, on the page. Do the following:

- Hover over the **Basic Info** 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:** You can't create attributes on all flexfields. 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 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 the Application Composer tool 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.

Related Topics

- Modify Flexfields Using Page Composer

Modify Flexfields Using Page Composer

In Page Composer, you get the Design view by default to edit pages. But on some pages, you must use the Source view. Set the **Source View for Page Composer Enabled** (FND_PAGE_COMPOSER_SOURCE_VIEW) profile option to **Yes** so that the **View** menu is available for you to switch between the **Design** view and the **Source** view. You can find the profile option in the **Manage Administrator Profile Values** task in the Setup and Maintenance work area.

Modify Extensible Flexfields

Here are the steps to modify extensible flexfields.

1. Activate a sandbox that has the Page Composer tool in it. Make sure the context layer of your sandbox is supported by the page you want to edit. Otherwise, you won't be able to edit the page.
2. Open the page with the flexfield you want to modify.
3. Click your user image or name in the global header and select **Edit Pages** from **Settings and Actions**.
4. From the **View** menu, select **Source**.
5. Expand the Source View pane at the bottom of the page.
6. In the Source View pane look for the component that's bound to an `EffContextsPageContainer` task flow.
7. Select the component and click the **Show the properties of EffContextsPageContainer** .
8. Edit the component properties and click **Ok** to save your changes.
9. Click **Close** in the Page Template Composer toolbar to save your changes.
10. Test your changes and publish the sandbox. See *Publish Sandboxes*.

Modify Descriptive Flexfields

Here are the steps to modify descriptive flexfields.

1. Activate a sandbox that has the Page Composer tool in it. Make sure the context layer of your sandbox is supported by the page you want to edit. Otherwise, you won't be able to edit the page.
2. Open the page with the flexfield you want to modify.
3. Click your user image or name in the global header and select **Edit Pages** from **Settings and Actions**.
4. From the **View** menu, select **Source**.
5. Expand the Source View pane at the bottom of the page.
6. In the Source View pane look for the component that's bound to a `descriptiveFlexfield` task flow.
7. Select the component and click the **Show the properties of descriptiveFlexfield**.
8. To edit the properties, do these tasks:
    a. Select or enter a value for each property.
    b. Optionally, you can also click the **Edit** icon next to the property.
        i. In the Expression Editor dialog box, select **Type a value or expression** .
        ii. You can enter any of these values:
            - True: The property is enabled for the UI element.
            - False: The property isn't enabled for the UI element.
            - EL expression: The evaluation of the EL expression decides whether the property will be enabled or disabled for the UI element.

iii. To make sure the value or expression you entered is correct, click **Test**.
iv. Click **Ok**.

c. On the Component Properties: descriptiveFlexfield dialog box, click **Ok** to save your changes.
d. Click **Close** in the Page Composer toolbar to save your changes.

9. Test your changes and publish the sandbox. See *Publish Sandboxes*.

Here's the list of descriptive flexfields properties that you can edit.

Property	Description
Read-only	Indicates whether the flexfield is rendered as read-only.
Auto Submit	Indicates whether the component submits automatically when the user changes the segment's value.
Rendered	Indicates whether the flexfield is rendered on the page. **Note:** If a value for this property is explicitly set within the descriptive flexfields metadata, it's shown as read-only. Users can't override the metadata settings using Page Composer.
Required	Indicates whether users must enter data for the component, before saving the page. **Note:** If a value for this property is explicitly set within the descriptive flexfields metadata, it's shown as read-only. Users can't override the metadata settings using Page Composer.
Inline Style	Specifies the inline style of the component. You can use this property to set individual properties, such as background color, font style, or padding.
Style Class	Specifies the style class of the flexfield. You can use this property to group a set of inline styles.
Mode	Indicates the UI mode of the descriptive flexfield component. You can use this property to render all the segments or just some of them. You can select any one of these values: • global: Renders only the global segments of the flexfield. You can use this value for a descriptive flexfield inside a table column to generate sub-columns for the global segments. • contextSensitive: Renders only the context-sensitive segments of the flexfield (including the context segment). You can use this value for a descriptive flexfield inside a table detail region to render the context values. **Note:** If no value is selected, all descriptive flexfield segments are rendered by default.
Auto Submit	Indicates whether the component submits automatically when the user changes the segment's value.
ShortDesc	Specifies a text that appears when a user hovers over an element.
Label	Specifies the display text for the component. For example, the field prompt or the single prompt for a group of checkboxes.
Columns	Specifies the width of the control in terms of the default font size of the browser.

Note: Flexfield segments can't be edited using their individual component elements. They can only be edited from the properties panel of the descriptiveFlexfield element they belong to. To find these components, open the properties panel of the descriptive flexfield, switch to the relevant tab, and search using the unique identifying information. For example, to find a context sensitive segment, open the properties panel for the flexfield and go to the Flexfield Context Segments tab. Search for the segment using its Context Value and Segment Code.

Modify Key Flexfields

Here are the steps to modify extensible flexfields.

1. Activate a sandbox that has the Page Composer tool in it. Make sure the context layer of your sandbox is supported by the page you want to edit. Otherwise, you won't be able to edit the page.
2. Open the page with the flexfield you want to modify.
3. Click your user image or name in the global header and select **Edit Pages** from **Settings and Actions**.
4. From the **View** menu, select **Source**.
5. Expand the Source View pane at the bottom of the page.
6. In the Source View pane look for the component that's bound to a `KffComponent` task flow.
7. Select the component and click the **Show the properties of KffComponent**.
8. To edit the properties, do these tasks:
    a. To enable the property for the UI element, select the checkbox next to the property.
    b. Optionally, you can also click the drop-down button next to the property and click **EL Expression**.
        i. In the Expression Editor dialog box, enter a value or expression:
            - True: The property is enabled for the UI element.
            - False: The property isn't enabled for the UI element.
            - EL expression: The evaluation of the EL expression decides whether the property will be enabled or disabled for the UI element.
        ii. To make sure the value or expression you entered is correct, click **Test**.
        iii. Click **Ok**.
9. On the Component Properties: KffComponent dialog box, click **Ok** to save your changes.
10. Click **Close** in the Page Composer toolbar to save your changes.
11. Test your changes and publish the sandbox. See *Publish Sandboxes*.

Here's the list of key flexfield properties that you can edit.

Property	Description
Required	Indicates whether user must enter data for the component, before saving the page. **Note:** If a value for this property is explicitly set within the key flexfields metadata, it's shown as read-only. Users can't override the metadata settings using Page Composer.
Read-only	Indicates whether the flexfield is rendered as read-only.
Rendered	Indicates whether the flexfield is rendered on the page. **Note:** If a value for this property is explicitly set within the key flexfields metadata, it's shown as read-only. Users can't override the metadata settings using Page Composer.

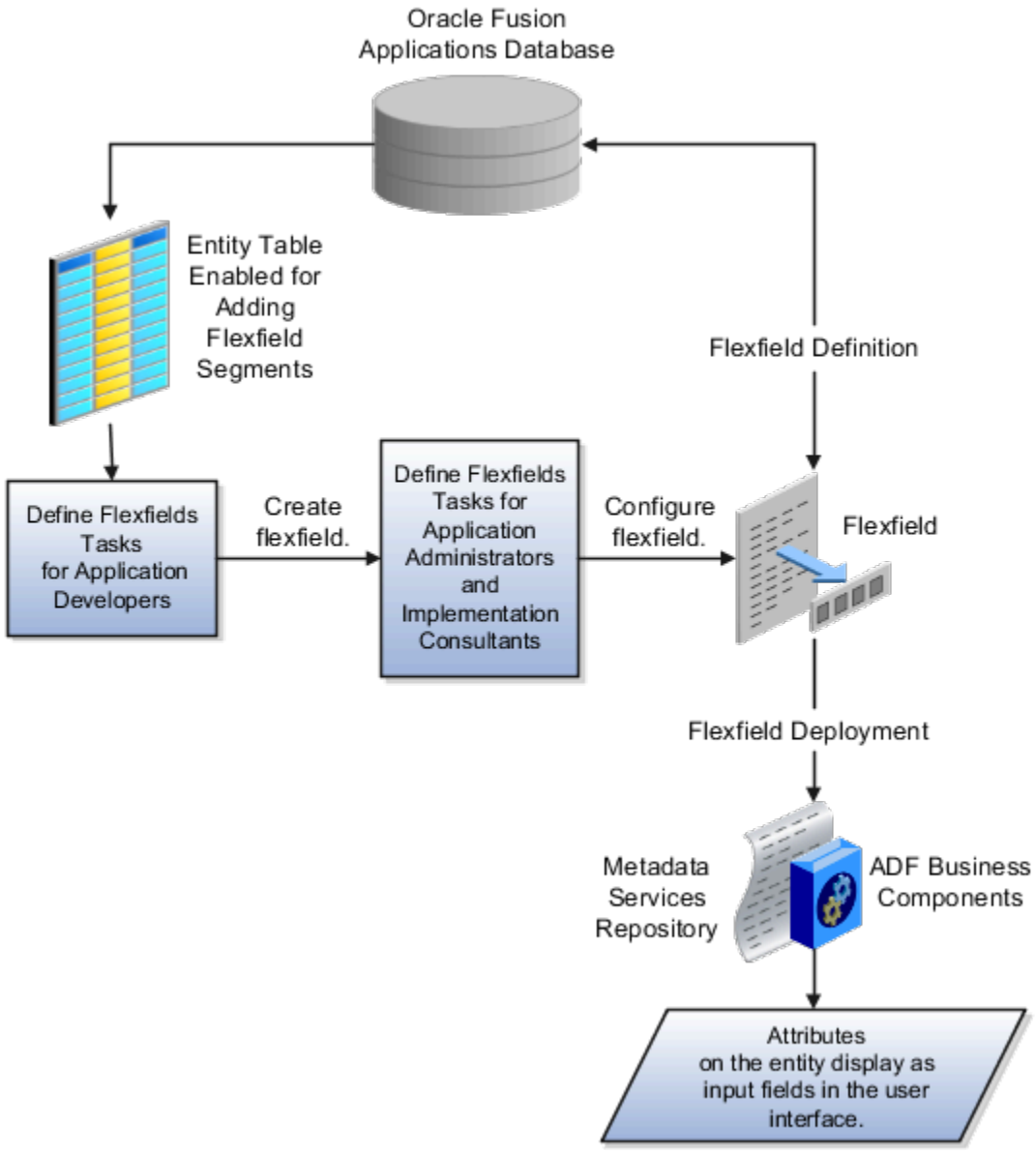
Property	Description
Auto Submit	Indicates whether the component submits automatically when the user changes the segment's value.
Visible	Indicates whether the flexfield is visible on the page.
Disabled	Indicates whether the flexfield is disabled on the page.
Immediate	Indicates whether the flexfield value is converted and validated immediately in different Oracle ADF life cycle phases.
Value	Indicates the value of the flexfield pointing to an iterator object. If you enter an EL expression, it should point to an iterator object.
Accessor	Indicates the name of the accessor between the product view object and the flexfield view object.
Label	Indicates the display text for the component. For example, the field prompt or the single prompt for a group of checkboxes.
Default Combination	Specifies the default value of the flexfield.
Help Topic Id	Specifies the help ID used to find a topic in the help content.
Inline Style	Specifies the inline style of the component. You can use this property to set individual properties, such as background color, font style, or padding.
Style Class	Specifies the style class of the flexfield. You can use this property to group a set of inline styles.

Related Topics

- Flexfields at Runtime
- Set Profile Option Values

How Flexfields Work with Oracle Application Cloud Architecture

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.



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 Fusion Cloud Applications architecture work together:

- Integration
- Deployment
- Import and export
- Runtime
- Patching

Integration

The attributes that you add by configuring flexfields are available throughout the Oracle 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 Transactional 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 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.

Runtime

The latest definitions of a flexfield reflect on the user interface 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 configurations for a flexfield using the Application Composer tool, the same flexfield segments can appear differently on different pages.

Patching

Flexfield configurations are preserved during patching and upgrading.

Related Topics

- How Flexfield Deployment Status is Calculated

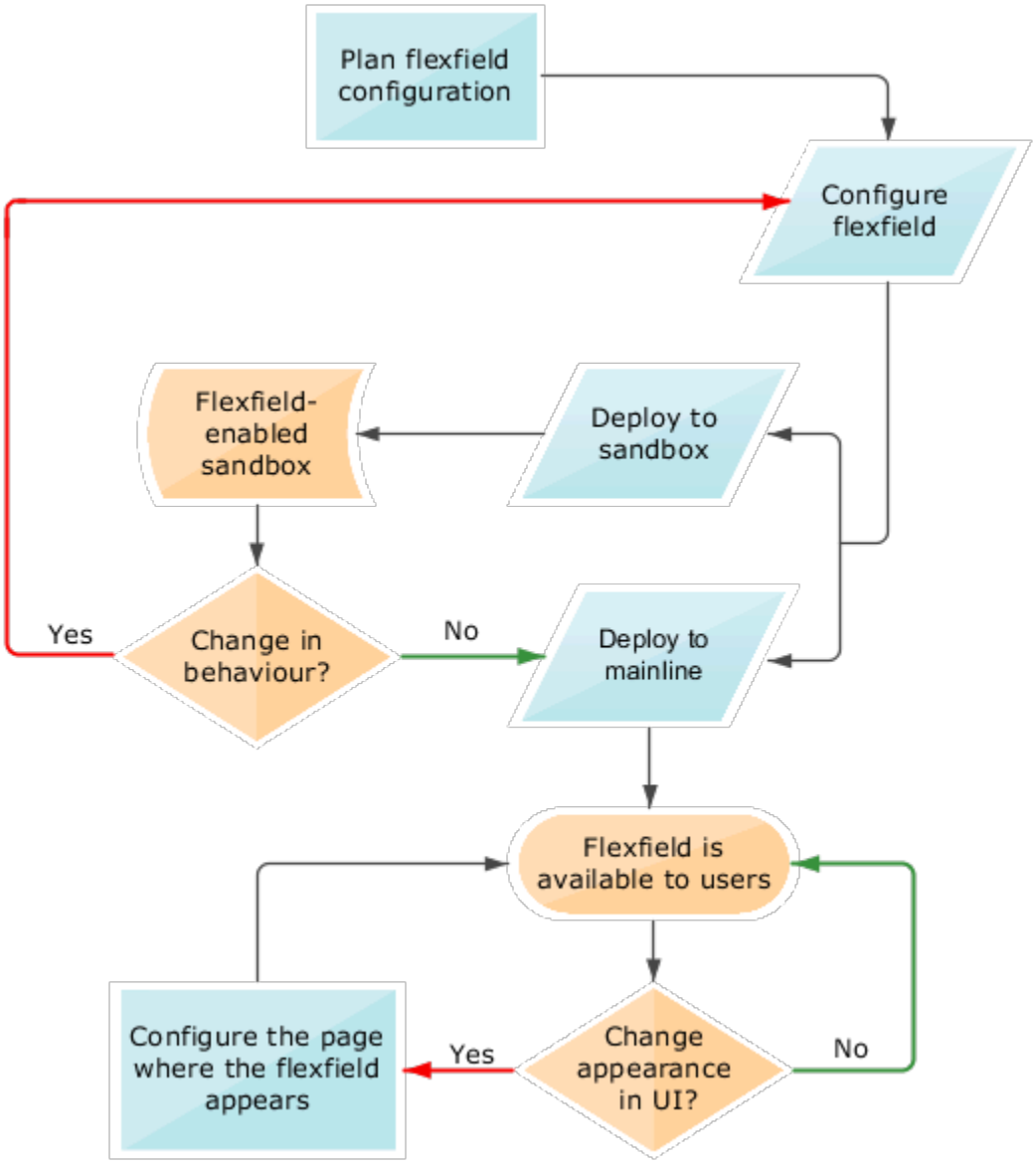
Flexfield Management

Considerations for Managing Flexfields

Managing flexfields involves 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 modify the UI page to change how the flexfield segments appear to users on that page.

The following 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:

- 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 page and troubleshooting

Planning Flexfields

Before you begin planning flexfields, determine what type is appropriate to your needs, and which business objects are available for modifying 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 determine and configure how the flexfield appears on the page. See Oracle Fusion Cloud Applications 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:

- Use the manage flexfield tasks in the Setup and Maintenance work area.
- Use the Highlight Flexfields command in the Administration menu while viewing a 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 next to the segment. 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 can't 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 Transactional 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 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.

Note: You can't deploy your changes in value sets, key flexfields, and extensible flexfields to your sandbox. You must deploy them directly to the mainline environment. However, if you're making changes to any descriptive flexfields, you must deploy your flexfield to the sandbox and test your changes before publishing your sandbox.

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 modify 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 user-defined global attributes so that they're interspersed with the core attributes in the same parent layout. You can modify 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. Click your user image or name in the global header to open the **Settings and Actions** menu, and select **Set Preferences**. Then change the text to the translated text for that locale.

Identifying Flexfields on a Page

The **Highlight Flexfields** command in the Administration menu of the Setup and Maintenance work area identifies the location of flexfields on the 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 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.

Related Topics

- Modify Flexfields Using Page Composer
- Overview of Value Sets
- Considerations for Managing Descriptive Flexfields
- Considerations for Managing Extensible Flexfields
- Considerations for Managing Key Flexfields

Flexfield Segment Properties

Independent of the value set assigned to a segment, segments may have properties that affect how they're 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.

Property	Description
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.
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 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's lesser than that of the high value segment.
- Non-range validated segments can exist between a range validated pair, but range validated pairs can't 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 must match a specified format or be restricted to a list of values. You can use a value set or a Groovy validator 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.

You can also use Groovy validation to set additional restrictions or requirements for what values are allowed for certain attributes of business objects. This is useful when you need to use the same value set to validate multiple segments, but the exact validation requirement changes with each case. These validators can be defined at the global segment level, or at the context level, based on your business needs. They have a validator code, validation expression, error message, and description. After adding a new validator, click the **Groovy Expression Builder** icon to open the expression builder window where you define your validation expression. Groovy validation is done when a user tries to save their values to an attribute that has a Groovy validator. If the value for this attribute fails validation against the Groovy expression, the text defined in the **Error Message** column is displayed as an error message.

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.

Related Topics

- Considerations for Managing Extensible Flexfields

Naming Conventions for Flexfield APIs

Application Programming Interface (API) name is a name for your flexfield component, which isn't exposed to users. These names are used to identify flexfield components in various integration points, including web services, rules, and business intelligence.

How to Frame API Names

You must use only alphanumeric characters for API names. For example, enter a name with the characters A-Z, a-z, or 0-9, with a non-numeric leading character. Don't use spaces, underscores, multi-byte characters, or leading numeric characters in your API names.

CAUTION: Do not change API names after they're created. Doing so could break integration points.

Words You Can't Use for API Names

You can't use certain words for API names when configuring flexfields because they're reserved words in ADF, SQL, PL/SQL, Java, Groovy, and so on.

This table lists some of the reserved words that can't be used as API names. This list isn't exhaustive.

Letter	Reserved Words
A	ABORT, ABSTRACT, ACCEPT, ACCESS, ACTIONENABLED, ADD, ALL, ALLROWSINRANGE, ALTER, AND, ANY, ARRAY, ARRAYLEN, AS, ASC, ASSERT, ASSIGN, AT, ATTRIBUTEDEF, ATTRIBUTEDEFS, ATTRIBUTEVALUE, ATTRIBUTEVALUES, AUDIT, AUTHORIZATION, AVG
B	BEGIN, BETWEEN, BINDINGS, BODY, BOOLEAN, BREAK, BY, BYTE, BYVALUE
C	CASCADE, CASE, CAST, CATCH, CATEGORY, CHAR, CHECK, CHILDREN, CLASS, CLONE, CLOSE, CLUSTER, CLUSTERS, COALESCE, COLAUTH, COLUMN, COLUMNS, COMMENT, COMMIT, COMPRESS, CONNECT, CONST, CONSTANT, CONSTRIANT, CONTAINS, CONTINUE, COUNT, CRASH, CREATE, CURRENT, CURRENTROW, CURRVAL, CURSOR
D	DATABASE, DATACONTROL, DATE, DBA, DEBUGOFF, DEBUGON, DECIMAL, DECLARE, DEFAULT, DEFINITION, DELAY, DELETE, DELTA, DESC, DESCRIPTION, DIGITS, DISPLAY, DISPLAYDATA, DISPLAYHINT, DISPLAYHINTS, DISPOSE, DISTINCT, DIV, DO, DOUBLE, DROP
E	ELSE, ELSIF, EMPTY, ENABLED, ENABLEDSTRING, END, ENTRY, EQ, EQUALS, ERROR, ESTIMATEDROWCOUNT, EXCEPTION, EXCLUSIVE, EXISTS, EXIT, EXTENDS
F	FALSE, FETCH, FILE, FINAL, FINALIZE, FINALLY, FINDMODE, FLOAT, FOR, FORM, FROM, FULLNAME, FUNCTION, FUTURE
G	GE, GENERIC, GETCLASS, GOTO, GRANT, GROUP, GROUPBY, GT
H	HASHCODE, HAVING, HINTS
I	IDENTIFIED, IF, IMMEDIATE, IMPLEMENTS, IMPORT, IN, INCREMENT, INDEX, INDEXES, INDICATOR, INITIAL, INNER, INPUTVALUE, INSERT, INSTANCEOF, INT, INTEGER, INTERFACE, INTERSECT, INTO, IS, ITERATORBINDING
J	JAVA
K	KEY

Letter	Reserved Words
L	LABEL, LABELS, LABELSET, LE, LEVEL, LIKE, LIMITED, LOCK, LONG, LOOP, LT
M	MANDATORY, MAX, MAXEXTENTS, MIN, MINUS, MLSLABEL, MOD, MODE, MODIFY
N	NAME, NATIVE, NATURAL, NE, NEW, NEXTVAL, NOAUDIT, NOCOMPRESS, NOT, NOTFOUND, NOTIFY, NOTIFYALL, NOWAIT, NULL, NULLIF, NUMBER
O	OF, OFFLINE, ON, ONLINE, OPEN, OPERATIONENABLED, OPERATOR, OPTION, OR, ORDER, ORDERBY, OTHERS, OUT, OUTER, OVERLAPS
P	PACKAGE, PARTITION, PCTFREE, POSITIVE, PRAGMA, PRIMARY, PRIOR, PRIVATE, PRIVILEGES, PROCEDURE, PROTECTED, PUBLIC
R	RAISE, RANGE, RANGESET, RANGESIZE, RANGESTART, RAW, REAL, RECORD, REFERENCES, RELEASE, REMR, RENAME, RESOURCE, REST, RESULT, RETURN, REVERSE, REVOKE, ROLLBACK, ROOTNODEBINDING, ROW, ROWID, ROWLABEL, ROWNUM, ROWS, ROWTYPE, RUN
S	SAVEPOINT, SCHEMA, SELECT, SELECTEDVALUE, SEPARATE, SESSION, SET, SHARE, SHORT, SIZE, SMALLINT, SPACE, SQL, SQLBUF, SQLCODE, SQLERRM, START, STATEMENT, STATIC, STDDEV, SUBTYPE, SUCCESSFUL, SUM, SUPER, SWITCH, SYNCHRONIZED, SYNONYM, SYSDATE
T	TABAUTH, TABLE, TABLES, TASK, TERMINATE, THEN, THIS, THROW, THROWS, TO, TOOLTIP, TOSTRING, TRANSIENT, TRIGGER, TRUE, TRY, TYPE
U	UID, UNION, UNIQUE, UPDATE, UPDATEABLE, USE, USER
V	VALIDATE, VALUES, VAR, VARCHAR, VARCHAR2, VARIANCE, VIEW, VIEWS, VOID, VOLATILE
W	WAIT, WHEN, WHENEVER, WHERE, WHILE, WITH, WORK
X	XOR

Related Topics

- Flexfield Segment Properties
- Considerations for Managing Flexfields

How Flexfield Segments are Rendered on Pages

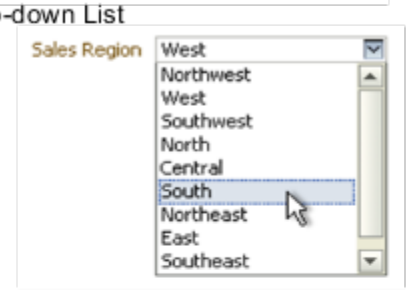
Flexfield segments appear on pages as attributes of business objects. When you create or edit flexfield segments, the value you enter for the display type determines how the segment appears on a page.

How Display Type Values Appear

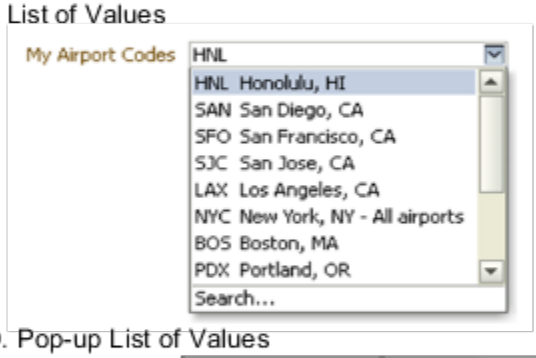
The series of figures (A to L) represent how the display types render on the UI. Each display type screenshot is assigned an alphabetic character that maps to the display type and its description in the table.

A. Check Box

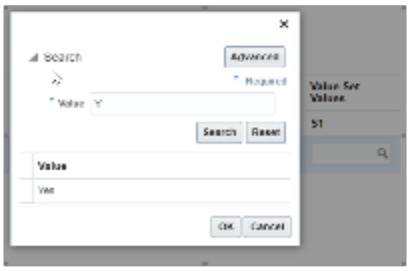
B. Drop-down List



C. List of Values



D. Pop-up List of Values

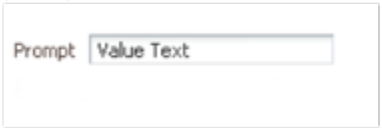


E. Search Enabled List of Values

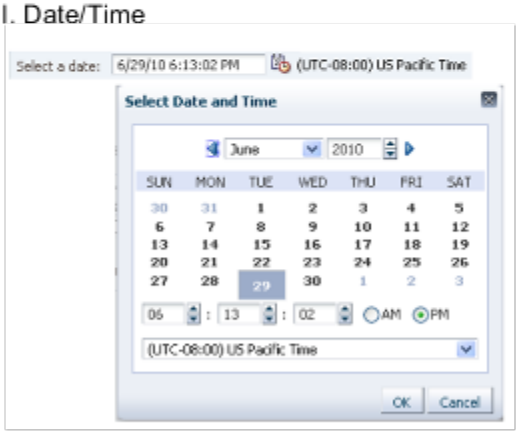
F. Radio Button Group

G. Text Area

H. Text Box

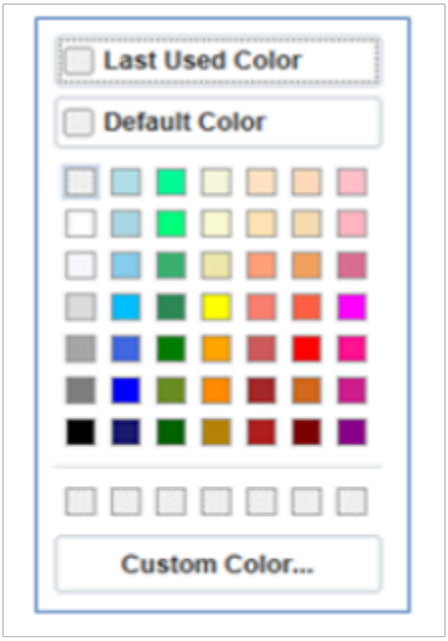


I. Date/Time

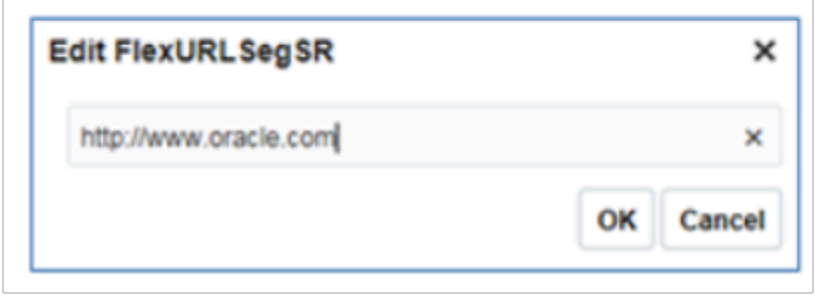


J. Rich Text Editor

K. Color



L. Static URL



The table describes each display type:

Figure Reference	Display Type	Description
A	Check Box	The field appears as a check box. If you select the check box, the checked value is used. Otherwise, the deselected value is used.
B	Drop-down List	The field appears as a list of values from which you can select. It supports a smaller number of values.
C	List of Values	The field appears as a list of values from which you can select.

Figure Reference	Display Type	Description
D	Pop-up List of Values	The field appears as a list of values from which you can select. It supports large volumes of values.
E	Search Enabled List of Values	The field appears as a text field with a **Search** icon. You can type a value in the text field. You can also click the Search icon that opens another window for searching.
F	Radio Button Group	The field appears as a set of radio buttons. You can click a button to select an option. When you select an option, previously selected option is automatically deselected.
G	Text Area	The field appears as a text area where you can enter multiple lines of text. The display width and height specify the visible width and number of lines in the text area.
H	Text Box	The field appears as a text field where you can enter a single line of text. The display width controls the width of the text box.
I	Date Time	The field enables you to enter a date if the data type is Date. You can enter a date and time if the data type is Date Time. You can select the date in a calendar. If the data type is Date Time, you can also specify the hour, minutes, seconds, AM or PM, and the time zone.
J	Rich Text Editor	The field appears as a text area where you 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. **Note:** This display type is available for extensible flexfields only.
K	Color	The field displays a color palette. You can select a color and assign it to the segment. You can select the display type only if the value set for the segment is set to ORA_FND_COLOR_#RRGGBB.
L	Static URL	The field appears as a text field where you can enter a fixed URL that opens the web page when clicked.

Figure Reference	Display Type	Description
		Note: The length of the URL must not exceed 255 characters.

Note: If you select the display type as Hidden, the field isn't displayed.

How Flexfields and Value Sets 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 user-defined attributes to be captured on the value set values 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.

Related Topics

- Overview of Flexfield Deployment
- How Flexfields Work with Oracle Application Cloud Architecture
- Validation Type Options for Value Sets

Default Segment Values

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

> **Note:** Default values for global, context, and context-sensitive segments are shown only when a descriptive flexfield row is created. Also, when a context segment value is changed on a UI that renders new context-sensitive segments, these segments display default values for both new and existing rows.

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 setting a default value or deriving a default value from a parameter, only those attributes designated by development as parameters are available for selection. Different combinations of making the segments read only or editable in combination with the default or derivation value or both, have different effects.

If your segment's default type is Groovy, you can set the Groovy expression you need using the expression builder. To open the expression builder, select **Groovy Expression** as your **Default Type** and click the **Groovy Expression Builder** icon. But you should know that Groovy defaulting doesn't support derivation when a dependent parameter changes. The expression is evaluated only at segment creation.

> **Note:** New usages of Groovy default values and validators in flexfields are deprecated because they won't be automatically upgraded in future releases. However, flexfields already using Groovy will continue to work. If you want to add Groovy expression to new flexfields, you must set the **Flexfield Groovy Usage Support Restricted** (ORA_FND_FLEX_GROOVY_RESTRICTED) profile option to **N**.

The following table maps these different combinations. Initial action corresponds to the row for the attribute value being created in the entity table. If the default value is read only, it can't 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 action	Action 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.
Groovy Expression	Yes	N/A	The default value of the segment is determined by evaluating the groovy expression.	Groovy expressions are evaluated only at segment creation. They're not evaluated when a dependent parameter is modified.
SQL	Yes	No	The default segment value is the value returned by executing SQL statement	N/A
SQL	Yes	Yes	The default segment value is the value returned by executing SQL statement	The changed parameter derivation value updates segment value

Note: For Descriptive Flexfield, only the `PARAMETER` bind variable is supported as the default type in a SQL statement in the specified format: `#{PARAMETER.<PARAMETER_NAME>}` Other bind variables, such as `SEGMENT`, `CONTEXT`, and `VALUESET` aren't supported in a SQL statement. Also, the default type in a SQL statement is evaluated only when you create a segment. They aren't evaluated when a dependent parameter is modified.

Related Topics

- How can I set a default value for a flexfield segment?
- Groovy Scripting Reference

Flexfield Usages

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 primary usage. Segments are based on the primary 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 use of an extensible flexfield context determines the scenarios or user interfaces in which the segments of a context appear to 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's associated only with the supplier usage appears only on the Supplier page and not on the Buyer page.

Value Sets

The use 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 manage flexfields tasks.

1. In the Setup and Maintenance work area, go to any of the following tasks:
    - Manage Descriptive Flexfields
    - Manage Extensible Flexfields
    - Manage Key Flexfields
2. On the page for the type of flexfield you're looking for, 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.
3. Click a flexfield to view its details.

For configuration that's not available through the manage flexfields tasks and the UI, ask your help desk to log a service request.

Related Topics

- How do I update existing setup data?

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 on 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 users. The flexfield segments that users see correspond to the flexfield definition last deployed successfully.

Note: Oracle Fusion Cloud Sales Automation and Oracle Fusion Cloud Fusion Service don't support flexfields.

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

Related Topics

- How Flexfield Deployment Status is Calculated

Why did my flexfield changes not appear in the 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 to view flexfield definition changes reflected in the application user interface page.

Related Topics

- How Flexfield Deployment Status is Calculated

How can I resolve synchronization issues between flexfields and business objects in Fusion Applications?

Use the **Synchronize Flexfields with Business Objects** scheduled process. This process automatically runs each time the application is patched and updated to synchronize the flexfield changes. You can also manually submit it from the **Scheduled Processes** work area.

1. In the **Scheduled Processes** work area, schedule a new process and select **Synchronize Flexfields with Business Objects**.
2. Set the parameters and submit the process.

   Here are the parameters to set, depending on what you need to do.

What You Need	Parameters to Select	What the Process Does
Resync all flexfields	Flexfield Type: **All Flexfields**	Deploys all flexfields marked as deployed.
Resync a single flexfield	○ Flexfield Type: **Descriptive Flexfields** or **Key Flexfields** ○ Flexfield Code **Note:** You can find the flexfield code using setup tasks like Manage Descriptive Flexfields or Manage Key Flexfields. For example, the flexfield code for Values Set Values is FND_VS_VALUES_B.	Resyncs the specific flexfield identified by the code you entered.
Force resync all flexfields	○ Flexfield Type:**All Flexfields** ○ Enforce Synchronization	Overrides the deployment time stamp and resyncs all flexfields.
Force resync a single flexfield	○ Flexfield Type: **Descriptive Flexfields** or **Key Flexfields** ○ Flexfield Code ○ Enforce Synchronization	Ignores any existing state or deployment history for a specific flexfield. It directly resyncs the business objects component of that flexfield with its latest definition, effectively resolving any discrepancies or outdated information.

After the process is completed, check for inconsistency errors in the log files.

Related Topics

- Submit Scheduled Processes and Process Sets
- View Status and Other Details for Scheduled Processes

Flexfield Deployment

Overview of Flexfield Deployment

Deployment generates or refreshes the Application Development Framework (ADF) business component objects that render flexfields on the UI. Deployment adds user-defined attributes to the Web Services Description Language (WSDL) schemas exposed by Oracle Fusion Cloud ADF services and used by SOA composites.

Flexfields are deployed for the first time when the application is provisioned. After you configure a flexfield, you must deploy it to make the changes 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 environment. You can deploy extensible flexfields as a background process.

After deployment, the user-defined 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 user-defined attributes. You must sign out and sign back in to Oracle Fusion Cloud Applications to see the changes you deployed in the production environment.

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. Check the deployment status of your flexfield after patching. The following table lists the different deployment statuses a flexfield can have.

Deployment Status	Meaning
Edited	The flexfield metadata definition hasn't been deployed yet. Updates of the metadata definition aren't applied in the production 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 production environment.
Deployed to Sandbox	The current metadata for the flexfield is deployed in ADF artifacts and available as a flexfield-enabled sandbox.

Deployment Status	Meaning
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 Fusion Cloud Applications 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 production environment 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 user-defined 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 *https://support.oracle.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`.

Related Topics

- How Flexfield Deployment Status is Calculated
- How Flexfields Work with Oracle Application Cloud Architecture
- Considerations for Deploying a Descriptive Flexfield to a Sandbox
- Why did my flexfield changes not appear in the UI?

How Flexfield Deployment Status is Calculated

Flexfield deployment status indicates how the flexfield metadata definition in the Oracle Fusion Cloud Applications database relates to the Application Development Framework (ADF) business components residing in 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, ensure that 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 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 application matches with 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's viewing the status.

If you deploy the flexfield successfully to the mainline metadata, the status is Deployed. The latest flexfield metadata definition in the 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 and 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 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 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 remains 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.

Related Topics

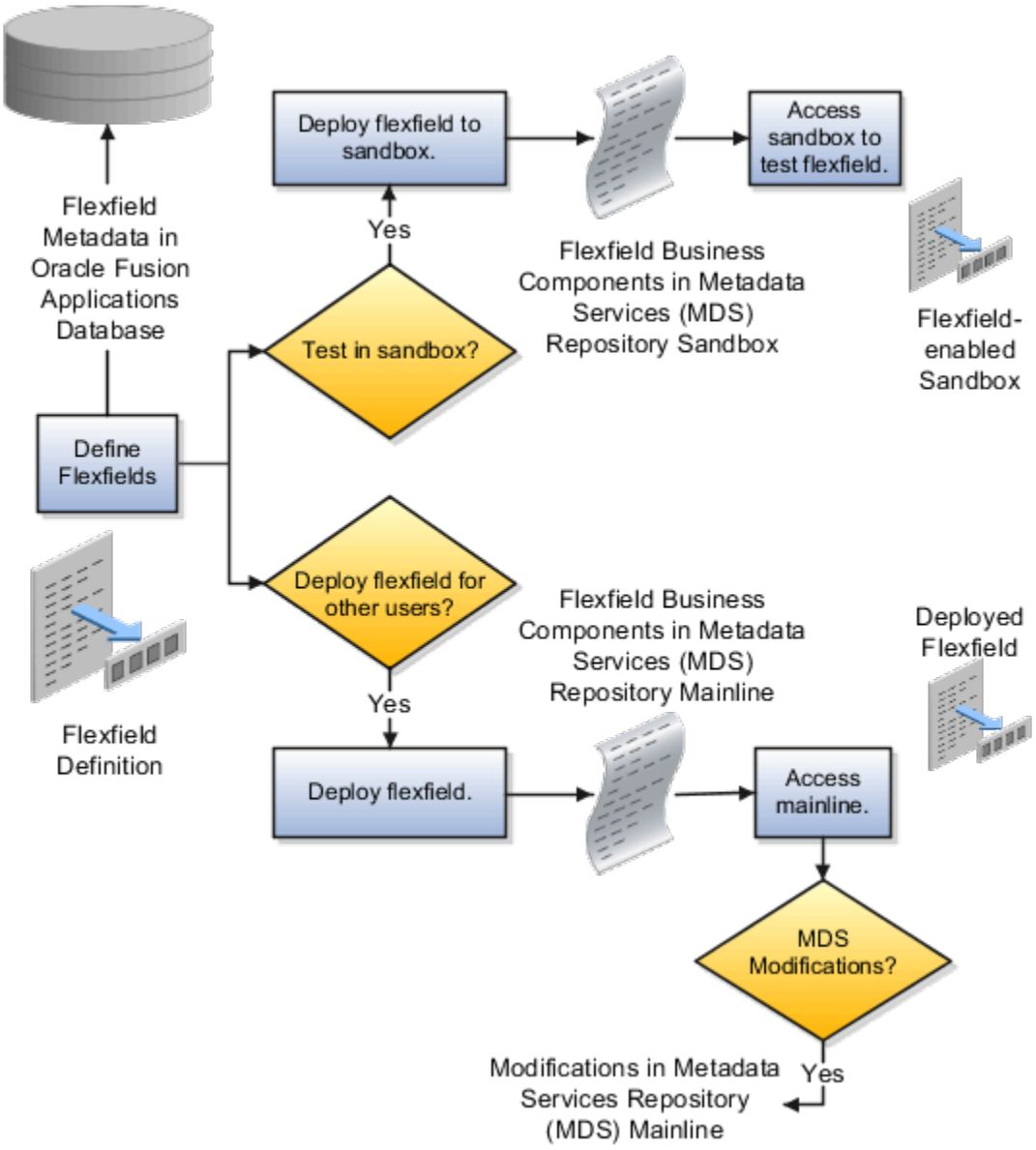
- How Flexfields Work with Oracle Application Cloud Architecture
- Considerations for Managing Extensible Flexfields

How Deploying a Flexfield-Enabled Sandbox Works with Mainline Metadata

When you deploy a flexfield to a sandbox, the flexfield definition corresponds to the flexfield metadata definition in the Oracle Fusion Cloud Applications database. When the flexfield is ready for use, it must be deployed to the mainline metadata.

A flexfield-enabled sandbox uses the following components.

- Flexfield metadata in the Fusion Applications database
- Flexfield business components in a sandbox Oracle Metadata Services (MDS) repository
- User interface modifications for the flexfield in 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 modify flexfield segment display properties using Page Composer in a flexfield sandbox as these changes are lost when you deploy your flexfield to the mainline metadata.

Mainline Metadata Services Repository Data

The Oracle Fusion Cloud 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

- How Flexfield Deployment Status is Calculated

Considerations for Deploying a Descriptive Flexfield to a Sandbox

You can deploy and test the behavior of a descriptive flexfield in the flexfield-enabled sandbox. Use the **Manage Descriptive Flexfields** task to change the flexfield definition, if required.

When you deploy a descriptive flexfield to a sandbox, the process reads the metadata about the segments from the database. It generates ADF Business Components artifacts for flexfields based on the definition, and stores the generated artifacts in the sandbox.

You can't deploy your changes in value sets, key flexfields, and extensible flexfields to your sandbox. You must deploy them directly to the mainline environment. However, if you're making changes to any descriptive flexfield, you must deploy your flexfield to the sandbox and test your changes before publishing your sandbox. After you deploy a descriptive flexfield to a sandbox, you must sign out and sign back in to the application to see your changes.

Note: You can deploy a descriptive flexfield to a sandbox using the **Manage Descriptive Flexfields** task only.

Value Sets

Overview of Value Sets

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.

A 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 open the Manage Value Sets page, use the Manage Value Sets task. You can also use the Manage Descriptive Flexfields and Manage Extensible Flexfields tasks for configuring a segment, including its value set. To open 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 users enter data instead of 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. If required, you may add table validated value sets to the list of available value sets available for configuration.

For a dependent value set, you must assign an existing independent value set to it. You can’t assign a value set whose validation type is Dependent.

> **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 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. Disabling security for individual usages of the same value set isn't possible.
- 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.

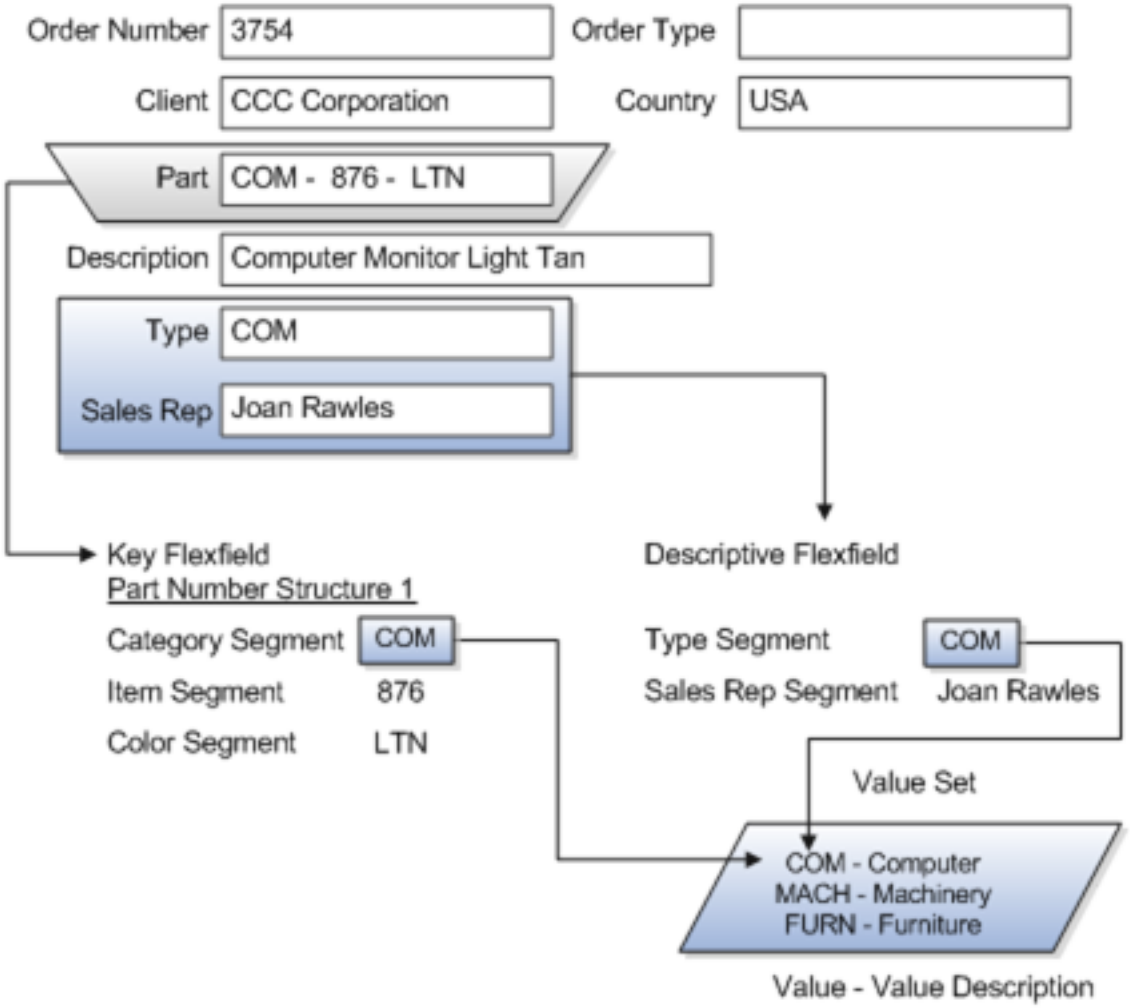
For key flexfields, the attributes in the view object corresponding to the account combination ID, structure instance number (SIN), and data set number (DSN) can't be transient. They must exist in the database table. For key flexfields, the SIN segment is the discriminator attribute, and the account combination 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.



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:** References to protected value sets aren't restricted. 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

- How Flexfields and Value Sets Work Together
- Why can't I edit my flexfield or value set configuration?
- Default Segment Values
- Chart of Accounts Components
- What's the difference between a lookup type and a value set?

Validation Type Options for Value Sets

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 UI 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 can't 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

You can use the format only validation type to enter any value as long as it meets your specified formatting rules. The value must not exceed the maximum length specified for that value set, and it must meet any format requirements for that value set. For example, if the value set permits only numbers, you can enter the value 456 for a value set with a maximum length of three or more, but not the value ABC.

You can also define a range of valid values for your value set by using the Minimum Value and Maximum Value fields. After you've defined a range of values, you can't define a new valid value that falls outside this range. For example, you might create a value set with a number type format where the user can enter only the values between 0 and 100.

> **Note:** For value sets with the value data type of Character, you can enforce minimum and maximum values only if the value set code ends with `_VALIDATE_CHAR_MINMAX`.

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 can't 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 user-defined 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 configure 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's 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 on multiple prior segments in the same context structure. You can't reference other flexfield segments in the table-validated value set's WHERE clause. Which means, the WHERE clause can't 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 options 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 options 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 can't specify any other action.
- When defining a condition that's 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's 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 can't 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 can't change this option if you decide to provide them later.

Related Topics

- Considerations for Bind Variables in Table-Validated Value Sets
- What happens if a value set is security enabled?
- Add Attributes to the Manage Value Sets Page

Considerations for Planning Value Sets

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

The following aspects are important in planning value sets:

- List of values
- Plain text
- 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
- User-defined list. Also include a sub list.
- Dependent user-defined 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 user-defined 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 comprise 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're 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 applications are running in different locales, you might need to provide different translations for the values and descriptions.

Plain Text

Use a format-only value set when you want to allow 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 align the text to either side, 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 text alignment and zero-filling of entered numbers (affects values that include only the digits 0-9).

Option	Description

Note: You can't 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 can't 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 can't edit the data security resource name after you save your changes.

Related Topics

- Overview of Value Sets
- How Flexfields and Value Sets Work Together
- Validation Type Options for Value Sets
- Default Segment Values
- What's the difference between a lookup type and a value set?

Create Value Sets

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.

A 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.

Considerations for Bind Variables in Table-Validated Value Sets

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

These 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's 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> = :{SEGMENT.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's 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 can't 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's assigned to the value set that's identified by the `value_set_code`. The data type of the bind value is the same as the data type of the segment's column.

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

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

Flexfield Internal Code

`:{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 can't assign a table value set to a context segment if the WHERE clause uses VALUESET.value_set_code or SEGMENT.segment_code bind variables.

Related Topics

- Create and Edit Table-Validated Value Sets Based on Lookups

Create and Edit Table-Validated Value Sets Based on Lookups

While creating and editing value sets, use a predefined list of values that are stored in an application table.

You can specify which table you want to use, along with a WHERE clause to limit the values you want to use for your set. For example, let's use the FND_LOOKUPS view as the basis for a table-validated value set. The lookup meaning

corresponds to the **Value Column Name** field and the lookup description corresponds to the **Description Column Name** field.

Create a Value Set Based on a Lookup

1. In the Setup and Maintenance work area, go to the **Manage Value Sets** task.
2. On the Manage Value Sets page, click the **Create** icon.
3. On the Create Value Set page, enter these values:

Field	Value
Value Set Code	SAMPLE_VALUE_SET_CODE
Description	Sample lookup-based table validated value set
Module	Oracle Middleware Extensions for Applications
Validation Type	Table
Value Data Type	Character

4. In the Definition section, enter these values:

Field	Value
FROM Clause	FND_LOOKUPS
Value Column Name	Meaning
Description Column Name	Description
Start Date Column Name	start_date_active
End Date Column Name	end_date_active
Enabled Flag Column Name	'Y'
ID Column Name	lookup_code
WHERE Clause	lookup_type = 'SAMPLE_LOOKUP_TYPE'

Field	Value
ORDER BY Clause	display_sequence, meaning

5. Click **Save and Close**.

Edit a Value Set Based on a Lookup

1. In the Setup and Maintenance work area, go to the **Manage Value Sets** task.
2. On the Manage Value Sets page, use the Search text field to search for a value set.
3. Select a value set and click the **Edit** icon.
4. On the Edit Value Set page, modify the existing values.
5. Click **Save and Close**.

Related Topics

- How do I update existing setup data?

Add Attributes to the Manage Value Sets Page

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, 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.

Import Value Set Values

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 hundreds of values might be more efficient than creating them individually using the **Manage Value Sets** task. However, for just a few values, it might be quicker to perform the relevant tasks. Here are a few things you need to keep in mind while creating these files:

- Use a vertical bar or pipe (|) as a delimiter between fields for both the header and the value rows.
- As per the Oracle Fusion Cloud WebCenter Content specification, set the file encoding to UTF-8 without the Byte Order Mark (BOM).

- Make sure that the first line in the file is a header line, and the files look exactly the same as shown in the sample file.

Here's a sample file for value set value. You can see the header values at the beginning of the file, followed by the line entries of the two value set codes that are to be imported.

```
VALUESETCODE|VALUE|ENABLEDFLAG
FLEX_KFF1_IND_CHR_L10_ZEROFILL|5000|Y
FLEX_KFF1_IND_CHR_L10_ZEROFILL1|5001|N
```

To import value set values, do these tasks:

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

   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** task in the Setup and Maintenance work area.
2. Upload the flat file to the content repository using the File Import and Export page.
3. Import the file using the appropriate **Manage Value Sets** task in the Setup and Maintenance work area. To import the file, do these tasks:
   a. In the Manage Value Sets page, click **Actions** > **Import Related Values**.
   b. In the **Account** field, select the user account containing the flat file.
   c. In the **Related Value Set File Name** field, enter the name of the related value set file you uploaded using the File Import and Export page.
   d. In the **Related Values File Name** field, enter the name of the related values file you uploaded using the File Import and Export page.
   e. 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

- Requirements for Flat Files to Upload Value Set Values
- Upload Value Set Values Process
- Overview of Files for Import and Export

Import Related Value Sets

You can import related value sets to your application by creating separate text files containing the related value set and related values. You can then upload these files to the Oracle Fusion Cloud WebCenter Content document repository.

Here are a few things you need to keep in mind while creating these files:

- Use a vertical bar or pipe (|) as a delimiter between fields for both the header and the value rows.

- As per the Oracle WebCenter Content specification, set the file encoding to UTF-8 without the Byte Order Mark (BOM).
- Make sure that the first line in the file is a header line, and the files look exactly the same as shown in the sample file.

Here's a sample file for related value set. You can see the header values at the beginning of the file, followed by the line entries of the two value set codes to be imported.

```
VALUE_SET_CODE1|VALUE_SET_CODE2|ENABLED_FLAG
FLEX_KFF1_IND_CHR_L10_ZEROFILL|FLEX_KFF1_IND_NUM_P6_S0|Y
FLEX_KFF1_IND_CHR_L10_ZEROFILL|FLEX_KFF1_IND_CHR_L2_STATES|N
```

Here's a sample file for related values. You can see the header values at the beginning of the file, followed by the line entries of the two value set codes to be imported.

```
VALUE_SET_CODE1|VALUE_SET_CODE2|VALUE1|VALUE2|ENABLED_FLAG
FLEX_KFF1_IND_CHR_L10_ZEROFILL|FLEX_KFF1_IND_NUM_P6_S0|1234567890|0|Y
FLEX_KFF1_IND_CHR_L10_ZEROFILL|FLEX_KFF1_IND_CHR_L2_STATES|1234567890|AK|Y
```

To import related value sets, do these tasks:

1. Create flat files containing the values and the related values in the value set that you want to add or update.

   When creating the files, you must specify an existing value set code and values. If the related value sets don't exist, add the related value sets using the appropriate **Manage Value Sets** task in the Setup and Maintenance work area.
2. Upload the flat files to the content repository using the File Import and Export page.
3. Import the files using the appropriate **Manage Value Sets** task in the Setup and Maintenance work area. To import the file, do these tasks:

   a. In the Manage Value Sets page, click **Actions** > **Import Related Values**.
   b. In the **Account** field, select the user account containing the flat file.
   c. In the **Related Value Set File Name** field, enter the name of the related value set file you uploaded using the File Import and Export page.
   d. In the **Related Values File Name** field, enter the name of the related values file you uploaded using the File Import and Export page.
   e. Click **Upload**.

If you upload the related value set file along with the related values file, first the related value set file is processed and then the related value file is processed. If the relationship between two value sets doesn't exist then a new relationship is created in the FND_VS_RELATED_VALUES table. Otherwise, the Enabled column is updated.

Related Topics

- Overview of Files for Import and Export

Requirements for Flat Files to Upload Value Set Values

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 values in the value set you want to add or update. You upload these flat files using the File 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 mandatory columns. The following table lists the three mandatory columns and their data types.

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 mandatory columns. The following table lists the four mandatory columns and their data types.

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 optional columns. The following table lists the optional columns for both dependent and independent value sets.

Column Name	Type
`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	Type
`User-defined Value Attribute1 ... User-defined Value Attribute10`	VARCHAR2(30)

Optional Columns to Support Descriptive Flexfields

You can also capture these additional columns on the value set values in the Manage Value Sets page. To capture these additional columns, you must configure the FND_VS_VALUES descriptive flexfield. This descriptive flexfield expects the context code to correspond to the value set code.

Column Name	Description
`globalSegment1ApiName`	This column corresponds to the API Name for a global segment defined in the FND_VS_VALUES descriptive flexfield.
`contextSensitiveSegment1ApiName`	This column corresponds to the API Name for a context sensitive segment defined in the FND_VS_VALUES descriptive flexfield.

Here's a sample file that contains the value set codes:

```
ValueSetCode|IndependentValue|IndependentValueNumber|IndependentValueDate|IndependentValueTimestamp|Value|
ValueNumber|ValueDate|ValueTimestamp|TranslatedValue|Description|EnabledFlag|StartDateActive|EndDateActive|
SortOrder|SummaryFlag| globalSegment1ApiName | contextSensitiveSegment1ApiName
VS1|||||DFF|||||Testing for WS updated|Y|2000-01-11|2000-01-2|1||seg1v1|ndseg1vs3
VS1|||||KFF|||||Testing for WS updated without segments|Y|2000-1-11|2000-01-22|1|seg1v2|nd1seg1vs4
```

Related Topics

- Import Value Set Values
- Overview of Files for Import and Export

Manage Values

Independent, dependent, and subset value sets require a user-defined list of valid values. Use the Manage Values page to create and manage values for a value set. You can also manage the order in which the value appears.

Create Values for a Value Set

1. In the Setup and Maintenance work area, go to the **Manage Value Sets** task.
2. Use the Search text field to search for a value set.
3. Select a value set and click **Manage Values**.
4. On the Manage Values page, click the **Create** icon and enter new values for the value set.
5. Click **Save and Close**.

Delete Values from a Value Set

1. In the Setup and Maintenance work area, go to the **Manage Value Sets** task.
2. Use the Search text field to search for a value set.
3. Select a value set and click **Manage Values**.
4. On the Manage Values page, click the **Delete** icon to delete a value from the value set.
5. Click **Save and Close**.

Upload Value Set Values Process

This process uploads a flat file containing value set values for flexfields. You can use the 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 could be more efficient than using the Import action in the Manage Value Sets tasks in the Setup and Maintenance work area. However, for a 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 File Import and Export page.

Note: Here are some alternate ways of doing this task:

- Using the File Import and Export page. For more information, see *Export and Import of CSV File Packages*.
- Using the SOAP web services or REST APIs. For more information, see the Value Sets REST Endpoints section in the *REST API for Common Features in Oracle Fusion Cloud Applications* guide.

Parameters

Flat File Name

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

Account

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

Related Topics

- Import Value Set Values
- Export and Import of CSV File Packages
- REST API for Common Features in Oracle Fusion Cloud Applications
- What are scheduled processes?

Upload Comma-Separated Values File for Import Process

This process uploads a flat file containing comma-separated values for various artifacts, such as flexfields, messages, lookups, and so on. You can use this process to add setup data for these artifacts or update existing ones.

Before running this process, you must create the flat file in the given format and upload it to the content repository using the File Import and Export work area. To run this process, use the Scheduled Processes work area and specify the required parameters.

Note: Here are some alternate ways of doing this task:

- Using the File Import and Export page. For more information, see *Export and Import of CSV File Packages*.
- Using the SOAP web services or REST APIs. For more information, see the Value Sets REST Endpoints section in the *REST API for Common Features in Oracle Fusion Cloud Applications* guide.

Parameters

Flat File Name

Enter the name of the file you uploaded using the File Import and Export page.

Account

Enter the user account containing the file in the content repository to upload.

File Type

Select the file type for the file you uploaded using the File Import and Export page.

Here are the supported file types and the corresponding file formats and examples for various application artifacts.

Artifact	File Type	File Formats and Examples
`ValueSet`	VALUE_SET	*File Format for Importing Value Set and Value Set Values*
`Value Set Values`	VALUESET_VALUES	
`Related Value Set`	RELATEDVALUESET	*File Format for Importing Related Value Sets*
	RELATEDVALUESETVALUES	
`Messages`	MESSAGES	*File Format for Importing Messages*
	MESSAGE_TOKENS	
`Lookups`	STANDARD_TYPE	*File Format for Importing Lookups*

Artifact	File Type	File Formats and Examples
	COMMON_TYPE	
	SETID_TYPE	
	COMMON_CODE	
	STANDARD_CODE	
	SETID_CODE	
Profiles	PROFILE_OPTIONS	*File Format for Importing Profile Values*
	PROFILE_CATEGORIES	
	PROFILE_CATEGORY_OPTIONS	
	USER_PROFILE_VALUES	
Descriptive Flexfields	DFF_CONTEXT	*File Format for Importing Descriptive Flexfields*
	DFF_LABELED_SEGMENT	
	DFF_SEGMENT	
Extensible Flexfields	EFF_CONTEXT	*File Format for Importing Extensible Flexfields*
	FileEFF_CONTEXT_SEGMENT	
Data Security Objects Data	DS_OBJECTS	*File Format for Importing Data Security Instance Sets and Grants*
	DS_CONDITIONS	
	DS_ACTIONS	
	DS_GRANTS	
Document Sequence Data Loaders	DOCUMENT_SEQUENCE_CATEGORY	*File Format for Importing Document Sequences*
	DOCUMENT_SEQUENCE	
	DOCUMENT_SEQUENCE_ASSIGNMENT	

Artifact	File Type	File Formats and Examples
`Hierarchies Data Loaders`	HIERARCHYNODE	

Related Topics

- Export and Import of CSV File Packages
- REST API for Common Features in Oracle Fusion Cloud Applications
- What are scheduled processes?
- Submit Scheduled Processes and Process Sets

Translation of Flexfield and Value Set Configurations

When you first configure a flexfield or segment, the **translatable** text 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, sign in with that locale, or use the **Settings and Actions** menu. Click your user image or name in the global header to open this menu, and select **Set Preferences** to translate the text for a particular locale. 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 the value set is of type Character with a subtype Translated text. You define the translations by setting the current session to the locale for which you want to define the translation. Then use the Manage Value Sets task to enter the translated values and descriptions for that locale.

For table-validated value sets, if the underlying table supports translations, use a translation view as the basis for a table-validated value set. You can set the **ID Column Name** field to the non-translated primary key and the **Value Column Name** field to the translated key.

Related Topics

- Considerations for Planning Value Sets
- Create and Edit Table-Validated Value Sets Based on Lookups

File Format for Importing Data Security Instance Sets and Grants

When you want to create data security rules in bulk, import data security instance sets and grants into an application, create separate text files for the instance sets and grants.

Then, upload them to the Oracle WebCenter Content document repository. After placing the files in the document repository, you can import the instance sets and grants into the application.

Here are some considerations:

- Only instance sets and grants are supported.
- Create objects or functions before loading data.
- For instance sets, only filter-based instance sets are supported. Filter-based means that the condition is defined as a structured expression.

The following sections contain specific details about each file format.

Instance Sets

Create a file containing the headers listed in the following table:

Table that lists headers for Instance Sets

Header	Data Type	Data Size	Description	Required or Optional	Editable? (Y/N)
`ObjName`	String	80	This value determines the data security object name.	Required	N
`InstanceSetName`	String	80	This value determines the name of the condition.	Required	N
`DisplayName`	String	2000	This value determines the display name of the condition.	Required	N
`Description`	String	2000	This value determines the description of the condition.	Optional	Y
`Conjunction`	String	3	This value determines whether the condition matches the specified attribute. Value for this field is either All or Any.	Required	Y
`TreeOperator`	String	3	This value determines whether an operator takes the first list of values or the second list of values. The available values are Yes or No. If set to No, then the operator takes the first list of values. If set to Yes, then the operator takes the second list of values.	Required	Y
`Operator`	String	20	If the condition is not a TREE operator, then the value for this field is one of the following values:	Required	Y

Header	Data Type	Data Size	Description	Required or Optional	Editable? (Y/N)
			• EQUALTO • NOTEQUALTO • LESSTHAN • GREATERTHAN • LESSTHANEQUAL • GREATERTHANEQ • BETWEEN • NOTBETWEEN • CUSTOM • LIKE • STARTSWITH • ENDSWITH • ISNULL • ISNOTNULL • CONTAINS • DOESNOTCONTAI If the condition is a TREE operator, then the value for this field is one of the following values: • IS_CHILD_OF • IS_ DESCENDENT_ OF • IS_LAST_ DESCENDENT_ OF • IS_PARENT_OF • IS_ANCESTOR_ OF • IS_FIRST_ ANCESTOR_OF • IS_SIBLING_OF		
`Value`	String	2000	The value for this field is a list of comma-separated values.	Required	Y
`TreeStructureCode`	String	80	The value for this field is a tree structure code for the condition type tree.	Optional	Y
`TreeCode`	String	80	The value of this field is tree code for the condition type tree.	Optional	Y

Header	Data Type	Data Size	Description	Required or Optional	Editable? (Y/N)
FilterColumn	String	80	This value determines whether the condition that is specified matches with the values in the Value column.	Optional	Y
DataType	String	30	This value determines the data type for the filter column.	Optional	Y

Here's a sample file for instance sets:

```
ObjName|InstanceSetName|DisplayName|Description|Conjuction|TreeOperator|Operator|Value|TreeStructureCode|
TreeCodeFLEX_KFF1_IND_CHR_L10_SEC|TEST_DSUI_OBJ1_INS1|TEST_DSUI_OBJ1_INS1 DN|TEST_DSUI_OBJ1_INS1 DES|All|
No|EQUALTO|1000||FLEX_KFF1_IND_CHR_L10_SEC|TEST_DSUI_OBJ1_INS1|TEST_DSUI_OBJ1_INS1 DN|TEST_DSUI_OBJ1_INS1
DES|All|No|GREATERTHAN|100||FLEX_KFF1_IND_CHR_L10_SEC|TEST_DSUI_OBJ1_INS2|TEST_DSUI_OBJ1_INS2 DN|
TEST_DSUI_OBJ1_INS2 DES|All|No|GREATERTHAN|100||FLEX_KFF1_IND_CHR_L10_SEC|TEST_DSUI_OBJ1_INS3|
TEST_DSUI_OBJ1_INS3 DN|TEST_DSUI_OBJ1_INS3 DES|All|Yes|IS_SIBLING_OF|GB3,QA_CHAR_IND_BI|QA_FLEX_BI_TEST_TS2|
QA_FLEX_BI_TEST_TS2_T2FLEX_DFF1_IND_CHR_L10_SEC|TEST_DS_SEEDED_OBJ_INS1|TEST_DS_SEEDED_OBJ_INS1 DN|
TEST_DSUI_OBJ1_INS1 DES|All|No|EQUALTO|1000||FLEX_DFF1_IND_CHR_L10_SEC|TEST_DS_SEEDED_OBJ_INS1|
TEST_DS_SEEDED_OBJ_INS1 DN|TEST_DSUI_OBJ1_INS1 DES|All|No|GREATERTHAN|100||FLEX_DFF1_IND_CHR_L10_SEC|
TEST_DS_SEEDED_OBJ_INS2|TEST_DS_SEEDED_OBJ_INS2 DN|TEST_DSUI_OBJ1_INS2 DES|All|No|GREATERTHAN|100||
FLEX_DFF1_IND_CHR_L10_SEC|TEST_DS_SEEDED_OBJ_INS3|TEST_DS_SEEDED_OBJ_INS3 DN|TEST_DSUI_OBJ1_INS3 DES|All|Yes|
IS_SIBLING_OF|GB3,QA_CHAR_IND_BI|QA_FLEX_BI_TEST_TS2|QA_FLEX_BI_TEST_TS2_T2
```

Grants

Create a file containing the headers listed in the following table:

Table that lists headers for Grants

Header	Data Type	Data Size	Description	Required or Optional	Editable? (Y/N)
ObjName	String	80	This value determines the data security object name.	Required	N
StartDate	Date	NA	The format of date must be DD/MM/YYYY.	Optional	Y
EndDate	Date	NA	The format of date must be DD/MM/YYYY.	Optional	Y
InstanceSetName	String	80	This value determines the condition name to be used in the grant.	Optional	Y
Name	String	80	This value determines the name of the grant to be created or updated.	Optional	Y

Header	Data Type	Data Size	Description	Required or Optional	Editable? (Y/N)
Description	String	2000	This value gives your grant a description.	Optional	Y
RoleName	String	256	This value determines the role name to be used in the grant.	Optional	N
RoleNameSpace	String	256	This value determines the role name space in which RoleName is searched to get the grant key information.	Optional	N
GranteeType	String	30	This value determines the grantee type of the RoleName. If no value is provided, then 'GROUP' is used as default value.	Optional	N
GranteeKey	String	128	This value determines the grantee key of the RoleName.	Optional	N
GrantType	String	30	This value determines the grant type of the grant. If no value is provided, then 'ALLOW' is used as default value.	Optional	N
ContextName	String	128	This value determines the context name used in the grant.	Optional	Y
ContextValue	String	400	This value determines the context value used in the grant.	Optional	Y
Actions	String	2000	This field contains a comma-separated list of Actions to be used in grant. If no Actions are specified, then 'read' action is used.	Optional	Y
InstancePk1Value	String	256	This value determines the first instance private key value.	Optional	Y
InstancePk2Value	String	256	This value determines the second instance private key value.	Optional	Y
InstancePk3Value	String	256	This value determines the third instance private key value.	Optional	Y
InstancePk4Value	String	256	This value determines the fourth instance private key value.	Optional	Y

Header	Data Type	Data Size	Description	Required or Optional	Editable? (Y/N)
InstancePk5Value	String	256	This value determines the fifth instance private key value.	Optional	Y
Parameter1	String	256	This value determines the first parameter value.	Optional	Y
Parameter2	String	256	This value determines the second parameter value.	Optional	Y
Parameter3	String	256	This value determines the third parameter value.	Optional	Y
Parameter4	String	256	This value determines the fourth parameter value.	Optional	Y
Parameter5	String	256	This value determines the fifth parameter value.	Optional	Y
Parameter6	String	256	This value determines the sixth parameter value.	Optional	Y
Parameter7	String	256	This value determines the seventh parameter value.	Optional	Y
Parameter8	String	256	This value determines the eighth parameter value.	Optional	Y
Parameter9	String	256	This value determines the ninth parameter value.	Optional	Y
Parameter10	String	256	This value determines the tenth parameter value.	Optional	Y

Here's a sample file for grants:

```
ObjName|InstanceSetName|Name|Description|RoleName|RoleNameSpace|GranteeType|GranteeKey|Actions|InstancePk1Value|
InstancePk2Value|InstancePk3Value|InstancePk4Value|InstancePk5Value|Parameter1|Parameter2|Parameter3|
Parameter4|Parameter5|Parameter6|Parameter7|Parameter8|Parameter9|Parameter10FLEX_KFF1_IND_CHR_L10_SEC|
TEST_DSUI_OBJ1_INS1|TEST_DSUI_OBJ1 Policy1|TEST_DSUI_OBJ1 Policy1|ORA_FND_APPLICATION_ADMINISTRATOR_JOB|
fscm|GROUP|BAB069E21C3803B495EA7DDC0116A83C|read|Instance Pk1 Value|Instance Pk2 Value|Instance
Pk3 Value|Instance Pk4 Value|Instance Pk5 Value|Parameter1 Value|Parameter2 Value|Parameter3 Value|
Parameter4 Value|Parameter5 Value|Parameter6 Value|Parameter7 Value|Parameter8 Value|Parameter9 Value|
Parameter10 ValueFLEX_KFF1_IND_CHR_L10_SEC|TEST_DSUI_OBJ1_INS2|TEST_DSUI_OBJ1 Policy2|TEST_DSUI_OBJ1
Policy2|ORA_FND_APPLICATION_ADMINISTRATOR_JOB|fscm|GROUP|BAB069E21C3803B495EA7DDC0116A83C|read|||||
Instance Pk5 Value|Parameter1 Value||||Parameter5 Value|||||Parameter10 ValueFLEX_KFF1_IND_CHR_L10_SEC|
TEST_DSUI_OBJ1_INS3|TEST_DSUI_OBJ1 Policy3|TEST_DSUI_OBJ1 Policy3|ORA_FND_APPLICATION_ADMINISTRATOR_JOB|fscm|
GROUP|BAB069E21C3803B495EA7DDC0116A83C|read|||||Instance Pk5 Value|Parameter1 Value||||Parameter5 Value|||||
Parameter10 ValueFLEX_DFF1_IND_CHR_L10_SEC|TEST_DS_SEEDED_OBJ_INS1|TEST_DS_SEEDED_OBJ Policy1|TEST_DS_SEEDED_OBJ
```

```
Policy1 Description|ORA_FND_APPLICATION_ADMINISTRATOR_JOB|fscm|GROUP|BAB069E21C3803B495EA7DDC0116A83C|read|
Instance Pk1 Value|Instance Pk2 Value|Instance Pk3 Value|Instance Pk4 Value|Instance Pk5 Value|Parameter1
Value|Parameter2 Value|Parameter3 Value|Parameter4 Value|Parameter5 Value|Parameter6 Value|Parameter7 Value|
Parameter8 Value|Parameter9 Value|Parameter10 ValueFLEX_DFF1_IND_CHR_L10_SEC|TEST_DS_SEEDED_OBJ_INS2|
TEST_DS_SEEDED_OBJ Policy2|TEST_DS_SEEDED_OBJ Policy2 Description|ORA_FND_APPLICATION_ADMINISTRATOR_JOB|fscm|
GROUP|BAB069E21C3803B495EA7DDC0116A83C|read|||||Instance Pk5 Value|Parameter1 Value||||Parameter5 Value|||||
Parameter10 ValueFLEX_DFF1_IND_CHR_L10_SEC|TEST_DS_SEEDED_OBJ_INS3|TEST_DS_SEEDED_OBJ Policy3|TEST_DS_SEEDED_OBJ
Policy3 Description|ORA_FND_APPLICATION_ADMINISTRATOR_JOB|fscm|GROUP|BAB069E21C3803B495EA7DDC0116A83C|read|||||
Instance Pk5 Value|Parameter1 Value||||Parameter5 Value|||||Parameter10 Value
```

FAQs for 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 role of the user in the application.

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 application.

For example, Western-region employees may choose only California, Nevada, Oregon, and so on as valid values. They can't select non-Western-region states. Eastern-region employees may choose only New York, New Jersey, Virginia, and so on as valid values, but can't select non-Eastern-region states. Value set security is implemented using Oracle Fusion Cloud Applications data security.

How can I set a default value for a flexfield segment?

When you define or edit a flexfield segment, you pick a value from the assigned value set and set it as default.

You can set the default value for a descriptive flexfield segment to be a parameter. The mapped entity object attribute 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 select 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.

Related Topics

- Default Segment Values

Descriptive Flexfields

Overview of Descriptive Flexfields

Use descriptive flexfields to add 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. See the Overview of Descriptive Flexfields topic in *Configuring and Extending Applications* for more information.

Considerations for Planning Descriptive Flexfields

Once you have identified a flexfield to configure, plan the configuration. Make a list of UI pages and other artifacts that are affected by the configuration. Verify that you have the appropriate roles to view and configure the flexfields.

You can locate flexfields on a page using the **Settings and Actions** menu. Click your user image or name in the global header to open this menu, and select **Highlight Flexfields** to locate flexfields on a page. Once you have configured the flexfields, plan to deploy them for test and production users. Review the tools and tasks available for managing flexfields to add and edit 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 user-defined 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 Transactional 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 Fusion Cloud Applications Help for product-specific information about the use of those values.

Plan the Segments, Segment Properties, and Structure

Identify the user-defined 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's only one context segment available for descriptive flexfields. If you have more than one group of user-defined 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 user-defined 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 numeric 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 list?
    - If you're 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?
    - 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 user-defined attribute.

Plan How Segments Map to Oracle Transactional 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.

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.

Related Topics

- Flexfield Segment Properties
- Overview of Value Sets
- Considerations for Enabling Descriptive Flexfield Segments for Business Intelligence
- Default Segment Values
- Overview of Transactional Business Intelligence Configuration of Descriptive Flexfields

Considerations for Managing Descriptive Flexfields

Configuring descriptive flexfields involves managing the available flexfields registered with your Oracle Fusion Cloud Applications 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 can't enter a number for a segment if that number 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's 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 can't 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're 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 can't 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 using the **Add Segment** icon button is either an independent value set or a format-only value set. The following table shows which type of value set is created depending on the segment display component you select.

Display Component	Value Set Created Using Add Segment
Check Box	Independent
Drop-down List	Independent
List of Values	Independent
Radio Button Group	Independent
Text Field With Search	Independent

Display Component	Value Set Created Using Add Segment
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 `PARAMETER` bind variable in the WHERE clause of the SQL statement.

    - `:{PARAMETER.<parameter_code>}`: Identifies a parameter.

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.

Related Topics

- Validation Type Options for Value Sets
- Default Segment Values
- Considerations for Enabling Descriptive Flexfield Segments for Business Intelligence
- Flexfield Segment Properties
- Why can't I edit my flexfield or value set configuration?

Manage Descriptive Flexfields

Edit Descriptive Flexfields

You can edit your descriptive flexfield by changing its values.

Here are the steps:

1. In the Setup and Maintenance work area, go to the **Manage Descriptive Flexfields** task in the Application Extensions functional area.
2. On the Manage Descriptive Flexfields page, use the text fields in the Search section to search for a descriptive flexfield.
3. Select a descriptive flexfield and click the **Edit** icon.
4. Edit the descriptive flexfield values and click **Save and Close**.

Related Topics

- How do I update existing setup data?

Validate Descriptive Flexfields

Before deploying your descriptive flexfields, you can validate their values to identify and fix errors in advance.

Here are the steps:

1. In the Setup and Maintenance work area, go to the **Manage Descriptive Flexfields** task in the Application Extensions functional area.
2. On the Manage Descriptive Flexfields page, use the text fields in the Search section to search for a descriptive flexfield.
3. Select the descriptive flexfield and click **Actions Validate Flexfield**.
4. In the confirmation dialog box, click **OK**.

Related Topics

- How do I update existing setup data?

Deploy Descriptive Flexfields

After editing the descriptive flexfield, you can deploy your changes. After you deploy the flexfield, you can view your changes in the UI.

To deploy the descriptive flexfield, do these steps:

1. In the Setup and Maintenance work area, go to the **Manage Descriptive Flexfields** task in the Application Extensions functional area.
2. On the Manage Descriptive Flexfields page, use the text fields in the Search section to search for a descriptive flexfield.
3. Select the descriptive flexfield and click **Actions Deploy Flexfield**.
4. In the confirmation dialog box, click **OK**.

Related Topics

- How do I update existing setup data?

Manage Global Segments

Global segments are those segments in a descriptive flexfield that are available throughout the enterprise. These segments always appear regardless of the context. You can create, edit, and delete global segments for a descriptive flexfield.

Note: Once you delete a segment, you can never recover it. So, you should delete only unwanted segments.

Create Global Segments

1. In the Setup and Maintenance work area, go to the **Manage Descriptive Flexfields** task in the Application Extensions functional area.
2. On the Manage Descriptive Flexfields page, use the text fields in the Search section to search for a descriptive flexfield.
3. Select the descriptive flexfield and click the **Edit** icon.
4. On the Edit Descriptive Flexfield page, click the **Create** icon in the Global Segments section.
5. On the Create Segment page, enter the values for the segment.
6. Click **Save and Close**.

Edit Global Segments

1. In the Setup and Maintenance work area, go to the **Manage Descriptive Flexfields** task in the Application Extensions functional area.
2. On the Manage Descriptive Flexfields page, use the text fields in the Search section to search for a descriptive flexfield.
3. Select the descriptive flexfield and click the **Edit** icon.
4. On the Edit Descriptive Flexfield page, select a segment and click the **Edit** icon in the Global Segments section.
5. On the Edit Segment page, edit the segment values.
6. Click **Save and Close**.

Delete Global Segments

1. In the Setup and Maintenance work area, go to the **Manage Descriptive Flexfields** task in the Application Extensions functional area.
2. On the Manage Descriptive Flexfields page, use the Search text field to search for a descriptive flexfield.
3. Select the descriptive flexfield and click the **Edit** icon.

4. On the Edit Descriptive Flexfield page, select a segment and click the **Delete** icon in the Global segments section.
5. In the confirmation dialog box, click **Yes**.

Related Topics

- How do I update existing setup data?

Manage Contexts

A descriptive flexfield may have a context segment to provide context sensitivity. You can create, edit, and delete context segments for a descriptive flexfield.

Create Context

1. In the Setup and Maintenance work area, go to the **Manage Descriptive Flexfields** task in the Application Extensions functional area.
2. On the Manage Descriptive Flexfields page, use the text fields in the Search section to search for a descriptive flexfield.
3. Select the descriptive flexfield and click the Edit icon.
4. On the Edit Descriptive Flexfield page, click **Manage Contexts**.
5. On the Manage Contexts page, click the **Create** icon.
6. On the Create Context page, enter the values for the context.
7. Click **Save and Close**.

Edit Context

1. In the Setup and Maintenance work area, go to the **Manage Descriptive Flexfields** task in the Application Extensions functional area.
2. On the Manage Descriptive Flexfields page, use the text fields in the Search section to search for a descriptive flexfield.
3. Select the descriptive flexfield and click the **Edit** icon.
4. On the Edit Descriptive Flexfield page, click **Manage Contexts**.
5. On the Manage Contexts page, use the Search text field to search for a context.
6. Select the context and click the **Edit** icon.
7. On the Edit Context page, edit the values for the context.
8. Click **Save and Close**.

Delete Context

You can delete unwanted contexts, but once you delete a context, you can never recover it. So, before you delete, be sure that you don't need it anymore.

1. In the Setup and Maintenance work area, go to the **Manage Descriptive Flexfields** task in the Application Extensions functional area.
2. On the Manage Descriptive Flexfields page, use the text fields in the Search section to search for a descriptive flexfield.
3. Select the descriptive flexfield and click the **Edit** icon.
4. On the Edit Descriptive Flexfield page, click **Manage Contexts**.
5. On the Manage Contexts page, use the Search text field to search for a context.
6. Select the context and click the **Delete** icon.

7. In the confirmation dialog box, click **Yes**.

Related Topics

- How do I update existing setup data?

Manage Context-Sensitive Segments

A context-sensitive segment is available based on the context value. Once you define the context value, you can create a context-sensitive segment for the defined context value. You can also edit and delete a context-sensitive segment.

Create Context-Sensitive Segment

1. In the Setup and Maintenance work area, go to the **Manage Descriptive Flexfields** task in the Application Extensions functional area.
2. On the Manage Descriptive Flexfields page, use the text fields in the Search section to search for a descriptive flexfield.
3. Select the descriptive flexfield and click the **Edit** icon.
4. In the Context Sensitive Segments section, select the context from the **Context** list.
5. Click the **Create** icon.
6. On the Create Segment page, enter the values for the segment.
7. Click **Save and Close**.

Edit Context-Sensitive Segment

1. In the Setup and Maintenance work area, go to the **Manage Descriptive Flexfields** task in the Application Extensions functional area.
2. On the Manage Descriptive Flexfields page, use the text fields in the Search section to search for a descriptive flexfield.
3. Select a descriptive flexfield and click the **Edit** icon.
4. On the Edit Descriptive Flexfield page, select a segment and click the **Edit** icon under the Context-Sensitive Segments section.
5. On the Edit Segment page, edit the segment values.
6. Click **Save and Close**.

Related Topics

- How do I update existing setup data?

Enable Auditing for Descriptive Flexfields

You can enable auditing for your descriptive flexfield to keep a track of all the changes made on various objects and attributes in the application.

1. In the Setup and Maintenance work area, go to the **Manage Audit Policies** task.
2. Go to the Oracle Fusion Applications section and click **Configure Business Object Attributes**.
3. Click **Product** and select the product area for which you want to enable audit. The business objects associated with that product appear in the Objects section.
4. Click the check boxes next to the listed business objects to select them. If the selected object has attributes, they appear in the Audited Attributes area.

5. Click the **Add** icon to add attributes from the Select and Add Audit Attributes page.
6. On the Select and Add Audit Attributes page, click the Flexfields (Additional Attributes) check box to add the flexfields attribute.
7. Click **OK**.
8. Click **Save and Close**. You return to the Manage Audit Policies page.
9. In the Oracle Fusion Applications section, select **Auditing** from the **Audit Level** drop-down. This setting enables auditing for descriptive flexfields.
10. Click **Save and Close**.

Configure Descriptive Flexfield Segments for a Lookup Type

Depending on the configuration level of a lookup type, you can configure the lookup codes or their meanings.

To capture more information for the lookup type, you can also configure the descriptive flexfield segments for the lookup type. For example, let's configure the descriptive flexfield segments for the US_STATE lookup type. Here are the tasks we need to do:

- Create the US_STATE context code
- Create the context-sensitive segments
- Deploy the changes
- Add a context segment value for the lookup type

Create the US_STATE Context Code

1. In the Setup and Maintenance work area, go to the **Manage Descriptive Flexfields** task in the Application Extensions functional area.
2. On the Manage Descriptive Flexfield page, use the Search text field to search for FND_LOOKUP_VALUES_B descriptive flexfield.
3. Select the descriptive flexfield and click the **Edit** icon.
4. On the Edit Descriptive Flexfield page, go to the Context Segment section and specify these values:

Field	Value
Default Type	Parameter
Default Value	LookupType
Derivation Value	LookupType
Display Type	List of Values

5. Click **Manage Contexts**.
6. On the Manage Contexts page, go to the Search Results section and click the **Create** icon.
7. On the Create Context page, enter the context code as **US_STATE**.
8. Click **Save and Close**.

Create the Context Sensitive Segments

1. On the Manage Contexts page, go to the Search Results section and select the **US_STATE** context code.
2. Click the **Edit** icon.
3. On the Edit Context page, go to the Context Sensitive Segments section and click the **Create** icon.
4. On the Create Segment page, specify these values:

Field	Value
Code	Capital
Data type	Character
Value Set	GL_100_CHARACTERS
Default Type	Parameter
Default Value	LookupType
Derivation Value	LookupType
Display Type	List of Values

5. Click **Save and Close**.
6. On the Manage Contexts page, click **Save and Close**.
7. On the Edit Descriptive Flexfield page, click **Save and Close**.

Deploy the Changes

1. On the Manage Descriptive Flexfield page, click **Deploy Flexfield** to deploy the changes made in the FND_LOOKUP_VALUES_B descriptive flexfield.
2. In the confirmation window, click **OK**.
3. Sign out and sign back in to see the changes.

Add a Context Segment Value for the Existing Lookup Type

After you add a new lookup code for the US_STATE lookup type, the context segment appears for the new lookup code. You can add value to the context segment.

1. In the Setup and Maintenance work area, go to the **Manage Common Lookups** task in the Application Extensions functional area.
2. On the Manage Common Lookups page, use the Search text field to search for the US_STATE lookup type.
3. Select the **US_STATE** lookup type.
4. Go to the Lookup Codes section and click the **New** icon.
5. In the **Lookup Code** field, enter **AZ**.
6. In the **Meaning** field, enter the meaning of the lookup code as **Arizona**.

7. Click **Save**.
8. Click the **Expand** icon to the left of the newly added lookup code.
9. In the **Capital** field, enter **Phoenix**.
10. Click **Save and Close**.

Considerations for Enabling Descriptive Flexfield Segments for Business Intelligence

A descriptive flexfield that's registered in the database as enabled for Oracle Transactional Business Intelligence (OTBI) includes a BI-enabled setting for each of its segments. When a global, context, or context-sensitive segment is BI-enabled, it's available for use in Oracle BI.

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 Data Extensions for Oracle Transactional Business Intelligence process to import the flexfield changes into the Oracle Transactional 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 information about logical objects and import, see the Creating and Administering Analytics and Reports guide for your products.

Flattening

When you deploy a BI-enabled descriptive flexfield, the deployment process generates an additional set of flattened Application Development Framework (ADF) business components. The ADF business component is generated in addition to the usual ADF business components and the ADF faces artifacts. The flattened business components include attributes for BI-enabled segments only. Flattening means each user-defined column in each context shows up as an attribute in the 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, what's included as part of the flattened components depend on whether you assign a label to the segment or not:

- 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 context-sensitive 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're 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 can't 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 Transactional Business Intelligence repository in Oracle Cloud implementations, run the Import Oracle Data Extensions for Oracle Transactional Business Intelligence process. For information about logical objects and import, see the Creating and Administering Analytics and Reports guide for your products.

> **Note:** When you import a flexfield into the Oracle Transactional Business Intelligence repository, you see both `<name>_` and `<name>_c` attributes for each segment, along with some other optional attributes. The `<name>` attribute contains the value. The `<name>_c` attribute contains the code of the value set that the value comes from, and is used for linking to the value dimension. You must import both attributes.

Related Topics

- Considerations for Enabling Key Flexfield Segments for Business Intelligence
- Overview of Transactional Business Intelligence Configuration of Descriptive Flexfields
- Overview of Flexfield Change Import

File Format for Importing Descriptive Flexfields

You can import descriptive flexfields to your application by creating separate text files containing contexts and segments. You can then upload these files to the Oracle WebCenter Content document repository.

Here are a few things that you need to keep in mind while creating these files:

- Use a vertical bar or pipe (|) as a delimiter between fields for both the header and the value rows.
- According to the Oracle WebCenter Content specification, set the file encoding to UTF-8 without the Byte Order Mark (BOM).
- Make sure the first line in the file is a header line, and the files look exactly the same as shown in the sample file.

The following sections contain specific details about each file format.

Context

To create a file containing the contexts, include the headers specified in this table:

Header	Data Type	Data Size	Description	Required or Optional
`ApplicationId`	Long	-	This value determines the application your flexfield and context belong to.	Required
`DFFCode`	String	40	This value is the code of the descriptive flexfield where you're adding your context. The value should match the code of a descriptive flexfield that's already in the application.	Required
`ContextCode`	String	80	This value is the code for your context.	Required
`Name`	String	80	This value determines the display name of your context.	Required
`EnabledFlag`	String	1	This value lets you enable or disable your context: • Y: Enable context • N: Disable context	Required
`Description`	String	240	This value describes your context.	Optional
`ContextIdentifier`	String	30	This value determines the API name for your context.	Optional

Header	Data Type	Data Size	Description	Required or Optional
			While naming APIs, follow the naming conventions.	

Here's a sample file for contexts. You can see the header values at the beginning of the file, followed by the line entries of the two value set codes to be imported. For importing several contexts, add more entries in the same format.

```
ApplicationId|DFFCode|ContextCode|Name|EnabledFlag|Description
0|TEST_DEPLOY_DFF1|LOADER_Con1|Loader con Name|Y|Description
```

Segment Label

To create a file containing context segments, include the headers specified in this table:

Header	Data Type	Data Size	Description	Required or Optional
`ApplicationId`	Long	-	This value determines the application your flexfield, context, and segment belong to.	Required
`DFFCode`	String	40	This value is the code of the descriptive flexfield where you're adding your segment. The value should match the code of a descriptive flexfield that's already in the application.	Required
`SegmentLabelCode`	String	30	This value is the code for your segment.	Required
`Name`	String	80	This value is the name of your segment.	Required
`Description`	String	240	This value describes your segment.	Optional
`BIObjectName`	String	128	This value provides the name for the BI-enabled segment.	Optional

Here's a sample file for the segment label. You can see the header values at the beginning of the file, followed by the line entries of the two value set codes to be imported. For importing several segment labels, add more entries in the same format.

```
ApplicationId|DFFCode|SegmentLabelCode|Name|Description|BIObjectName
0|FLEX_DFF1|Label Code 1|Label name 1|Description 1| BIObJ1
```

Segment

To create a file containing context segments, include the headers specified in this table:

Header	Data Type	Data Size	Description	Required or Optional
`ApplicationId`	Long	-	This value determines the application your flexfield, context, and segment belong to.	Required
`DFFCode`	String	40	This value is the code of the descriptive flexfield where you're adding your segment. The value should match the code of a descriptive flexfield that's already in the application.	Required
`ContextCode`	String	80	This value is the code for the context where you're adding your segment. The value should match a context code that's in the application. For Global Segments, use the value for the field as Global Data Elements.	Required
`SegmentCode`	String	30	This value is the code for your segment.	Required
`Name`	String	60	This value is the name of your segment.	Required
`ColumnName`	String	30	This value determines the table column that your segment uses to save data. The value must match a column that's in the application.	Required
`ValueSetCode`	String	60	This value is the code for the value set you want to use in your segment. The value must match a value set code in the application.	Required
`DisplayType`	String	30	This value determines the display type of the segment. Here are the valid values for this attribute: • `INPUT_SEARCH`	Required

Header	Data Type	Data Size	Description	Required or Optional
			• TEXT_BOX • TEXT_AREA • HIDDEN,LOV • POP_UP_LIST • DROP_DOWN_LIST • RADIO_BUTTON_GROUP • CHECKBOX • DATE_TIME	
Prompt	String	80	This value determines the display name of the segment.	Required
ShortPrompt	String	80	This value determines the shortened display name of the segment.	Required
EnabledFlag	String	1	This value lets you enable or disable your segment: • Y: Enable segment • N: Disable Segment	Required
RequiredFlag	String	1	This value indicates whether you must specify the segment: • Y: You must specify the segment. • N: You may not specify the segment.	Required
ReadOnlyFlag	String	1	This value lets you set your segment to be read-only: • Y: The segment is set to read-only. • N: The segment isn't set to read-only.	Required
Description	String	240	This value describes your segment.	Optional
UOMclass	String	20	This value specifies the unit for the data that you add to your segment.	Optional
TerminologyHelpText	String	80	This value describes the segment.	Optional

Header	Data Type	Data Size	Description	Required or Optional
`InFieldHelpText`	String	160	This value provides instructions on how to use the segment.	Optional
`SequenceNumber`	Integer	-	This value specifies the order in which your segments display in your context.	Optional
`DefaultType`	String	30	This value specifies the type of default value for your segment. Here are the types of default values: • `CONSTANT` • `SQL` • `GROOVY_EXPRESSION`	Optional
`DefaultValue`	String	4000	This value provides the default values for your segment.	Optional
`DisplayWidth`	Integer	-	This value specifies the maximum number of characters that can display in a line.	Optional
`DisplayHeight`	Integer	-	This value specifies the maximum number of lines that can display in the segment.	Optional
`RangeType`	String	30	This value specifies the range of values for the low-end and high-end fields. You can set the respective field value as `LOW` or `HIGH`.	Optional
`BIEnabledFlag`	String	1	This value indicates whether your segment is BI enabled: • Y: Your segment is BI enabled. • N: Your segment isn't BI enabled.	Optional
`SegmentIdentifier`	String	30	This value determines the API name for your segment. While naming APIs, follow the naming conventions.	Optional

Here's a sample file for a segment. You can see the header values at the beginning of the file, followed by the line entries of the two value set codes to be imported. For importing several segment labels, add more entries in the same format.

```
ApplicationId|DFFCode|ContextCode|SegmentCode|Name|ColumnName|ValueSetCode|DisplayType|Prompt|ShortPrompt|
EnabledFlag|RequiredFlag|ReadOnlyFlag|Description|UomClass|BIEnabledFlag|BIEqualizationTag
0|FLEX_DFF1|Global Data Elements|SWAMY_TEST_GSEG1|Global seg 1|ATTRIBUTE10_VARCHAR2|EFF_BASIC_FMT_CHR|
TEXT_BOX|Prompt|Short Prompt|Y|N|N|TEST desc aug LABEL TAG 1NOV LOAD && ||Y|LabelC1
```

Related Topics

- Overview of Descriptive Flexfields
- Naming Conventions for Flexfield APIs
- Overview of Files for Import and Export

Extensible Flexfields

Overview of Extensible Flexfields

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.
  - 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, use the Manage Extensible Flexfields task in the Setup and Maintenance work area.

The following aspects are important in understanding extensible flexfields:

- Usages

- Categories
- Pages
- Security
- Protected Extensible Flexfield Data

Usages

Similar to the 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 can 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's no restriction on page references to protected contexts. The pages you create may contain any context, whether protected or not.
- There's 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.

Related Topics

- Overview of Flexfield Deployment
- Considerations for Managing Flexfields
- Considerations for Managing Extensible Flexfields
- How can I access predefined flexfields?
- How do I update existing setup data?

Considerations for Planning Extensible Flexfields

Once you have identified a flexfield, plan the configuration. Make a list of the UI pages and other artifacts that are affected by the configuration. Verify that you have the appropriate roles to view and configure the flexfields.

You can locate flexfields on a page using the **Settings and Actions** menu. Click your user image or name in the global header to open this menu, and select **Highlight Flexfields** to locate flexfields on a page. After you configure the flexfields, plan to deploy them for test and production users. Review the tools and tasks available for managing flexfields to add and edit flexfield segments.

Planning an extensible flexfield involves:

1. Identifying the following:

- A hierarchical structure of categories
- Existing context values
- User-defined attributes, the relevant extensible flexfield segments, segment properties, and the structure

2. Planning the following:
    - Validation rules
    - Initial values
    - Security
    - 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 user-defined attributes of an entity. Some applications provide user interfaces to create and manage an extensible flexfield's category hierarchy.

Contexts and Existing Context Values

If related 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 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 user-defined 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. 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 permitted?
- 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 numeric systems, the radix separator is a decimal point.
- Should the value be within a range?
- Should the value be selected from a list of valid values? If yes, consider the following questions:

- Can you use an existing application table from which to obtain the list of valid values, or do you have to create a list?
- If you're using an existing table, do you have to limit the list of values using a WHERE clause?
- Does the list of valid values depend on the value in another flexfield segment?
- 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 user-defined attribute.

Plan Security

Determine what privileges to set for view and edit access to context attributes, such as providing all 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, 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 having 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.

Related Topics

- Considerations for Managing Extensible Flexfields
- Flexfield Segment Properties

Considerations for Managing Extensible Flexfields

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
    - 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 beginning 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 use format-only 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 beginning 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 in 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 these bind variables in the WHERE clause of the SQL statement.
    - `:{SEGMENT.<segment_code>}`: Identifies a segment in the same context.
    - `:{PARAMETER.<parameter_code>}`: Identifies a parameter.
    - `:{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 can't be a multiple-row context.
    - `:{VALUESET.<value_set_code>}`: Identifies the closest prior segment in the same context that's assigned to the specified value set.
    - `:{FLEXFIELD.<internal_code>}`: Identifies a flexfield.

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 following table shows which type of value set is created depending on the segment display component you select.

Display Component	Value Set Created Using 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.

Related Topics

- Why did my flexfield changes not appear in the UI?
- Flexfield Usages
- Why can't I edit my flexfield or value set configuration?

Considerations for Enabling Extensible Flexfield Segments for Business Intelligence

An extensible flexfield that's registered in the database as enabled for Oracle Transactional Business Intelligence (OTBI) includes a BI Enabled setting for each of its segment instances. When a segment instance is BI-enabled, it's available for use in OTBI.

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 OTBI

After you deploy a business intelligence-enabled flexfield, use the Import Oracle Data Extensions for OTBI process to import the flexfield changes into the Oracle Transactional Business Intelligence repository. Users can make use of the newly-generated attributes in business intelligence applications. For information about logical objects and import, see the Creating and Administering Analytics and Reports guide for your products.

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 Transactional Business Intelligence Repository

After you deploy a business intelligence-enabled flexfield, import the flexfield changes into the Oracle Transactional 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 Transactional 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 Data Extensions for Transactional Business Intelligence process. For information about logical objects and import, see the Creating and Administering Analytics and Reports guide for your products.

> **Tip:** When you import a flexfield into the Oracle Transactional Business Intelligence repository, you see both `<name>_` and `<name>_c` attributes for each segment, along with some other optional attributes. The `<name>_` attribute contains the value. The `<name>_c` attribute contains the code of the value set that the value comes from, and is used for linking to the value dimension. You must import both attributes.

Related Topics

- Considerations for Enabling Descriptive Flexfield Segments for Business Intelligence
- Overview of Extensible Flexfields
- Set Extensible Flexfields as BI-Enabled
- Overview of Flexfield Change Import

Considerations for Managing Extensible Flexfield Categories

Categories are a way of extending the number of context-sensitive segments for a flexfield beyond the columns reserved for flexfield segments.

For example, 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's 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 can't 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

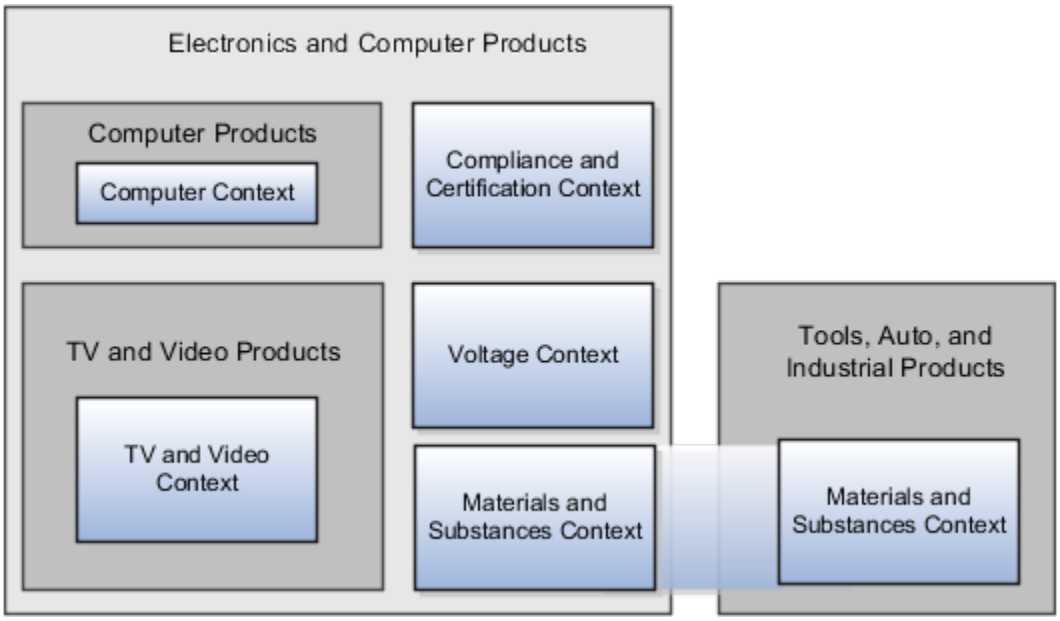
While configuring an extensible flexfield, you can disable a category. The Enabled column in the Category table of the Edit Extensible Flexfield page, indicates which categories are enabled.

> **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 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 following table shows an example of category hierarchy for an extensible flexfield. 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.

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

Example of Configuring an Item Extended Attributes Flexfield

The Item Extended Attributes flexfield provides segments for extending the Item business object. In the Manage Extensible Flexfields task, configure your product business object to include a technical specifications logical page on the UI for the Electronics and Computers category.

In this example, your configuration of this flexfield groups attributes into the following contexts:

- Materials and Substances
- Compliance and Certification
- Voltage

Scenario

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

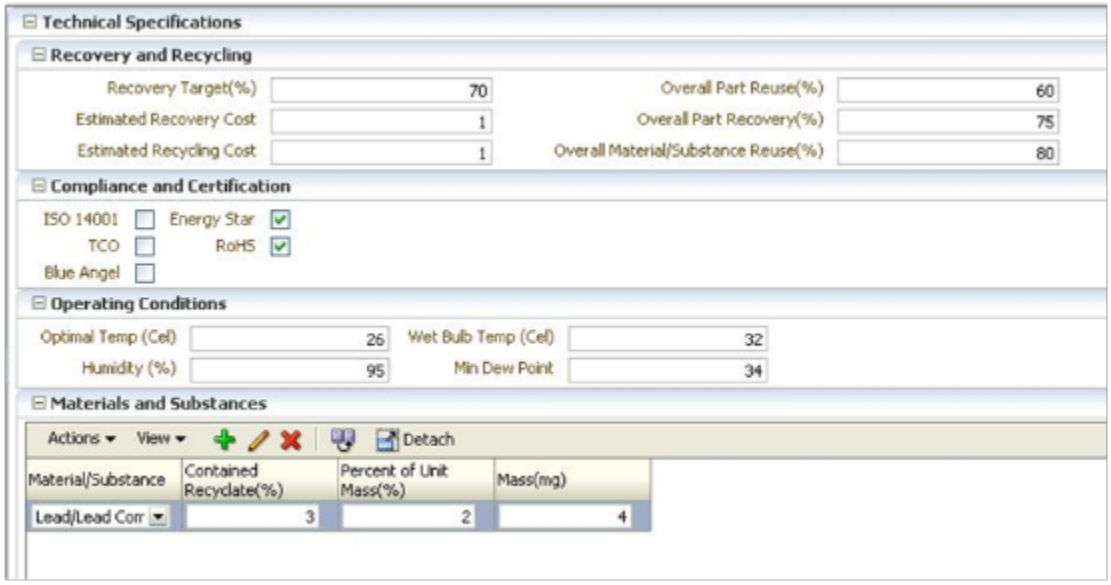
- Page: Electronics Information
    - Context: Compliance and Certification, single row
        - ISO 14001 (International Organization for Standardization for an Environmental Management System)
        - ENERGY STAR (energy efficiency guidelines)
        - ROHS (Restriction of the use of certain hazardous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

The following table summarizes key decisions for this scenario.

Decisions to Consider	In This Example
Which extensible flexfield is available for configuring a hierarchy of categories?	Item Extended Attributes flexfield

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.



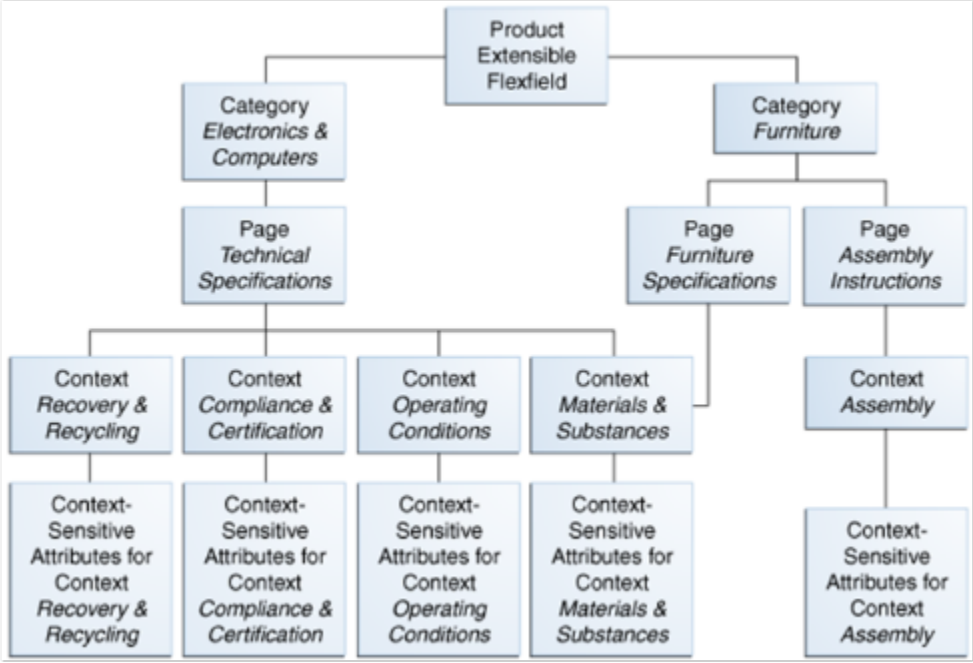
Use logical pages to determine how the contexts appear on the user interface. Use a context to store all the materials and substances required to make a single product. You can configure a context to store multiple rows per entity. The multiple rows are displayed in a table, like the Materials and Substances context.

The Technical Specifications logical page contains the attributes for the four contexts.

- Recovery and Recycling
- Compliance and Certification
- Operating Conditions
- Materials and Substances

The following figure is an example, where the Furniture category is configured to include a Furniture Specifications logical page and an Assembly Instructions logical page. The two

categories (Electronics and Computers and Furniture) share the Materials and 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	Action	Condition
ITEM_EFF_B	A	VOLTAGE_SPEC	edit_nontrans_voltage_ctx	All values
ITEM_EFF_VL	B	COMPLIANCE_SPEC	edit_trans_compliance_ctx	All values
ITEM_EFF_VL	C	COMPUTER_SPEC	edit_trans_attrs	ComputerCategoryFilter
ITEM_EFF_VL	D	TELEVISION_SPEC	edit_trans_attrs	TVCategoryFilter

The following table shows the privileges for the three flexfield contexts.

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

In this example, anyone can view the contexts' attributes, but the edit privileges are restricted as follows:

- Voltage: Only voltage specialists can edit this value.
- Compliance and Certification: Only compliance specialists can edit this value.
- 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.

To sum up, in this entire 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.

File Format for Importing Extensible Flexfields

You can import extensible flexfields to your application by creating separate text files containing value sets, contexts, and context segments. You can then upload these files to the Oracle WebCenter Content document repository.

Here are a few things that you need to keep in mind while creating these files:

- Use a vertical bar or pipe (|) as a delimiter between fields for both the header and the value rows.
- According to the Oracle WebCenter Content specification, set the file encoding to UTF-8 without the Byte Order Mark (BOM).
- Make sure the first line in the file is a header line, and the files look exactly the same as shown in the sample file.

The following sections contain specific details about each file format.

Value Sets

To create a file containing value sets, include the headers specified in this table:

Header	Data Type	Data Size	Description	Required or Optional
`ValueSetCode`	String	60	This value identifies your value set across components.	Required
`ModuleType`	String	60	This value determines the module type of your value set.	Required
`ModuleKey`	String	60	This value determines the specific module of your value set.	Required
`ValidationType`	String	30	This value determines your value set type. Here are the types of value set: • `DEP` • `INDEP`	Required

Header	Data Type	Data Size	Description	Required or Optional
			• **FORMAT** • **SUBSET** • **RELATED**	
ValueDataType	String	30	This value determines the data type that your value set uses. Here are the supported data types: • **VARCHAR2** • **NUMBER** • **DATE** • **TIMESTAMP**	Required
ValueSubType	String	30	This value determines the data subtype that your value set uses. Here are the supported data subtypes: • **TEXT** • **TIME_HM** • **NUMERIC** • **TIME_HMS** • **TRANSLATED**	Required for **ValueDataType VARCHAR2**
MaximumLength	Integer	-	This value determines the maximum length of values in your value set.	Required for **ValueDataType VARCHAR2**
Description	String	240	This value describes your value set.	Optional
Precision	Integer	-	This value determines the number of digits in the number data that you add to your value set.	Optional Use this header only for **ValueDataType NUMBER**
Scale	Integer	-	This value determines the number of digits after the decimal point in the number data that you add to your value set.	Optional Use this header only for **ValueDataType NUMBER**
UppercaseOnlyFlag	String	1	This value indicates whether only uppercase characters are supported for your value sets: • Y: Supports only uppercase characters.	Optional Use this header only for **ValueDataType VARCHAR2**

Header	Data Type	Data Size	Description	Required or Optional
			• N: Supports both uppercase and lowercase characters.	
ZeroFillFlag	String	1	This value lets you add zeroes to the left of the text you add in your value set till the text length matches the value in the **MaximumLength** header: • Y: Zeroes are added. • N: Zeroes aren't added.	Optional Use this header only for **ValueDataType** **VARCHAR2**.
SecurityEnabledFlag	String	1	This value indicates whether data security resource is added to your value set: • Y: Data security resource is added to your value set. • N: Data security resource isn't added to your value set.	Optional
DataSecurityObjectNam	String	60	This value selects the data security resource that you want to add to your value set. The value must match a data security resource in the application.	Required if **SecurityEnabledFlag** is set to Y.
MinimumValue	String	150	This value specifies the minimum value for the values in your value set.	Optional
MaximumValue	String	150	This value specifies the maximum value for the values in your value set.	Optional
IndependentValueSetCo	String	60	This value determines the independent value set your dependent or subset value set is linked to.	Required for only **ValidationType**, **DEP**, and **SUBSET** validation types. Don't use this header for any other validation types.

Here's a sample file for value sets. You can see the header values at the beginning of the file, followed by the line entries of the three value sets that are to be imported. For importing several value sets, add more entries in the same format.

```
ValueSetCode|ModuleType|ModuleKey|ValidationType|ValueDataType|ValueSubtype|MaximumLength|Description
```

```
VS_TEST_91|APPLICATION|FND|INDEP|VARCHAR2|TEXT|2|desc1
VS_TEST_92|APPLICATION|FND|INDEP|VARCHAR2|TEXT|3|
VS_TEST_93|APPLICATION|FND|INDEP|VARCHAR2|TEXT|3|desc3
```

Context

To create a file containing the contexts, include the headers specified in this table:

Header	Data Type	Data Size	Description	Required or Optional
`ApplicationId`	Long	-	This value determines the application your flexfield and context belong to.	Required
`EFFCode`	String	40	This value is the code of the extensible flexfield where you're adding your context. The value should match the code of an extensible flexfield that's already in the application.	Required
`ContextCode`	String	80	This value is the code for your context.	Required
`Name`	String	80	This value determines the display name of your context.	Required
`MultirowFlag`	String	1	This value indicates whether multiple rows are supported for your context: • Y: Your context supports multiple rows. • N: Your context supports only single row.	Required
`EnabledFlag`	String	1	This value lets you enable or disable your context: • Y - Enable context • N - Disable context	Required
`Description`	String	240	This value describes your context.	Optional
`TranslatableFlag`	String	1	This value indicates whether the segments in your context are translatable:	Optional

Header	Data Type	Data Size	Description	Required or Optional
			• Y: Segments in your context are translatable. • N: Segments in your context aren't translatable.	
`ContextIdentifier`	String	30	This value determines the API name for your context. While naming APIs, follow the naming conventions.	Optional
`InstructionHelpText`	String	400	This value provides user instructions on how to use the context region.	Optional
`FlexfieldUsageCode`	String	30	This value determines the usage code for your context. The value should match a usage code in the application.	Required
`ViewPrivilegeName`	String	400	This value determines the privileges that can view this context usage. The value should match a privilege in the application.	Optional
`EditPrivilegeName`	String	400	This value determines the privileges that can edit this context usage. The value should match a privilege in the application.	Optional
`CExtAttribute1`	String	150	This value adds a user-defined attribute to a context usage. To add your user-defined attributes, you can use up to 5 headers of this type, `CExtAttribute1` to `CExtAttribute5`.	Optional
`CONTEXT_ CExtAttribute1`	String	150	This value adds a user-defined attribute to a context. You can use up to 5 headers of this type, `CONTEXT_ CExtAttribute1` to `CONTEXT_ CExtAttribute5` to add your user-defined attributes.	Optional

Header	Data Type	Data Size	Description	Required or Optional
DatabaseViewNamePrefi:	String	15	This value is used as a prefix for the database view name for a given context. While entering this value, follow the naming conventions similar to that of flexfield API names.	Optional

Here's a sample file that contains the header values at the beginning and lists three contexts to be imported. For importing several contexts, add more entries in the same format.

```
ApplicationId|EFFCode|ContextCode|Name|EnabledFlag|MultirowFlag|Description|FlexfieldUsageCode|
ViewPrivilegeName|CExtAttribute1|CExtAttribute2|CExtAttribute3|CExtAttribute4|CExtAttribute5
0|FLEX_SN_EFF1|OBJ_TEST_4|Object test 44|Y|N|desc 44 3363|FLEX_SN_EFF1_USAGE2|flex_sn_sitems_view|||||
0|FLEX_SN_EFF1|OBJ_TEST_3|Object test 33|Y|N||FLEX_SN_EFF1_USAGE1|flex_sn_view|||||NEW_TEST11
0|FLEX_SN_EFF1|OBJ_TEST_3|Object test 33|Y|N|new desc aug 14|FLEX_SN_EFF1_USAGE2|flex_sn_sitems_edit|
CE1_TESTupd_aug15||||
```

Context Segment

To create a file containing context segments, include the headers specified in this table:

Header	Data Type	Data Size	Description	Required or Optional
ApplicationId	Long	-	This value determines the application your flexfield, context, and segment belong to.	Required
EFFCode	String	40	This value is the code of the extensible flexfield where you're adding your context segment. The value should match the code of an extensible flexfield that's already in the application.	Required
ContextCode	String	80	This value is the code of the context where you're adding your segment. The value should match a context code that's in the application.	Required
SegmentCode	String	30	This value is the code for your segment.	Required
Name	String	60	This value is the name for your segment.	Required
ColumnName	String	30	This value determines the table column that your	Required

Header	Data Type	Data Size	Description	Required or Optional
			segment uses to save data. The value must match a column that's in the application.	
`ValueSetCode`	String	60	This value is the code for the value set you want to use in your segment. The value must match a value set code in the application.	Required
`DisplayType`	String	30	This value determines the display type of the segment. Here are the valid values for this attribute: • `TEXT_BOX` • `TEXT_AREA` • `RICH_TEXT_EDITOR` • `HIDDEN,LOV` • `POP_UP_LIST` • `DROP_DOWN_LIST` • `RADIO_BUTTON_GROUP` • `STATIC_URL` • `CHECKBOX` • `COLOR` • `DATE_TIME`	Required
`Prompt`	String	80	This value determines the display name of the segment.	Required
`ShortPrompt`	String	80	This value determines the shortened display name of the segment.	Required
`EnabledFlag`	String	1	This value lets you enable or disable your segment: • Y - Segment is enabled • N - Segment is disabled	Required
`RequiredFlag`	String	1	This value indicates whether you must specify the segment.	Required

Header	Data Type	Data Size	Description	Required or Optional
			• Y: You must specify the segment. • N: You may not specify the segment.	
`ReadOnlyFlag`	String	1	This value lets you set your segment to be read-only. • Y: The segment is set to read-only. • N: The segment isn't set to read-only.	Required
`Description`	String	240	This value describes your segment.	Optional
`UOMclass`	String	20	This value determines the unit for the data you add to your segment.	Optional
`TerminologyHelpText`	String	80	This value provides a description for the segment.	Optional
`InFieldHelpText`	String	160	This value provides instructions on how to use the segment.	Optional
`SequenceNumber`	Integer	-	This value determines the order in which your segments are displayed in your context.	Optional
`DefaultType`	String	30	This value specifies the type of default value for your segment. Here are the types of default values: • `CONSTANT` • `SQL` • `GROOVY_EXPRESSION`	Optional
`DefaultValue`	String	4000	This value provides the default values for your segment.	Optional
`DisplayWidth`	Integer	-	This value specifies the maximum number of characters that display in a line.	Optional

Header	Data Type	Data Size	Description	Required or Optional
`DisplayHeight`	Integer	-	This value specifies the maximum number of lines that display in the segment.	Optional
`CheckboxCheckedValue`	String	30	This value determines the value of a selected check box in your segment.	Required for display type `CHECKBOX`.
`CheckboxUncheckedValu`	String	30	This value determines the value of a check box that's not selected in your segment.	Required for display type `CHECKBOX`.
`RangeType`	String	30	This value specifies the range of values for the low-end and high-end fields. You can set the value as `LOW` or `HIGH` for your low-end and high-end fields respectively.	Optional
`BIEnabledFlag`	String	1	This value indicates whether your segment is BI enabled. • Y: Your segment is BI enabled. • N: Your segment isn't BI enabled.	Optional
`MultirowUniqueKeyFlag`	String	1	This value indicates whether the segment is marked as a unique key for context with multiple rows. • Y: Segment is marked as a unique key. • N: Segment isn't marked as a unique key. This is the default value. At least one segment in a context with multiple rows must be marked as the unique key.	Required for contexts with multiple rows
`ShowValueDescription`	String	1	This value lets you display the value set description for the value set associated with the segment: • Y: The value set description is displayed for the value	Optional

Header	Data Type	Data Size	Description	Required or Optional
			set associated with the segment. • N: The value set description isn't displayed for the value set associated with the segment.	
`SegmentIdentifier`	String	30	This value determines the API name for your segment. While naming APIs, follow the naming conventions.	Optional

Here's a sample file for the context segment. You can see the header values at the beginning of the file, followed by the line entry of a context segment that's to be imported. For importing several context segments, add more entries in the same format.

```
ApplicationId|EFFCode|ContextCode|SegmentCode|Name|ColumnName|ValueSetCode|DisplayType|Prompt|ShortPrompt|
EnabledFlag|RequiredFlag|ReadOnlyFlag|Description|UomClass
0|FLEX_SN_EFF1|ColdSourceTargetContext|Cold_TEST_SEG1|cold seg 1|ATTRIBUTE_CHAR3|EFF_BASIC_FMT_CHR|TEXT_BOX|
Prompt|Short Prompt|Y|Y|N|TEST desc aug 14_3PM|
```

Related Topics

- Overview of Extensible Flexfields
- Import Extensible Flexfields
- Naming Conventions for Flexfield APIs
- Overview of Files for Import and Export

Import Extensible Flexfields

Use the Import option on the Manage Extensible Flexfield page to bulk import the flexfield data that includes value set, context, and context segment details.

Prerequisite

The files containing the flexfield details are available in the document repository of Oracle WebCenter Content.

Importing Flexfields

To import flexfields:

1. Sign in to the application as an implementation consultant or an administrator.
2. In the Setup and Maintenance work area, open the **Manage Extensible Flexfield** task or a similar task for importing flexfields.
3. In Search Results, from the Actions menu, select **Import**.
4. On the Upload Flexfield Data dialog box, select the WebCenter Content account to which the files were uploaded.

5. Enter the names of the separate files containing the value set, context, and context segment information. The names here must match with the names of the files uploaded to the selected account.
6. Click **Upload**. The flexfield details are imported.

> **Note:** If the import fails, click the link to the log file on the confirmation dialog box and examine the cause of failure.

Related Topics

- File Format for Importing Extensible Flexfields

FAQs for 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.

Related Topics

- Why did my flexfield changes not appear in the UI?

Key Flexfields

Overview of Key Flexfields

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
- How to Identify Segments
- How the Override, Visible, and Rendered Properties Work
- Structures and segment 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, use the Manage Key Flexfields task in the Setup and Maintenance work area. 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 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. Each segment contains these 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

How to Identify Segments

You can identify and control a segment within a key flexfield using a segment label. Segment labels act as tags for the segments, and these labels are defined by application developers. For example, say in the accounting flexfield, you want to identify which segment contains balancing information and which one contains natural account information. A segment label identifies the segment that contains natural account information. When you define your accounting flexfield, you must specify the segment labels for each segment. Some segment labels must be unique, and can't 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.

You can also use a segment label to search for segments, such as the Cost Center label for all segments across key flexfields that store a value for the cost center. To identify and control a segment in key flexfield, you can set several segment properties in the flexfield XML with a literal value or an EL expression. These properties are attributes of the `<flexfieldLabeledSegmentHint>` element of the flexfield. These property settings apply to all segments that have the specified segment label assigned to them.

Property	Description
SegmentLabel	Indicates the segment label for the segment being configured.
Rendered	Indicates whether the segment is available on the page. You can set the value to **True**, **False**, or **EL expression**
Required	Indicates whether users must enter some data for the component before they save. You can set the value to **True**, **False**, or **EL expression**.
Read-only	Indicates whether the users can modify the segment. You can set the value to **True**, **False**, or **EL expression**.
Visible	Indicates whether the segment is visible on the page. You can set the value to **True**, **False**, or **EL expression**.
Label	Indicates the display text for the segment.
ShortDesc	Indicates the text that appears when users hover over the component.
Columns	Indicates the width of the text control in terms of the number of characters shown. The number of columns is calculated based on the default font size of the browser. You can set this property with a literal value or **EL expression**.
Override	Indicates whether a certain child component inside a parent component is visible. You can set the value to **True**, **False**, or **EL expression**. When you use this property along with the visible property, it overrides the visible property of the parent component.

How the Override, Visible, and Rendered Properties Work

Let's look at a few scenarios:

- Here's a snippet of code without the override property:

```
<fnd:keyFlexfieldPartial value="#{bindings.Kff1PaInstanceIterator}" id="kfp1" visible="false">
 <fnd:flexfieldLabeledSegmentHint
 segmentLabel="SEGMENT_LABEL_R1" visible="true"/>
</fnd:keyFlexfieldPartial>
```

  In this case, the `visible` property of the child component is set to **true** and the `visible` property of the parent component is set to **false**. So, segments with `SEGMENT_LABEL_R1` aren't visible because the `visible` property of the child component is used in conjunction with the `visible` property of the parent component.

- Here's a snippet of code with the override property:

```
<fnd:keyFlexfieldPartial value="#{bindings.Kff1PaInstanceIterator}" id="kfp1" visible="false">
 <fnd:flexfieldLabeledSegmentHint
 segmentLabel="SEGMENT_LABEL_R1" visible="true" override="true"/>
</fnd:keyFlexfieldPartial>
```

  Here also, the `visible` property of the child component is set to **true** and the `visible` property of the parent component is set to **false**. But you're also setting the `override` property of the child component to **true**. So, segments with `SEGMENT_LABEL_R1` are visible because the `visible` property of the child component overrides the `visible` property of the parent component.

Remember, you must not use the `rendered` property along with the `override` property, because when you set the `rendered` property to **false** at the flexfield level, the Oracle ADF module ignores the rendered property at the segment level. For example, consider this snippet of code:

```
<fnd:keyFlexfieldPartial value="#{bindings.Kff1PaInstanceIterator}" id="kfp1" rendered="false">
 <fnd:flexfieldLabeledSegmentHint
 segmentLabel="SEGMENT_LABEL_R1" rendered="true" override="true"/>
</fnd:keyFlexfieldPartial>
```

Though you set the `rendered` property to **true** at the segment level, this property is ignored. That's because the `rendered` property is set to **false** at the flexfield level. So, make sure that you use the `visible` property instead of the `rendered` property along with the `override` property.

Structures and Segment Instances

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.

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 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. The following table lists the levels at which dynamic combination creation may be enabled.

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.

Related Topics

- Overview of Flexfield Deployment
- Considerations for Managing Flexfields
- Key Flexfield Structures
- How can I access predefined flexfields?
- How do I update existing setup data?

Considerations for Planning Key Flexfields

Your first step in planning your key flexfields is to determine which key flexfields your application requires.

Your plan should include these tasks:

- Determine the purpose of the key flexfield.
- Identify the number and length of its available segment columns.
- Identify if your key flexfield permits more than one structure.
- Identify if more than one structure must be defined.
- Identify the number, order and length of your segments for each structure.

Before You Begin

Once you have identified a flexfield, plan its 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're provisioned with the roles required to view and configure the flexfield. Use the Highlight Flexfields command in the Administration menu to view the run time page where the flexfield appears. Plan how you deploy the flexfield for test and production users and review the tools and tasks available for managing flexfields.

If you plan to use value sets, create them before configuring the key flexfield. You can't create value sets for key flexfields at the time that you add and configure key flexfield segments.

Access to Flexfield-Related Tasks

To configure flexfields and value sets, you must have access to the tasks for managing flexfields. Contact your security administrator for details. For information about product-specific flexfield tasks, such as Manage Fixed Assets Key Flexfields, refer to the product-specific documentation.

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. A 3-character value set with one thousand available values provides more room for changes than a 2-character value set with 100 available values.

Note the code name of the flexfield you intend to configure so that you find it easily in the tasks for managing key flexfields. In some cases you can configure how the flexfield appears on the page. See product-specific documentation to determine any restrictions on using product-specific key flexfields.

Note: Table-validated value sets with ID columns and translated independent or dependent value sets aren't supported in key flexfields.

Reporting

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. You can categorize and report on smaller discrete units of information.

Related Topics

- Considerations for Planning Value Sets

Considerations for Managing Key Flexfields

Consider the plans for a key flexfield, security, and resulting run time pages when configuring key flexfields.

Planning

Plan structures carefully and enable them 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 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.

Identify 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 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 permits 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 account combination IDs (account combination).

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, only a part of the configuration is likely to be visible. This enables the key flexfield to act more like a descriptive flexfield.

An account combination maintenance page or combinations page presents the combinations table. This enables directly creating and maintaining account 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 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, 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, 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 options for selecting combinations.

Related Topics

- Key Flexfield Structures
- Example of a Key Flexfields
- Key Flexfield Structure Instances and Segment Instances
- Considerations for Enabling Key Flexfield Segments for Business Intelligence
- Flexfield Usages

Key Flexfield Structures

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 can't 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.

Related Topics

- Key Flexfield Structure Instances and Segment Instances

Key Flexfield Structure Instances and Segment Instances

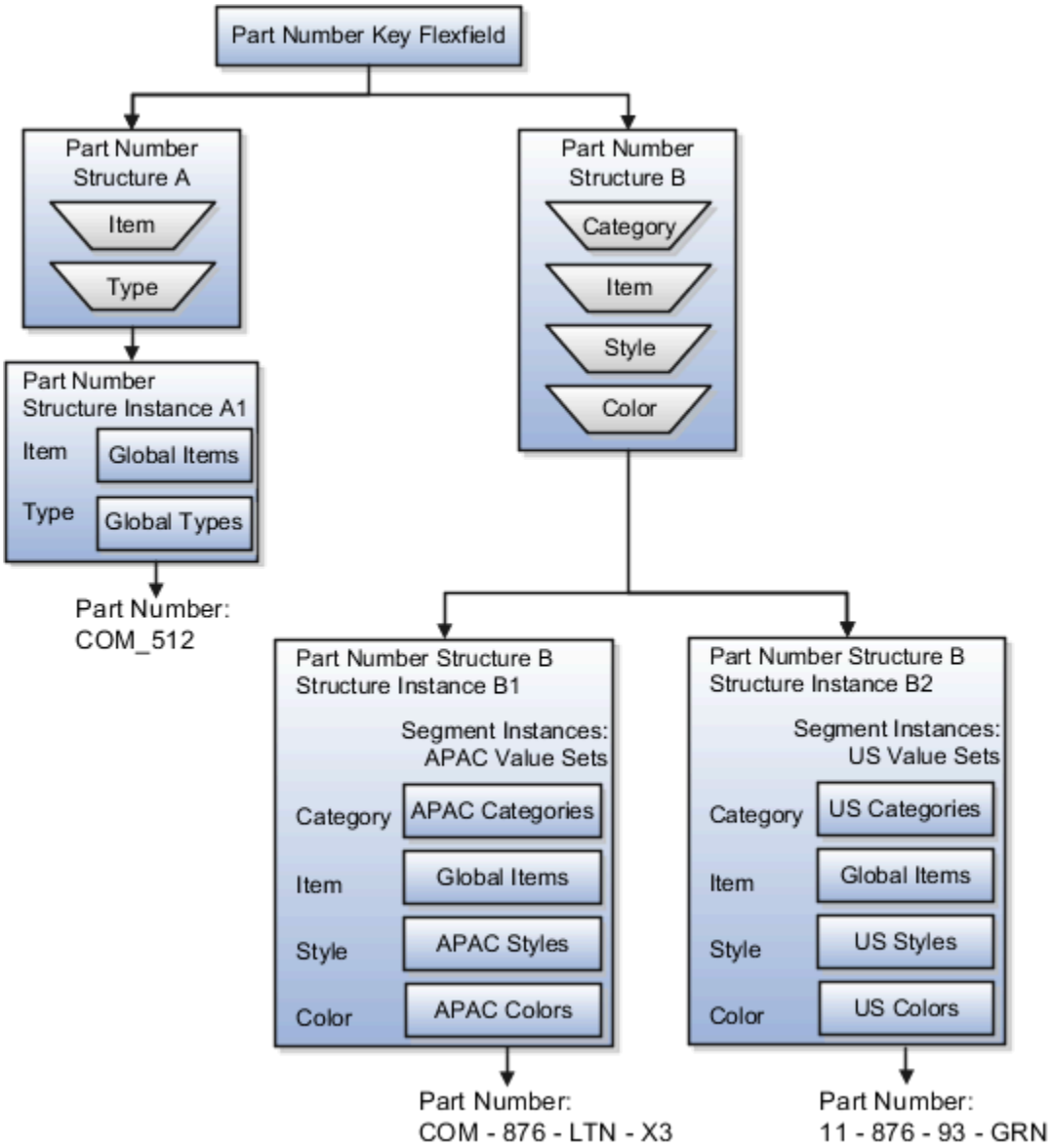
A key flexfield structure can have one or more alternate structure instances. The instances of a key flexfield structure share the following aspects of the structure.

- 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 permitted. 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 structures differ in the number of segments and the segment separators used. The structure instances share all the 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 a query for making it a selectively required attribute. A user can use it as a key flexfield combination search. On the Manage Key Flexfields UI page, if you indicate 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. A 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 query 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, you can assign the same or different tree codes to the different segment instances that use the same value set.

Related Topics

- Flexfield Segment Properties

Cross-Validation Rules

You can define cross-validation rules across segments to control the creation of key flexfield code combinations. These rules enforce whether a value of a particular segment can be combined with specific values of other segments to form a new combination.

The following 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 can'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 can't 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

The following table contains definitions used in cross-validation rules:

Rule Definition	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 in 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's 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.

Related Topics

- Considerations for Cross-Validation Rules

Considerations for Cross-Validation Rules

To validate the key flexfield combinations of segment values across segments, optimize your cross-validation rules to improve the experience of administrators and 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 logical order: If the condition filter is satisfied, then apply the validation filter.

The condition filter describes the event when the rule is evaluated. If the event specified in the condition filter isn't applicable, then the rule isn't evaluated, even if enabled. 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, your organization has determined that a certain company value called Operations can't use a specific cost center called 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.

Such a 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, it becomes difficult to interpret cross-validation error messages and rectify invalid key flexfield combinations.

Maintenance

To maintain consistent validation, review existing key flexfields when you update your cross-validation rules. Regardless of your current validation rules, you can use an existing key flexfield combination if it's 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.

To prevent users from using existing combinations that are no longer valid according to your cross-validation rules, disable them using the combinations page.

Related Topics

- Cross-Validation Rules

Edit a Cross-Validation Rule

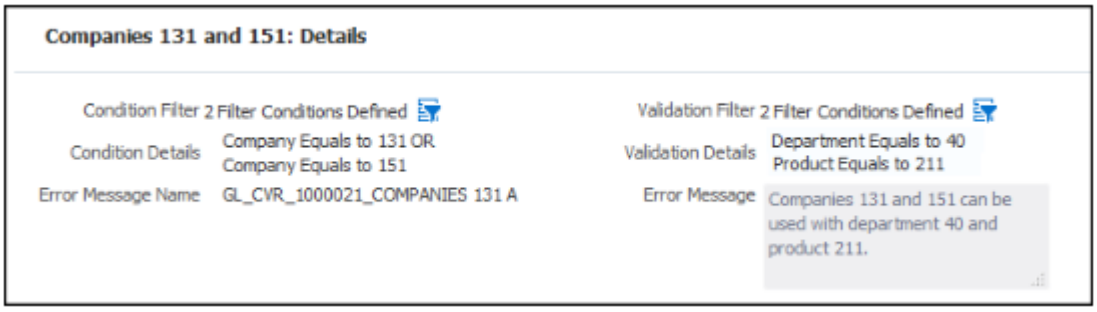
Cross-validation rules prevent specific combinations of segment values in account combinations. You can use the Manage Cross-Validation Rules task to edit existing rules or to create one-off rules.

Scenario

Your organization has a cross-validation rule named **Companies 131 and 151**, which restricts account combinations for those companies to department 40 and product 211. Account combinations for both companies should now include department 30. To edit the cross-validation rule, perform these steps.

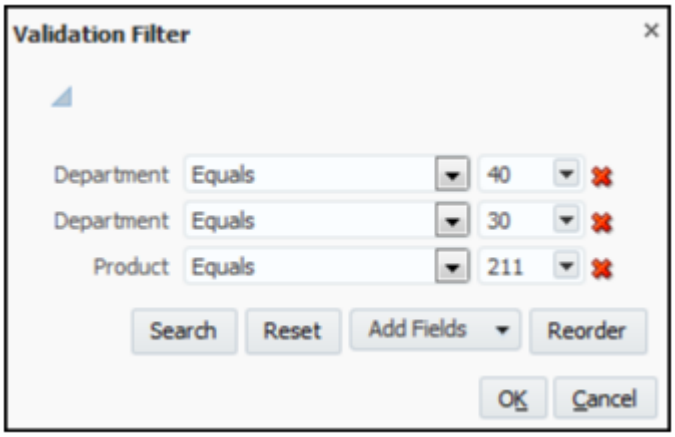
1. In the Setup and Maintenance work area, go to the following:
    - Offering: Financials
    - Functional Area: Financial Reporting Structures
    - Task: Manage Cross-Validation Rules
2. On the Rules tab, select the chart of accounts for your organization and select the **Companies 131 and 151** cross-validation rule.

    The following figure shows the section of the Edit Cross-Validation Rules page with the condition and validation filter details for companies 131 and 151. A condition is defined for company values equal to 131 or 151, and the validation specifies the department value equals 40 and the product value equals 211.



3. Click the **Validation Filter** icon.
4. Click **Add Fields** and select the **Department** segment.
5. Accept the default operator, which is Equals, and select department **30**.

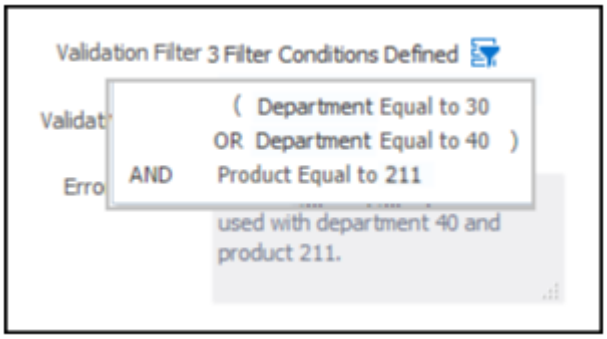
    The following figure shows the Validation Filter window with three validations: department equals 40, department equals 30, and product equals 211.



6. Click **OK**.

7. Click **Save**.

   The following figure shows the details for the updated validation on the Edit Cross-Validation Rules page. The validation specifies departments equal to 30 or 40, and the product equal to 211.



8. To update the error message, search for and select the Manage Messages for General Ledger task. Query the error message name for the cross-validation rule and edit the message to include department 30.

Related Topics

- How do I update existing setup data?

Considerations for Enabling Key Flexfield Segments for Business Intelligence

A key flexfield registered in the database as enabled for Oracle Transactional Business Intelligence (OTBI) 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 Transactional 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 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 information about logical objects and import, see the Creating and Administering Analytics and Reports guide for your products.

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 can't 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 Transactional Business Intelligence repository in Oracle Cloud implementations, run the Import Oracle Data Extensions for Transactional Business Intelligence process. For information about logical objects and import, see the Creating and Administering Analytics and Reports guide for your products.

> **Note:** When you import a flexfield into the Oracle Business Intelligence repository, you see both `<name>_` and `<name>_c` attributes for each segment, along with some other optional attributes. The `<name>_` attribute contains the value. The `<name>_c` attribute contains the code of the value set that the value comes from, and is used for linking to the value dimension. You must import both attributes.

Related Topics

- Considerations for Enabling Descriptive Flexfield Segments for Business Intelligence
- Overview of Key Flexfields
- Enable Key Flexfields for Business Intelligence Reporting
- Overview of Flexfield Change Import

Example of a Key Flexfields

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.

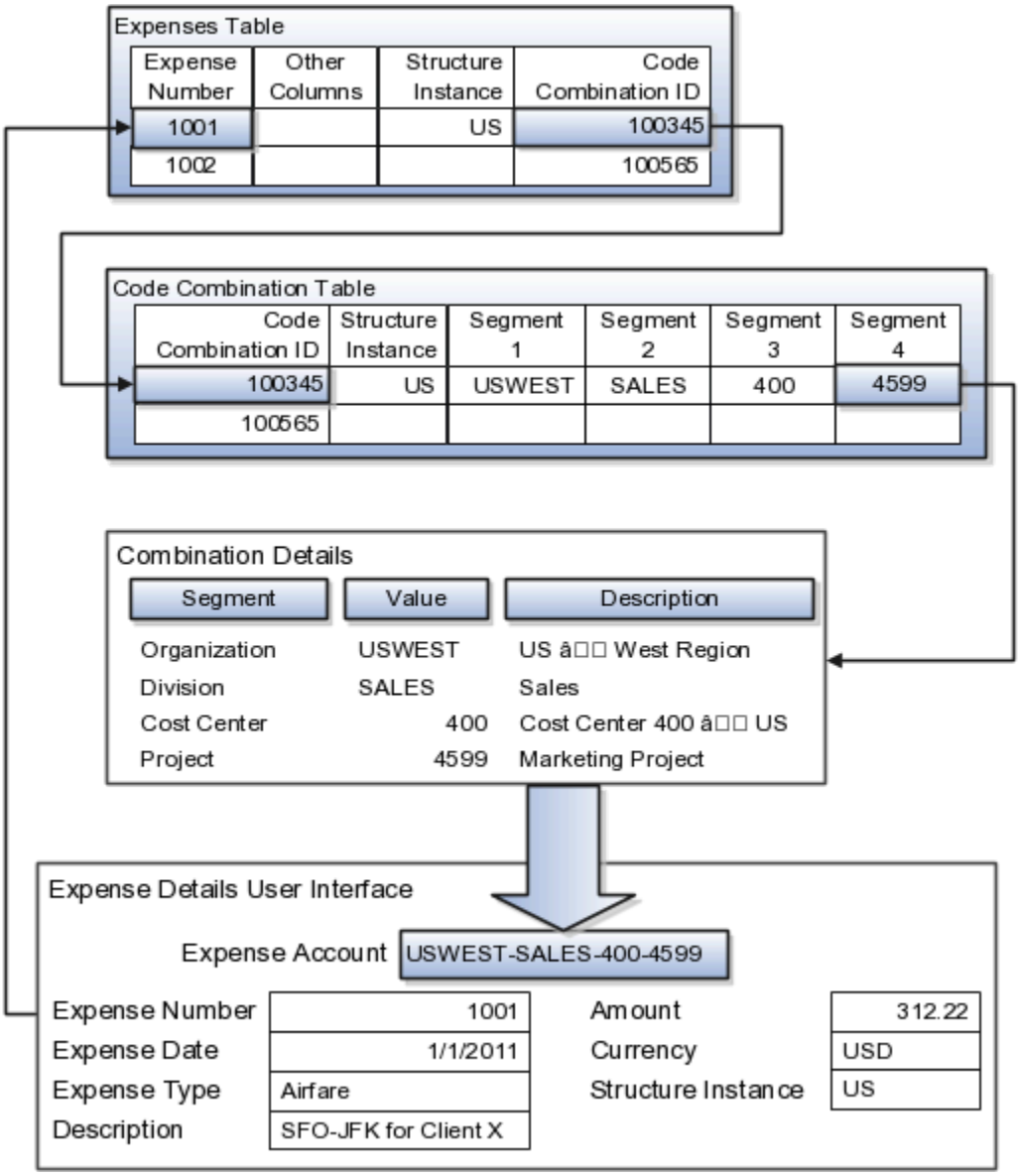
Entering Expense Accounts

A user interface for entering expenses helps the user select an expense account that identifies the cost center and other details needed for processing the expense.

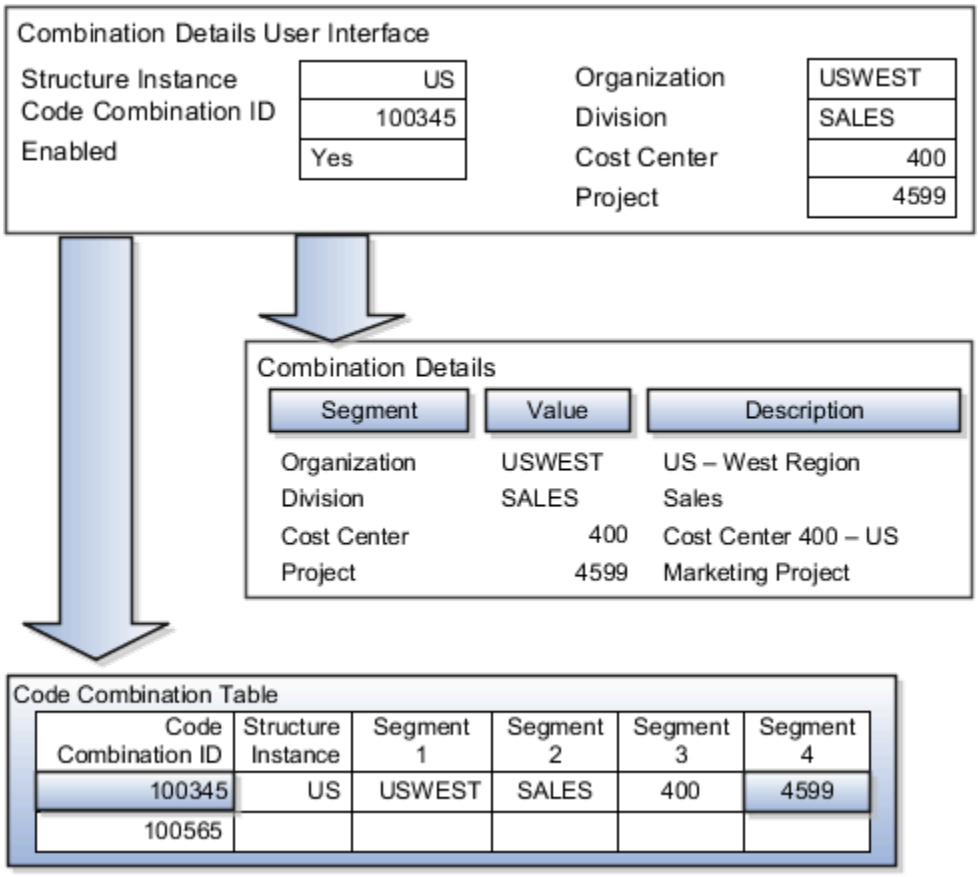
The expense account field is a foreign key reference to a account combination (EXPENSE_LINES.EXPENSE_ACCOUNT = ACCOUNT.COMBINATION).

Account combinations Table for Entering Accounts and Employees

The account combinations table supports entering account information, such as for expense accounts.



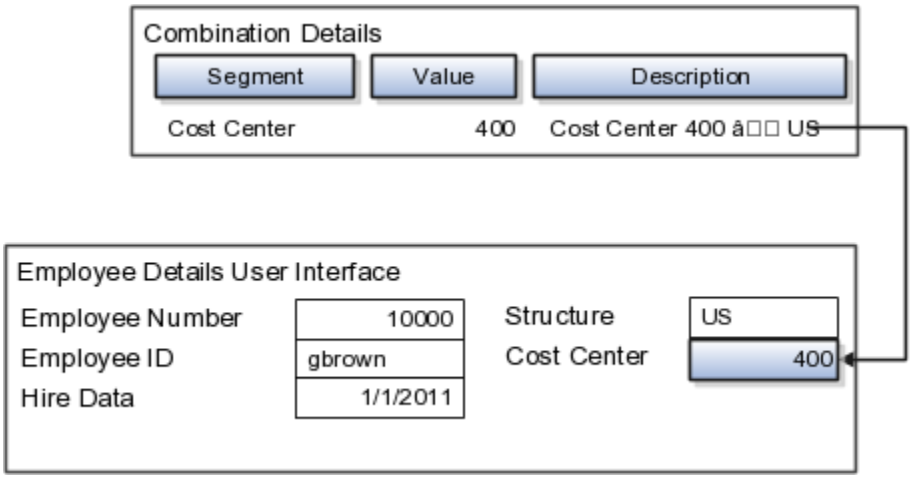
The combinations page, which is the maintenance page for the key flexfield, is for managing rows in the combinations table. In this example, managing the combinations means adding or editing account numbers that adhere to the key flexfield metadata rules.



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 preexisting account, or type in a new account that's 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.



Related Topics

- Example of One Chart of Accounts Structure with Many Instances

8 Configure Navigation

Get Started with Configuring Navigation

In the Structure work area, you can set up these elements used for navigating within the application.

Configurable Element	What It's Used For	Steps to Configure
Group	Organizes page entries and quick actions in a specific category. For example, to meet specific needs of your organization, you can create a group with page entries that open external web pages.	*Create and Edit Groups*
Page Entry	Opens a specific work area or page. For example, you can create a page entry and link it to one of your application pages.	*Create and Edit Page Entries*
Quick Action	Helps you initiate tasks quickly. For example, to access key tasks efficiently, you can add more quick actions or edit the existing ones.	*Create and Edit Quick Actions*

Here's what else you can do to configure navigation.

- Reorder navigation entries.
- Delete custom groups and page entries.

  Note: You can't delete quick actions.

- Create duplicate versions of predefined page entries and quick actions, then edit the duplicates.

Create and Edit Groups

1. In Sandboxes, create a sandbox or enter an existing one. Make sure the sandbox has the Structure tool.
2. From the sandbox **Tools** menu, select **Structure**.

3. Do one of these steps based on your need and proceed to Step 4:
    - Select **Create**, and then select **Create Group**.
    - Select the name of a group to edit.
4. Provide a name for the group and select an icon for it.
5. From the **Show on Navigator** list, choose whether to display the group on the Navigator and home page.

   If you select **EL Expression**, the evaluation of the EL expression decides whether the group will be visible for specific user roles. Select **Edit** next to the list and enter an expression without line breaks or double quotes. Validate the EL expression and make sure it evaluates to true or false.
6. Save the group and publish the sandbox.

Create and Edit Page Entries

1. In Sandboxes, create a sandbox or enter an existing one. Make sure the sandbox has the Structure tool.
2. From the sandbox **Tools** menu, select **Structure**.
3. Do one of these steps based on your need and proceed to Step 4:
    - Select **Create**, and select **Create Page Entry**.
    - Select the name of a page entry to edit.
    - Select the name of a page entry and then select **Create Duplicate** to duplicate it.
4. Provide a name for the page entry.
5. Search for and select an icon for the page entry.
6. Select the group you want to put the page entry in.
7. From the **Show on Navigator** list, choose whether to display the page entry on the Navigator and home page. Then, from the **Show on Springboard** list, choose whether to display the page entry on the home page.

   If you select **EL Expression**, the evaluation of the EL expression decides whether the page entry will be visible for specific user roles. Select **Edit** next to the list and enter an expression without line breaks or double quotes. Validate the EL expression and make sure it evaluates to true or false.

   > **Note:** If you select **EL Expression** for **Show on Navigator**, the EL expression must evaluate to true for the page entry to appear on the home page. If it evaluates to false, the entry won't be visible on the home page even if **Show on Springboard** is set to **Yes** or **EL Expression**. In other words, the springboard setting is used only if the corresponding Navigator EL expression evaluates to true.
8. From the **Mobile Enabled** list, specify whether the page entry will be available for mobile devices.
9. Select a link type and configure the link details. See *Configure Links for Page Entries and Quick Actions*.

   > **Note:** When you're editing a page entry, you can change the link configuration only for custom page entries.
10. Save the page entry and publish the sandbox.

Edit Page Entries with Tabs or Panel Tabs

Some pages have tabs or panel tabs, which you can edit using the Edit Page Entry page.

1. In Sandboxes, create a sandbox or enter an existing one. Make sure the sandbox has the Structure tool.

2. From the sandbox **Tools** menu, select **Structure**.
3. Select the name of the page entry you want to edit to modify its tabs or panel tabs.
4. Open the Tabs or Panel Tabs tab.
5. Select a row and make your changes.
    - Provide a name and an icon.
    - From the **Visible** list, choose whether to display the tab or the panel tab on the page.

      If you select **EL Expression**, the evaluation of the EL expression decides whether the tab or panel tab will be visible for specific user roles. Select **Edit** next to the list and enter an expression without line breaks or double quotes. Validate the EL expression and make sure it evaluates to true or false.
6. Use the **Move Up** and **Move Down** icons to adjust the relative position of the tabs or the panel tabs within the page.
7. Save the changes and publish the sandbox.

Create and Edit Quick Actions

For groups with associated quick actions, let's take a look at how you can create and edit quick actions. You can also edit quick action groups, which are used to group some related quick actions.

1. In Sandboxes, create a sandbox or enter an existing one. Make sure the sandbox has the Structure tool.
2. From the sandbox **Tools** menu, select **Structure**.
3. Select the name of any group that has quick actions.
4. On the Edit Group page, select the Quick Actions tab.
5. Do one of these steps based on your need and proceed to Step 6:
    - Select **Create**.
    - Expand a quick action group and select a quick action name to edit.
    - Expand a quick action group, select a quick action name and then select **Create Duplicate** to duplicate it.
6. Provide a name for the quick action.
7. Search for and select an image for the quick action.
8. Select the quick action group you want to place this quick action in.
9. (Only for editing a quick action) If you want the quick action to appear in the quick action list on the home page, select the **Show as quick action link on home** checkbox.
10. From the **Visible** list, choose whether to display the quick action on the home page.

    If you select **EL Expression**, the evaluation of the EL expression decides whether the quick action will be visible for specific user roles. Select **Edit** next to the list and enter an expression without line breaks or double quotes. Validate the EL expression and make sure it evaluates to true or false.
11. From the **Mobile Enabled** list, specify whether the quick action will be available for mobile devices.
12. Select a link type and configure the link details. See *Configure Links for Page Entries and Quick Actions*.

    > **Note:** When you're editing a quick action, you can change the link configuration only for custom quick actions.

13. Save the quick action and publish the sandbox.

Edit Quick Action Groups

1. In Sandboxes, create a sandbox or enter an existing one. Make sure the sandbox has the Structure tool.
2. From the sandbox **Tools** menu, select **Structure**.
3. Select the name of any group that has quick actions.
4. On the Edit Group page, select the Quick Actions tab.
5. Select the name of the quick action group.
6. Rename the group.
7. From the **Visible** list, choose whether to display the quick action group and its quick actions on the home page.

   If you select **EL Expression**, the evaluation of the EL expression decides whether the quick action group will be visible for specific user roles. Select **Edit** next to the list and enter an expression without line breaks or double quotes. Validate the EL expression and make sure it evaluates to true or false.
8. Save the changes and publish the sandbox.

Configure Links for Page Entries and Quick Actions

When you create or edit a page entry or quick action, you can select the link type to determine the destination for users. Dynamic URL type is available for page entries only.

Link Type	Description
Application Page	Any page within your application built with Oracle Application Development Framework (ADF).
Dynamic URL	The URL of an external page where the host, port, or context root might change. You can find the host and port details for a dynamic URL from a lookup based on the application name.
Static URL	The URL of an external page where the host, port, or context root doesn't change. Static URLs don't require lookups.
Static URL with secure destination	The URL for a partner application where secure tokens are appended. These tokens carry user identity and security information, enabling authentication without requiring users to enter credentials, such as user name and password.
VB Studio Page	A page built using Visual Builder Studio and Oracle Java Extension Toolkit.

Application Page

When you create or edit a page entry or quick action, you can link it to one of your application pages.

1. From the **Link Type** list, select **Application Page**.

2. Provide details for the fields.

Field	Description
Task Flow ID (for a quick action)	Enter the task flow ID.
Focus View ID	Enter the view ID attribute value associated with the target page. You can find it in the **`public_html/WEBINF/adfc-config.xml`** file of this application. **Note:** For a quick action, by default, the focus view ID is displayed as **`/FndOverview`**.
Web Application	Select the name of the web application, which is the application name that you used while creating this application using the Setup and Maintenance work area.
Secured Resource Name and Application Stripe	For secured access to the target application page, provide the secured resource name and the name of the policy store's application stripe. Here's an example of a secured resource name: **`oracle.apps.view.pageDefs.CaseList_Form_Attach_UIShellPagePageDef`** When a user clicks the link, the application checks the secured resource and the Oracle Platform Security Services (OPSS) 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`**.
○ Page Parameters List (for a page entry) ○ Task Flow Parameters (for a quick action)	If the page takes parameters or task flow parameters, then you can enter a semicolon-delimited string of name=value pairs (for example, **`org=m1;context=s1`**) in these fields. You can use 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 group on the home page or Navigator, the parameter might be set to **`status=Open`**, whereas if you're opening the page from another group, the parameter might be set to **`status=Closed`**.

Dynamic URL

You can link a page entry to an external website 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 registration. This change affects all page entries that contain dynamic links pointing to that web application.

Suppose you want to link to a test version of an application; you use the dynamic URL link type. When you move the application from the test to the production environment, you just change the host and port details of the web application registration.

Before you start, make sure you add the application details on the Manage Integration of Additional Applications page to register the web application. These application details are used for configuring dynamic URL links for page entries.

1. In Setup and Maintenance, click , and then click **Manage Setup Content**.
2. Click **Manage Integration of Additional Applications**, click **Actions**, and then click **Create**.
3. Enter the application details in the relevant fields. Ensure that the information you provide is complete and accurate. For example,
   - Begin the URL with `http://` or `https://`.
   - Provide correct access credentials for the selected security policy.

   Note: You can't edit the **Application Name** and **Partner Name** once you save the details.
4. Click **Save and Close**.

To link a page entry to a dynamic URL:

1. From the **Link Type** list, select **Dynamic URL**.
2. From the **Web Application** list, select the web application name and enter the destination for the web application.

   Let's look at an example:

   Suppose you want to link to a complete URL, `http://example:9011/myApp/faces/Page1`.

   a. From the **Web Application** list, select **myApp**. This list displays the application names that were added using the Manage Integration of Additional Applications page. Once you select the application name, the full URL that was added for this application (for example, `http://example:9011/myApp/`) is automatically used as the start of the dynamic URL.
   b. Enter the destination for the web application, for example, `/faces/Page1`. This value is appended to the full URL.

When users click this page entry, the target page opens in a new browser window or tab.

Static URL

When you create or edit a page entry or quick action, you can link it to an external website or application that has a constant host, port, or context root.

1. From the **Link Type** list, select **Static URL**.
2. Enter the URL destination. The URL must start with `http://` or `https://`.

   For example, you can use a static URL to link to `http://www.oracle.com`.

When users click this page entry or quick action, the target page opens in a new browser window or tab.

Static URL with Secure Destination

When you create or edit a page entry or quick action, you can link it to a partner application (external) using a secure token URL.

1. From the **Link Type** list, select the **Static URL**.
2. Click and select **Secure Destination**.
3. From the **Web Application** list, select the name of the web application.

4. Enter the destination for the web application. An HTTPS protocol is required to access the application.
5. Enter the name of the secure token. All secure tokens have a predefined lifetime, and they expire after that duration. So, users must refresh the page to regenerate the tokens.

The partner application validates the secure token and uses it to authenticate web services within the user context. Using this mode of modified access, a partner can directly do a task or display information to the specific user without any additional authentication. When users click this page entry or quick action, the target page opens in a new browser window or tab.

VB Studio Page

When you create or edit a page entry or quick action, you can link it to one of your VB Studio pages.

> **Tip:** If an existing page entry or quick action links to the same page that you want your new page entry or quick action to link to, you can enter the same details for all fields, except parameters. View the existing details, such as Focus View ID, Web Application, and Secured Resource Name, and fill in the fields for the new page entry or quick action.

1. From the **Link Type** list, select **VB Studio Page**.
2. Provide details for the fields.

Field	Description
Focus View ID	Enter the view ID attribute value associated with the target page. You can find it in the `public_html/WEBINF/adfc-config.xml` file of this application. - If you're linking to VB apps that have a URL starting with `fscmUI/redwood/`, then you enter the focus view ID as `/index.html`. For other VB apps, the focus view ID isn't required. But, because it's a required field on the UI, you need to enter the value as `/index.html`. - If you're linking to Fusion Applications, enter the focus view ID as `/FndOverview`. For a quick action, by default, the focus view ID is displayed as `/index.html`.
Web Application	Select the name of the web application, which is the application name that you used while creating this application using the Setup and Maintenance work area. If you're linking to VB apps that have a URL starting with `fscmUI/redwood`, select `ORA_FSCM UI`. For other VB apps, select one of these: `ORA FSCM UI`, `ORA HCM UI`, or `ORA CRM UI` .
Secured Resource Name and Application Stripe	For secured access to the target application page, provide the secured resource name and the application stripe name of the policy store. When users click the application link, the application checks the secured resource and the Oracle Platform Security Services (OPSS) policy store and then determines whether the user has the privilege to view the page. Here's an example of a secured resource name: `oracle.apps.view.pageDefs.CaseList_Form_Attach_UIShellPagePageDef`

Field	Description
	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**. **Tip:** To secure the custom page entry in the same way as an existing page entry, copy and paste the secured resource name value from a predefined page entry. Similarly, for a custom quick action, copy from a predefined quick action. You can't create a new resource.
VB Studio Flow	Enter the VB Studio flow that your VB page is grouped in.
Product Family	Enter the name of the product family your VB Studio page belongs to.
○ VB App UI ○ VB Page Name	If you're linking to VB apps that have a URL starting with **fscmUI/redwood**, fill out these fields: ○ VB App UI: Enter the web application UI name, which includes a UI component in the form of VB pages and flows. For example, **/digital-sales**. ○ VB Page Name: Enter the UI component that users interact with. For example, **/p/mycustomers/salesday**.
Page Parameters List	If the page takes parameters, then you can enter a semicolon-delimited string of name=value pairs (for example, **org=m1;context=s1**) in this field. You can use 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 group on the home page or Navigator, the parameter might be set to **status=Open**, whereas if you're opening the page from another group, the parameter might be set to **status=Closed**.

Reorder Navigation Entries

You can reorder groups, page entries, quick action groups, and quick actions. You can also move page entries and quick actions to different groups.

Reorder Groups and Page Entries

1. In Sandboxes, create a sandbox or enter an existing one. Make sure the sandbox has the Structure tool.
2. From the sandbox Tools menu, select **Structure**.
3. You can do these reordering tasks using the icons in the Order column.
    - Use the **Move Up** and **Move Down** icons to reorder groups and page entries.
    - Use the **Move To** icon to move page entries to different groups or to the top level.

4. Publish the sandbox.

Reorder Quick Actions

1. In Sandboxes, create a sandbox or enter an existing one. Make sure the sandbox has the Structure tool.
2. From the sandbox Tools menu, select **Structure**.
3. Click any group that has quick actions enabled.
4. On the Edit Group page, click the Quick Actions tab. You can do these reordering tasks using the icons in the Order column.
    - Use the **Move Up** or **Move Down** icons to reorder a quick action group.
    - Expand the group and use the **Move Up** or **Move Down** icons to reorder quick actions within a group.
    - Expand the group, and use the **Move To** icon to move a quick action to a different quick action group. On the Move To: Group dialog box, click the quick action group you want to move the quick action to.

    > **Note:** The reordering of quick actions within a quick action group affects only how the quick actions are displayed for your users on the home page or product map. But when the users view all the quick actions, they're alphabetically sorted, and the reordering isn't reflected there. In News feed, they're sorted under their respective groups.

5. Save the changes and publish the sandbox.

Delete Custom Groups and Page Entries

To delete custom groups or page entries, follow these steps:

1. In Sandboxes, create a sandbox or enter an existing one. Make sure the sandbox has the Structure tool.
2. From the sandbox Tools menu, select **Structure**.
3. Click the name of the group or page entry you want to delete.
4. On the Edit Group page or the Edit Page Entry page, click **Delete**.
5. Publish the sandbox.

What are the supported and unsupported EL expressions?

Use EL expressions to control the visibility of navigation entries such as groups, page entries, and quick actions, as well as infolets and announcements on the UI. You can tailor the visibility based on user roles. For example, you can use EL expressions to show or hide a group or page entry on the Navigator or product map. The evaluation of the EL expression decides whether these entries are displayed for users with specific roles.

Note:

- Don't include line breaks or double quotes in EL expressions.
- Don't copy EL expressions from Word into the expression editor, as hidden characters might cause errors. Use a text editor or type them manually instead.
- If you have custom menu entries in your environment that use an EL expression that checks for specific roles to access the custom menu, then the access might be limited to those menu entries. To access other custom menu entries, create a custom role with the necessary privileges using Security Console, and assign to the user.

Supported EL Expressions

Use this table to see examples of EL expressions that are supported on Redwood pages.

EL Expression Properties	Examples
• securityContext.regionViewable • securityContext.taskflowViewable • securityContext.userGrantedPermissic • securityContext.userGrantedResource • securityContext.userInRole • securityContext.userName • securityContext.userInAllRoles • securityContext.authenticated	• #{securityContext.regionViewable['oracle.apps.financials.assets.adjustments.desktopFinancial.di.page.Mass • #{securityContext.taskflowViewable['/WEB-INF/oracle/apps/financials/assets/adjustments/workArea/ui/flow/AdjustmentCapMainAreaTaskFlow.xml#AdjustmentCapMainAreaTaskFlow']} • #{securityContext.userGrantedPermission['permissionClass=oracle.adf.share.security.authorization.MethodI PREPARE_FIXED_ASSET_UNITS_OF_PRODUCTION_PERMISSION;action=invoke']} • #{securityContext.userGrantedPermission['oracle.adf.controller.security.TaskFlowPermission;target=/WEB-INF/oracle/apps/hcm/compensation/variableCompensation/allocations/publicUi/flow/VariableCompensationDashboard.xml#VariableCompensationDashboard;action=view']} • #{securityContext.userGrantedResource['resourceType=FNDResourceType;resourceName=FND_Scheduled_Processes_Menu;action=launch']} • #{(!securityContext.userInRole['GSE_HR_DASHBOARD_UHP']} • #{((securityContext.userName == '100472' or securityContext.userName == '107612')} • #{securityContext.userInAllRoles['LIVIO_ADMIN_DATA'] } • #{securityContext.authenticated and securityContext.userGrantedResource['resourceName=FND_Administration_Menu;resourceType=FNDResourceType;action=launch']}
• Profile.values	• #{Profile.values.PAY_PAYROLL_RESPONSIVE_ENABLED eq 'Y'}
• EndPointProvider.externalEndpointBy • EndPointProvider.externalEndpointBy	• #{(EndPointProvider.externalEndpointByAppShortName['ORA_SAAS_APP'] } • #{EndPointProvider.externalEndpointByModuleShortName['ORA_FSCM_UI']}
• applCorePrefs.userName • applCorePrefs.userDisplayName • applCorePrefs.locale • applCorePrefs.accessibilityMode	• #{applCorePrefs.userName ne null && applCorePrefs.userName eq 'HELPADMIN'} • #{applCorePrefs.userDisplayName} • #{EndPointProvider.externalEndpointByModuleShortName['TALEORECRUITMENTDASHBOARD']}?locale=#{applCorePrefs.locale}

EL Expression Properties	Examples
	• #{applCorePrefs.accessibilityMode}
• Offerings.implementationStatus • Offerings.isFeatureEnabledForImplem • Offerings.isFeatureChoiceEnabledForI • Offerings.isFeatureObsolete	• #{Offerings.implementationStatus.PER_CORE== 'IMPLEMENTED' or Offerings.implementationStatus.PER_CORE== 'IN_PROGRESS_FEATURE'} • #{Offerings.isFeatureEnabledForImplementation.PER_WKF_DEPL or Offerings.isFeatureEnabledForImplementation.PER_WKF_DEV or Offerings.isFeatureEnabledForImplementation.PER_CORE} • #{Offerings.isFeatureChoiceEnabledForImplementation['ORA_SVC_INTERNAL_SERVICE_REQUESTS'] or Offerings.isFeatureChoiceEnabledForImplementation['ORA_SVC_HR_SERVICE_REQUESTS']} • #{(Offerings.isFeatureObsolete.ORA_INV_REDWOOD_SUBS_LOCATORS)}
• ApplcoreCommonBean.anonymousUs • ApplcoreCommonBean.impersonation	• #{(ApplcoreCommonBean.anonymousUser)} • #{!ApplcoreCommonBean.impersonationSession}
• applCoreSecuredToken.trustToken	• #{values.SVC_ENGAGEMENT_DASHBOARD_URL}?atkToken=#{applCoreSecuredToken.trustToken}
• HcmFusePlusBean.userPersonId • HcmFusePlusBean.userPrimaryAssign • HcmFusePlusBean.loggedOnUserExcl • HcmFusePlusBean.enterpriseId • HcmFusePlusBean.userId • HcmFusePlusBean.userPartyId	• #{Profile.values.ORA_HCM_VBCS_PWA_ENABLED eq 'Y' and Profile.values.ORA_PER_ADDITIONAL_PERSONAL_INFO_REDWOOD_ENABLED eq 'Y' and HcmFusePlusBean.userPersonId ne null and false} • #{Profile.values.PER_DIRECTORY_RESPONSIVE_ENABLED eq 'Y' and Profile.values.HCM_RESPONSIVE_PAGES_ENABLED eq 'Y' and HcmFusePlusBean.userPrimaryAssignmentId ne null} • #{!(HcmFusePlusBean.loggedOnUserExcludedForIndexNameMap['fa-hcm-person']) and Profile.values.ORA_HCM_VBCS_PWA_ENABLED eq 'Y' and Profile.values.ORA_PER_EMPLOYMENT_INFO_REDWOOD_ENABLED eq 'Y' } • #{not(HcmFusePlusBean.enterpriseId eq Profile.values.ORA_HRT_PROFILE_DATA_MIGRATED or HcmFusePlusBean.enterpriseId eq Profile.values.ORA_HRT_ADDITIONAL_V2_ENTERPRISE)}
• mscAskOracleBean.SOPPCWPlan.statu • mscAskOracleBean.PCPCWPlan.status • mscAskOracleBean.REPLPCWPlan.sta • mscAskOracleBean.DMPCWPlan.statu • mscAskOracleBean.SPPCWPlan.status • mscAskOracleBean.DSPCWPlan.status	• #{mscAskOracleBean.DMPCWPlan.status} • #{mscAskOracleBean.DSPCWPlan.status}

Unsupported EL Expressions and their Alternatives

Use this table to see examples of unsupported EL expressions and their alternatives.

EL Expression Properties	Examples	Alternative EL Expression
• SystemInfo.podType.HCM	• #{SystemInfo.podType.HCM or SystemInfo.offering.WorkforceDeployment	• #{ EndPointProvider.externalEndpointByModuleShor HCM_UI'] !=null}

EL Expression Properties	Examples	Alternative EL Expression
• SystemInfo.offering.CompensationManage	• #{SystemInfo.podType.HCM or SystemInfo.offering.CompensationManage	• #{Offerings.implementationStatus.PER_CORE== 'IMPLEMENTED' or Offerings.implementationStatus.PER_CORE== 'IN_PROGRESS_FEATURE'}
• SystemInfo.offering.WorkforceDeployment	• #{(SystemInfo.podType.HCM or SystemInfo.offering.WorkforceDeployment or SystemInfo.offering.WorkforceDevelopmen or SystemInfo.offering.CompensationManage and (securityContext.userInRole['GSE_HCM_COMPENSATION_UHP'])}	• #{Offerings.implementationStatus.PER_WKF_DEPL== 'IMPLEMENTED' or Offerings.implementationStatus.PER_WKF_DEPL== 'IN_PROGRESS_FEATURE'}
• SystemInfo.offering.WorkforceDevelopmen	• #{(SystemInfo.podType.HCM or SystemInfo.offering.WorkforceDeployment or SystemInfo.offering.WorkforceDevelopmen or SystemInfo.offering.CompensationManage and (securityContext.userInRole['GSE_HCM_COMPENSATION_UHP'])}	• #{Offerings.implementationStatus.PER_WKF_DEV== 'IMPLEMENTED' or Offerings.implementationStatus.PER_WKF_DEV== 'IN_PROGRESS_FEATURE'}
• facesContext.externalContext.request.serve	• {BASE_URL}/{PATH}?username=#%7BsecurityContext.userName%7D%26amp;host=https://%23%7BfacesContext.externalContext.requ%7D%26amp;jwt=%23%7BapplCoreSecuredToken.trustToken%7D%26amp;jwt=%23%7BapplCoreSecuredToken.trustToken%7D	• #{EndPointProvider.externalEndpointByAppShortNa FSCM_UIAPP']=={BASE_URL}
• facesContext.externalContext.requestHead	• #{(facesContext.externalContext.requestHe == 'fa-emvg-dev1-saasfaprod1.fa.ocs.oraclecloud.com' or facesContext.externalContext.requestHead == 'fa-emvg-dev2-saasfaprod1.fa.ocs.oraclecloud.com')}	• #{EndPointProvider.externalEndpointByAppShortNa FSCM_UIAPP']=={BASE_URL}
• SystemInfo.offering.Sales	• #{(SystemInfo.podType.CRM or SystemInfo.offering.Sales)}	• #{Offerings.implementationStatus.ZBS_SALES== 'IMPLEMENTED' or Offerings.implementationStatus.ZBS_SALES== 'IN_PROGRESS_FEATURE'}
• SystemInfo.podType.CRM	• #{(SystemInfo.podType.CRM or SystemInfo.offering.Sales) and securityContext.userInRole['ZBS_SALES_ADMINISTRATOR_JOB']} #{(SystemInfo.podType.CRM or SystemInfo.offering.Sales) and (securityContext.userInRole['ZBS_SALES_ADMINISTRATOR_JOB'])}	• #{EndPointProvider.externalEndpointByModuleShor CRM_UI'] !=null}

EL Expression Properties	Examples	Alternative EL Expression
• SystemInfo.podType.ERP	• #{SystemInfo.podType.ERP or SystemInfo.podType.HCM or SystemInfo.offering.WorkforceDeployment}	• #{EndPointProvider.externalEndpointByModuleShort FSCM_UI'] !=null}

Unsupported EL Expressions without Alternatives

Use this table to see examples of unsupported EL expressions without their alternatives.

EL Expression Properties	Examples
• HcmCommonPayFuseBean.userPayLd	• #{Profile.values.PAY_PAYROLL_RESPONSIVE_ENABLED eq 'Y' and Profile.values.HCM_RESPONSIVE_PAGES_ENABLED eq 'Y' and HcmCommonPayFuseBean.userPayLdgId eq 300000047826641}
• facesContext.viewRoot.locale.languag	• #{facesContext.viewRoot.locale.language =='ar'} • #{facesContext.viewRoot.locale.language!='en' ? true : false}
• ManageSandboxesBean.unifiedSandb	• #{!(ManageSandboxesBean.unifiedSandboxEnabled) and (!securityContext.userInRole['GSE_MIGRATION_UHP'])}
• facesContext.externalContext.requestl	• {BASE_URL}/{PATH}?username=#%7BsecurityContext.userName%7D%26amp;host=https://%23%7BfacesContext.externalContext.request.serverName%7D%26amp;lang=%23%7BfacesContext.externalContext.requestLocale%7D%26amp;jwt=%23%7BapplCoreSecuredToken.trustToken%7D
• webCenterProfile	• #{WebCenterProfile[securityContext.userName].businessCountry != 'USA'}
• SetupProfileBean.coachingStatus	• #{HcmFusePlusBean.userPersonId ne null and SetupProfileBean.coachingStatus eq 'yes' and ! securityContext.userInRole['CIMB_NIAGA_EMPLOYEE_ABSTRACT_CUSTOM']}
• SupplierPortalMenuBean.implemente	• #{ SupplierPortalMenuBean.implementedContracts and (!((securityContext.userInRole['ICA_SUPPLIER_USER']))) }
• WCSecurityContext.userInAppRole	• #{!WCSecurityContext.userInAppRole['CUSTOM_GENERAL_RESTRICTION']}
• WCSecurityContext.currentUser	• #{!(WCSecurityContext.currentUser['AMY.HARTINGTON'] or WCSecurityContext.currentUser['hank.lowe'])}
• WCSecurityContext.userInScopedRole	• #{WCSecurityContext.userInScopedRole[ORA_ASM_APPLICATION_IMPLEMENTATION_CONSULTANT_JOB]}
• WCTruncator	• #{WCTruncator[securityContext.userName]['9']=='CRM.STUDE...' || WCSecurityContext.currentUser['CRM.INSTRUCTOR'] || WCSecurityContext.currentUser['CRM.INSTRUCTOR01'] || WCSecurityContext.currentUser['CRM.INSTRUCTOR02']}

Review and resolve the unsupported EL expressions by editing them. See *How do I resolve unsupported EL expressions?*

How do I resolve unsupported EL expressions?

1. In Sandboxes, create a sandbox or enter an existing one. Make sure the sandbox has the Structure tool.
2. From the sandbox **Tools** menu, select **Structure**.
3. In Structure, scan for unsupported EL expressions from the banner of the page.
4. Review and resolve the unsupported EL expressions by editing them. See *What are the supported and unsupported EL expressions?*

> **Note:** It's always recommended you resolve unsupported EL expressions to avoid potential issues in future Redwood experiences.

FAQs for Navigation

Why can't I delete certain navigation entries?

- You can delete only custom groups and page entries created using the Structure work area, not those created using any other tool.
- You can’t delete predefined groups and page entries.
- You can't delete quick actions, whether predefined or custom.

Why can't I hide or move some navigation entries?

While configuring the navigation entries, you can't do these tasks:

- Override the visibility setting for the Structure page entry and the Tools group. The default setting is **Yes**, so the Structure page entry and the Tools group always appear. You can't hide them.
- Move the Structure page entry to a different group or to the top level. The default group is Configuration, so the Structure page entry always appears in the Configuration group.

Why are certain navigation entries not available to users?

Probably due to these reasons:

- The page entry or quick action, or the group under which they're categorized might be hidden. Make them visible using the Structure work area.

- The offering associated with the group or page entry isn't enabled for the user. Enable it using the Offerings work area.
- The user might not have the required privileges. Make sure the privileges are assigned by the security administrator.

Deep Links

You can use deep links to open pages without using menus or navigating through pages. Deep links come in handy if you want users to go directly from various locations to Oracle Applications Cloud.

You can add deep links to a business intelligence report, an email notification, or a company website hosted on Oracle Cloud. For example, you can enable direct navigation to the My Team page. If users click your deep link URL from the company website, but aren't yet signed in to Oracle Applications Cloud, they're automatically redirected to the Sign In page. To see a complete list of the available deep links, use the Deep Links work area (**Navigator** > **Tools** > **Deep Links**). For access to this work area, you need to have the View Administration Link (FND_VIEW_ADMIN_LINK_PRIV) privilege.

How It Works

You can copy a URL from the Deep Links page and use it to create a link, for example on your company website. You can use deep links as is, to take users to a high level page, for example an overview or search page. Or, you can edit the URL so that it opens a page where the user can take action, for example to create something. For more information, refer to your product-specific documents.

URL Pattern of Deep Links

Here's a typical URL pattern of a deep link.

```
https://<pod-name>/fndSetup/faces/deeplink?
objType=<objID>&objKey=<name1=value1;name2=value2>&action=<action>&returnApp=<returnappURL>&returnAppParams=<returnAppPa
```

To direct users to a product-specific action page, you need to add specific parameters to the deep link URL. This table describes the parameters and gives you some examples of parameter values.

Parameter	Description	Example of Parameter Values
`pod-name`	The host name.	`pod.oraclecloud.com`
`objType`	The object type, such as work areas.	`SUPPLIER`
`objKey`	The object identifier.	`SupplierNumber`

`action`	The action that users can do on the page that the URL opens up, for example to edit something.
	
	Here are a few things to know:
	
	• If you don't add any value for the action parameter, this value is considered as `NONE` by default. So users will be taken to a
		`NONE`
		
		`VIEW`
		
		`CREATE`
		
		`EDIT`

Parameter	Description	Example of Parameter Values
	high-level page, such as an overview page or a search page. • If the value you added for the action parameter doesn't match with any action on the UI, or the user doesn't have the privileges to access that action, they will be taken to a high-level page.	
`returnAppURL`	The application URL where users will be automatically redirected to when they try to return to the previous page, or after they complete their task. If you're using a third party URL, make sure the application name is registered using the Manage Integration of Additional Applications page in the Setup and Maintenance work area.	MyOracle
`returnAppParams`	Parameters for the return application URL.	type=photo

Where to Find Information About Product-Specific Deep Links

Here's some information about specific deep links depending on the products you're using. This isn't a comprehensive list of all available deep links.

- Oracle Fusion Cloud Supply Chain & Manufacturing

  See: *Deep Links for SCM*
- Oracle Fusion Cloud Procurement

  See: *Deep Links for Procurement*
- Oracle Fusion Cloud Succession Planning

  See: *Overview of Succession Plan Deep Links*
- Oracle Fusion Cloud Financial Reporting Compliance

  See: *Deep-Link URLs*

Related Topics

- Manage Integration of Additional Applications

9 Help Content

How You Manage Different Types of Help

You can edit the help content that comes with the application. You can also add your own help to help windows, for example to provide company policies. Let's look at the different types of help that you can manage.

Help Type	How You See This Help	How You Edit or Add to It
Fields and other UI elements	Help text appears when you hover over or click certain UI elements on the page.	You can edit this text using the User Interface Text tool, or add your own help text to buttons using Page Composer.
Help windows	Many pages have help icons. Click these icons to open help windows that contain help text, links, or both.	You can use the User Interface Text tool to edit the help text. You can add or remove links by clicking the **Manage Help Content** link in the help window. But you can't create new help windows to put on the page. In some cases, the same help window appears on more than one page. So whatever you do to the help window on one page applies to the same help window in other places.
Getting Started work area	Select **Getting Started** from the Navigator to open this work area and review the videos and other information for new users.	You can edit or add to this content by clicking the **Edit Getting Started** link on these pages.

What Else You Can Do

Here's what you can do with all the help that you or other people added to help windows and the Getting Started work area:

- If you add a lot of links in help windows, you might want to manage your help in a central place. You can use the **Manage Help Content** task in the Setup and Maintenance work area to view and edit all of the help that anyone added.
- If you create help in a test environment, you can migrate all added help by exporting and then importing a configuration package.

Related Topics

- Add Your Content to a Help Window
- How do I modify text using the User Interface Text tool?
- Manage Getting Started Content
- Manage Help for Fields and Other UI Elements

Manage Help for Fields and Other UI Elements

Help text might appear when you hover over or click certain UI elements on the page. For example, you might see help text when you put your cursor in a specific field, or hover over an icon.

How You Go About It

- Use the User Interface Text tool to edit the help text for most types of UI elements, such as buttons, icons, and fields. You can also use this tool to edit the sentence or two of help text before the links 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, help for UI elements, message, and so on. You search for existing text and enter what you want to replace it with.
    - So if you need to edit the help text for a specific UI element on a specific page, it works best if the help text isn't too generic. For example, when you hover over an icon, say you currently see Create. In the User Interface Text tool, you'd search for Create and get many results. It might be hard to tell which result is for the specific icon you're working on. But if the help text is something like Create an Object, then you can search for Create an Object and most likely find the instance you want to replace with your own text.
- Use Page Composer to add help text, usually for buttons that don't have existing help text. Open the button's properties and enter your help text in the **Short Desc** field.

Related Topics

- How do I modify text using the User Interface Text tool?
- Modify UI Components Using Page Composer

Why can't I see the Manage Help Content or Edit Getting Started link?

You need the Manage Help Content (ATK_CUSTOMIZE_HELP_TOPICS_PRIV) privilege, and the Help Content Management feature must be enabled for any offering.

Without both, you can't see the **Manage Help Content** link in help windows or the **Edit Getting Started** link in the Getting Started work area.

Related Topics

- Set Up Help
- Give People Access to Create and Edit Help

Help Windows

Before You Start Adding Help Links

Here's what you or your administrators need to do so that you and others can add content to help windows.

1. In the Offerings work area, enable the Help Content Management feature at the offering level.
2. Give people access to create and edit help.
3. Do this if you want to make some help visible only to restricted sets of users:
    a. In the Offerings work area, enable the Security for Added Help feature at the offering level.
    b. Create help security groups.
4. Confirm that help icons show by default, so that users can find the content you're adding. In the Setup and Maintenance work area, use the **Set Help Options** task in the Application Extensions functional area.

Related Topics

- Configure Offerings
- Give People Access to Create and Edit Help
- Create Groups to Limit Access to Added Help
- Hide or Show Help Icons

Add Your Content to a Help Window

The easiest way to add help is to start in the help window where you want the help link to appear. You can add links to web pages, upload files, or create help pages in a text editor.

Watch video

1. Go to the page where you want to add help and click the help icon. If you don't see any help icons, click your user image or name in the global header and select **Show Help Icons**. But not all pages have help icons.
2. In the help window, click the **Manage Help Content** link.
3. In the Manage Help Content dialog box, click **Create**.
4. Select a help type:
    - Text: Use a rich text or HTML editor to enter your help content. You can include images, videos, and links.
    - File: Upload a file. Your file opens when users click the link in the help window.
    - URL: Enter the full URL to a website or a file of any type, for example a YouTube video.
5. Select a help security group if you need to restrict access to your help content. Only users with roles in the security group will be able to see the help. The predefined Secured group includes all internal employees and contingent workers.
6. Set the status as Active if you want people to see your help in the help window. Or, you can inactivate your help so that it doesn't appear until you activate it later.
7. Enter the title, which is the text of the link in the help window.

8. If your help type is File, enter a description which users might see after they click the link on the help window, depending on their browser setting.
9. Select another language if your content isn't in American English. In help windows, users see added help in the language they're using for the application.
10. Select a country if your help is targeted at a specific country. Based on what users set in the **Territory** field for their regional preferences, they see generic added help plus added help tagged for their country. There's no impact on predefined help.
11. If your help type is Text, enter your content in the text editor. Or you can click the **Source Code Editing Mode** icon to switch to an HTML editor.
12. Save your work. Be aware that you won't be able to see the new link in the Manage Help Content dialog box or the help window if you set it up for a language, country, or security group that doesn't apply to you.
13. To check your work, click your link in the help window.

Tip: Help that you add using the Text or File type has a unique URL. So, you can link to it from other help or web pages. Updates to the application won't affect the URL or your help content.

How can I add YouTube videos to help windows?

Create help using the URL for the YouTube video.

1. Find the video in YouTube.
2. Click **Share**.
3. Click **Embed**.
4. Copy the URL within the embed code, for example `https://www.youtube.com/embed/<unique ID>`.

   Note: Make sure that your copied URL starts with `https`.
5. Open the help window and click **Manage Help Content**.
6. Click **Create**.
7. Select **URL** as the help type.
8. Paste your copied URL in the **URL** field.
9. Enter other information, and click **Save and Close**.

Add Existing Help to Another Help Window

You can use the same added help in multiple help windows. Let's say you already created some help content for a Create page, and now you want to add it to the corresponding Edit page too.

1. Open the Edit page and click the help icon.
2. In the help window, click the **Manage Help Content** link.
3. Click the **Select and Add** icon in the Added Help section.
4. Find the help you want to link to and select it in the search results.
5. Click **Add**.

Note: When you use the same help in multiple windows, be aware that what you do in one window affects the other windows too. For example, if you change the content or status, the changes apply to all help windows where you added the help. Similarly, if you delete the help in one window, it's removed from the other windows too.

Determine Which Links Appear and in What Order

You can choose to show or hide any of the help you added or any of the predefined help. For the help you added, you can also change the order they appear in the help window.

- While you create help or edit the help that you added, you can set the status so that the link is either shown or hidden in the help window.
- To show or hide predefined help:

  a. In the Manage Help Content dialog box, select a title in the Predefined Help section.
  b. Click the **Change Status** button.
  c. In the Change Status dialog box, select **Active** or **Inactive**.
  d. Click **Save and Close**.

  If multiple help windows link to the same predefined help, the status you set here affects all those windows.
- Click the **Reorder** button on the Manage Help Content dialog box to set the order for links to the help that you added. Your links always show up first in the help window, before any predefined help.

How can I edit the text that appears before the links in help windows?

Many help windows have a sentence or two of help text before the links to help content. To change this text, use the User Interface Text tool. Here are the key things you do in the Search and Replace tab:

- Enter the current help window text in the **Find** field, either the full text or just the part you want to change.
- Enter what you want the new text to be in the **Replace** field.
- Select the **User Interface Text** check box.

Related Topics

- How do I modify text using the User Interface Text tool?

Getting Started Work Area

Manage Getting Started Content

The Getting Started work area provides information to introduce new users to the application. There's one predefined page introducing Oracle Fusion Cloud Applications, and you can add your own pages so that people get content specific to your company.

If there are no added pages for your company, you open the predefined page when you click Getting Started from the Others menu group in the Navigator. But, if you have the Manage Help Content (ATK_CUSTOMIZE_HELP_TOPICS_PRIV) privilege, clicking Getting Started opens a launch page containing a set of links:

- The Oracle Applications link opens the predefined page. You can hide this page for all users, and you can add your own pages too.
- The links for individual product families open blank pages where you can add your own pages if you want to provide content that's specific to particular applications. If you add pages to these links, your users will now see the launch page when they click Getting Started so they can choose which set of content they want to view.

> **Tip:** You can change the text of the links on the launch page using the User Interface Text tool. This flexibility means you could design sets of content for different roles or different locations, for example, rather than for different product families.

To edit Getting Started content:

1. Click **Navigator** > **Others** > **Getting Started**.
2. Click a link to open the set of pages you want to change.
3. Click the **Edit Getting Started** link.

Here's what you can do:

- 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 added pages (not predefined ones).

Related Topics

- How do I modify text using the User Interface Text tool?

How can I add YouTube videos to Getting Started pages?

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. Open the set of pages 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**.

All Added Help

Manage All Added Help Content

Use the Manage Help Content task to review and manage all the help that's been added across all help windows and in the Getting Started work area. For example, find all the help you created that's inactive, and activate them.

1. In the Setup and Maintenance work area, go to the **Manage Help Content** task in the Application Extensions functional area.
2. On the Manage Help Content page, search for added help, for example, by title, status, or who last updated the content.
3. Select search results and do what you need, for example, to edit the help content, change its status, or delete it. Or, click topic titles to see the help content.

Related Topics

- How do I update existing setup data?

Page or Section Values

The page or section value identifies a specific help window. Use this value on the Manage Help Content page to search for help that was added to a specific help window, or to add more help to that window.

Let's see how we can figure out where a help window is, based on its page or section value. If you still can't find the page or section value for a specific help window, go to the help window and add help from there first.

Where You Find Help Windows

First, you need to know where you might find help windows. In most cases, they're next to page or section titles, but here are some examples of where else you might find them:

- On tabs, dialog boxes, panel tabs, dashboards, dashboard regions, and even desktop-integrated Excel workbooks
- In the Setup and Maintenance work area, for example within the table of tasks on the Setup page
- In the Offerings work area, for example within the table on the Edit Features page
- In the Learn More column in the Features Overview work area

How to Read the Values

The page or section value reflects how you navigate 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.

For the Setup and Maintenance, Offerings, and Features Overview work areas, the page or section value tells you the object that the help window is for. These values end with functional area, feature, task, or task list. So for example, **Set Help Options task** refers to the help window that you see for the Set Help Options task, on the Setup page.

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. Here are a couple of examples:

- **Create and Edit Message pages** means that same help window is on both the Create Message and Edit Message pages. Note the plural **pages**.
- **Process Monitor section** means that this help window is in a section that appears on multiple pages. You can usually tell if you see something like a section or tab name without a page or window name in front of it.

Related Topics

- How Functional Setup Manager Components Work Together
- Guidelines for Using Desktop Integrated Excel Workbooks

Create and Edit Added Help

To create or edit help content, usually you start in the help window you want to work on. But you can also use the Manage Help Content page, where you have access to all the help that's already added.

On the Manage Help Content page, you can find sets of added content, such as all the help in a selected language or all the help last updated by a particular colleague. Other than editing this help, you can create help by duplicating and editing the added help, or just creating new help from scratch. After you create help, you can make it available to users in help windows or elsewhere, for example as links on web pages.

Create, Edit, or Duplicate Help

Here's what you do:

1. In the Setup and Maintenance work area, go to the **Manage Help Content** task in the Application Extensions functional area.
2. On the Manage Help Content page, click **Create**. Or, search for your help, select it in the search results, and click **Edit** or **Duplicate**.
3. Select a help type if you're creating brand new help:
    - Text: Use a rich text or HTML editor to enter your help content. You can include images, videos, and links.
    - File: Upload a file of any type. Your file opens when users click the link in the help window.
    - URL: Enter the full URL to a website or a file of any type, for example a YouTube video.
4. Select a help security group if you need to restrict access to your help content. Only users with roles in the security group will be able to see the help. The predefined Secured group includes all internal employees and contingent workers.
5. Set the status as Active if you want people to see your help in any help window you add it to. Or, you can inactivate your help so that it doesn't appear until you activate it later.
6. Enter the title, which is the text of the link in help windows.
7. If your help type is File, enter a description that users might see after they click the link to open your file, depending on their browser setting.
8. Select another language if your content isn't in American English. In help windows, users see added help in the language they're using for the application.

9. Select a country if your help is targeted at a specific country. Based on what users set in the **Territory** field for their regional preferences, they see generic added help plus added help tagged for their country. There's no impact on predefined help.
10. If your help type is Text, enter your content in the text editor. Or you can click the **Source Code Editing Mode** icon to switch to an HTML editor.
11. Save your work.

Make Help Available

After you create help, you should put it on at least one help window so that people can find it where they need it most. To do that, you can go to the help window and add the new topic. Or you can search for the page or section on the Manage Help Content page then select and add your new help.

Help that you create using the Text or File type has a unique URL. From the Manage Help Content page, you can open your help and copy the URL from the browser. So, other than linking to it from help windows, you can also link to it from other help or web pages. Updates to the application won't affect the URL or your help content.

Related Topics

- Create Groups to Limit Access to Added Help

Select and Add Help to Any Help Window

To put existing added help on a help window, usually you open that help window, then select and add the help. But you can also use the Manage Help Content page to put added help on any help window.

It's helpful to use the Mange Help Content window in these cases:

- When you don't have access to the page where you want to add help
- When you want to add help to several help windows without opening up each one

Here's how you put added help on any help window:

1. In the Setup and Maintenance work area, go to the Manage Help Content task in the Application Extensions functional area.
2. On the Manage Help Content page, select a value for the **Page or Section** list in the Search section. The page or section value represents a help window, in this case, the one where you want to add a link.
3. In the Search Results section, click the **Select and Add to Help Location** icon.
4. On the Select and Add page, find the help you want to link to.
5. Select the title and click **Add**.

Related Topics

- Page or Section Values
- How do I update existing setup data?

Copy Added Help for Migration

If you need a copy of all the help that's been added, say for testing, migration, or other purposes, then you're in luck. In the Setup and Maintenance work area, create a configuration package, then use the export and import.

Your configuration package must include one of these:

- The Application Extensions functional area
- A source implementation project that contains the Define Help Configuration task list, with these objects selected to export:
    - Help Configuration
    - Help Topic

Related Topics

- Export Setup Data Using Offering or Functional Area
- Export Setup Data Using Implementation Project

10 Guided Learning

Manage Guided Learning for Onboarding and Training

After you create content in Oracle Guided Learning, you need to apply it to the application so that you can test things out. Then, make your Guided Learning content available to your users.

If you have Oracle Guided Learning, you can create interactive guides and more for onboarding and training, to help people learn how to use Oracle Fusion Cloud Applications. For example, a guide takes users through each step of a task while they're in the application, down to which buttons to click and which fields to fill out. Most importantly, they get through the task in a way that fits your own business processes and practices.

Here's what you do after you create or update Guided Learning content:

1. Set the **Guided Learning Enabled** (FND_GUIDED_LEARNING_ENABLED) profile option to **Yes** at the Site level.
    - Use the Manage Applications Core Administrator Profile Values task in the Application Extensions functional area.
    - You only need to do this once.
2. Activate a sandbox with the Guided Learning tool selected.
3. Define some settings to make sure that your Guided Learning content shows up in the application the way you want it.
    a. In the Setup and Maintenance work area, go to the **Configure Guided Learning** task in the Application Extensions functional area.
    b. On the Configure Guided Learning page, enter the application ID that was given to you when you first got Oracle Guided Learning.
    c. Leave the default value in the **OGL Server** field unless you need to replace it with a different server, for example a test server.
    d. Click **Save and Close**.
4. Test your Guided Learning content while you're in the sandbox.
5. Go back to Guided Learning to make any changes to your guides, including changing content from draft to published.
6. When you're ready, publish the sandbox to make your content available to users.

Next time you have new or changed Guided Learning content, start from step 2 again.

Related Topics

- Create and Activate Sandboxes
- Publish Sandboxes
- Oracle Guided Learning User Guide
- How do I update existing setup data?
- Set Profile Option Values

11 Oracle Fusion Cloud Applications Identity Domain

Overview of Using Oracle Fusion Cloud Applications Identity Domain to Build Extensions

All Oracle Fusion Cloud Applications environments are now provisioned with the Oracle Cloud Infrastructure Identity and Access Management (IAM) identity domain.

This Fusion Applications identity domain is preintegrated with the inbuilt identity system of Fusion Applications and serves as the identity backbone for several tools of Fusion Applications, such as Oracle Visual Builder Studio. It enables federated Single Sign-On (SSO) and token-based authentication across Fusion Applications. This Fusion Applications identity domain is also available to you to deploy your extensions and integrations. Using the Fusion Applications identity domain for building extensions comes with several benefits:

- Avoids the cost for you to create additional OCI identity domains for supporting extensions. No additional licensing cost is incurred when using the Fusion Applications Identity domain to build extensions in Fusion Applications. Note that the Fusion Applications identity domain is provisioned with the *Oracle Apps domain type*. In addition to saving on licensing cost, using the Fusion Applications identity domain instead of creating more OCI identity domains, minimizes operational and governance overhead. It also prevents identity proliferation and any potential risks associated with it.
- User synchronization is configured by default. All users created in Fusion Applications through the security console or through batch processes are actively synchronized to the Fusion Applications identity domain. This synchronization enables assigning Fusion Applications users to OCI-based extensions and integrations.
- Federated SSO is configured by default with the Fusion Applications as the Identity Provider (IdP) and the Fusion Applications identity domain as the Service Provider (SP). This configuration enables single sign-on between the Fusion Applications and custom-built extensions.
- The OAuth token-based authentication allows custom-built applications to connect with the Fusion Applications REST APIs.

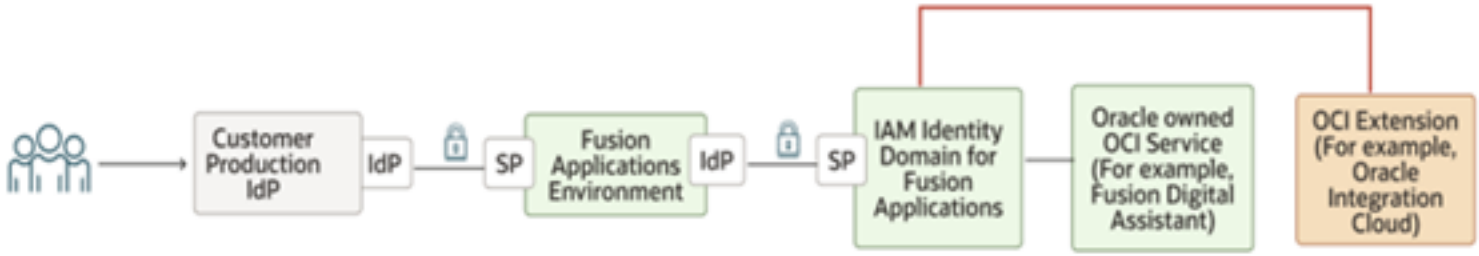
The allowed use cases for the Fusion Applications identity domains are solely Fusion Applications extensions and integrations to the subscribed Oracle SaaS applications, such as Oracle Fusion Cloud HCM, Oracle Fusion Cloud ERP, Oracle Fusion Cloud SCM, and so on.

Here are a few examples of the allowed use cases.

Use Case	Example
Fusion Applications extension	A custom web application for partners. **Note:** It can be a standalone application, but it must leverage data from Fusion Applications.
Fusion Applications integration	Data synchronization of Fusion Applications with a customer product hub. **Note:** Fusion Applications must always be a part of the integration process.

Use Case	Example

Here's a visual representation of how the overall model looks like. Note that the user is synchronized from Fusion Applications to the IAM identity domain. The synchronization ensures that the user identities and access privileges are managed centrally and consistently across the enterprise. This synchronization is configured by default and allows Fusion Applications users to also authenticate Oracle Integration Cloud extensions.



Note: Circular SSO configuration with Fusion Applications as the SP and Fusion Applications identity domain as the IdP isn't supported. So, Multifactor Authentication (MFA) to sign in to Fusion Applications using the Fusion Applications identity domain is also not supported.

Guardrails

The Fusion Applications identity domain includes guardrails to ensure that certain Oracle-owned configurations aren't modifiable. The guardrails ensure that you can use this IAM Identity domain for building extensions without impacting Fusion Applications. For example, you're prevented from doing the following tasks:

- Activate or deactivate Fusion Applications.
- Disable or enable user synchronization.
- Add or modify resource scopes of Oracle resource applications.
- Update provisioning operations, such as authoritative synchronization, account creation, account updation, account activation or deactivation, and account deletion.

Note: This list isn't comprehensive and Oracle may expand the number of protected configurations.

Provision a Oracle Cloud Integration Service into the Fusion Applications Identity Domain

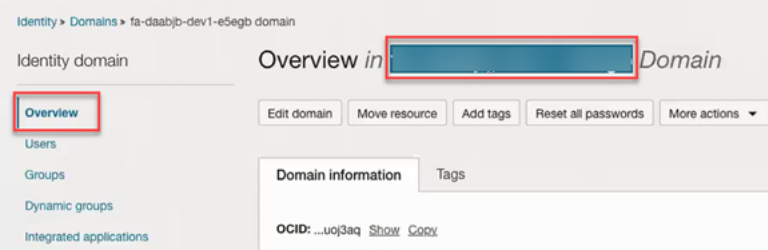
You can provision a new Oracle Cloud Infrastructure Service, such as Oracle Integration, into a Fusion Applications Identity and Access Management (IAM) identity domain so that you don't have to create another IAM domain for Fusion Applications extensions.

This section assumes that you've Oracle Universal credits to create a new instance of an OCI service, such as Oracle Integration. The instructions detail how to create an Oracle Integration instance using the Fusion Applications identity domain.

1. Navigate to *cloud.oracle.com*, enter your cloud account name, and click **Next**.
2. On the Oracle Cloud Infrastructure UI, select the Fusion Applications identity domain from the **Sign in with an identity domain** field and click **Next**.

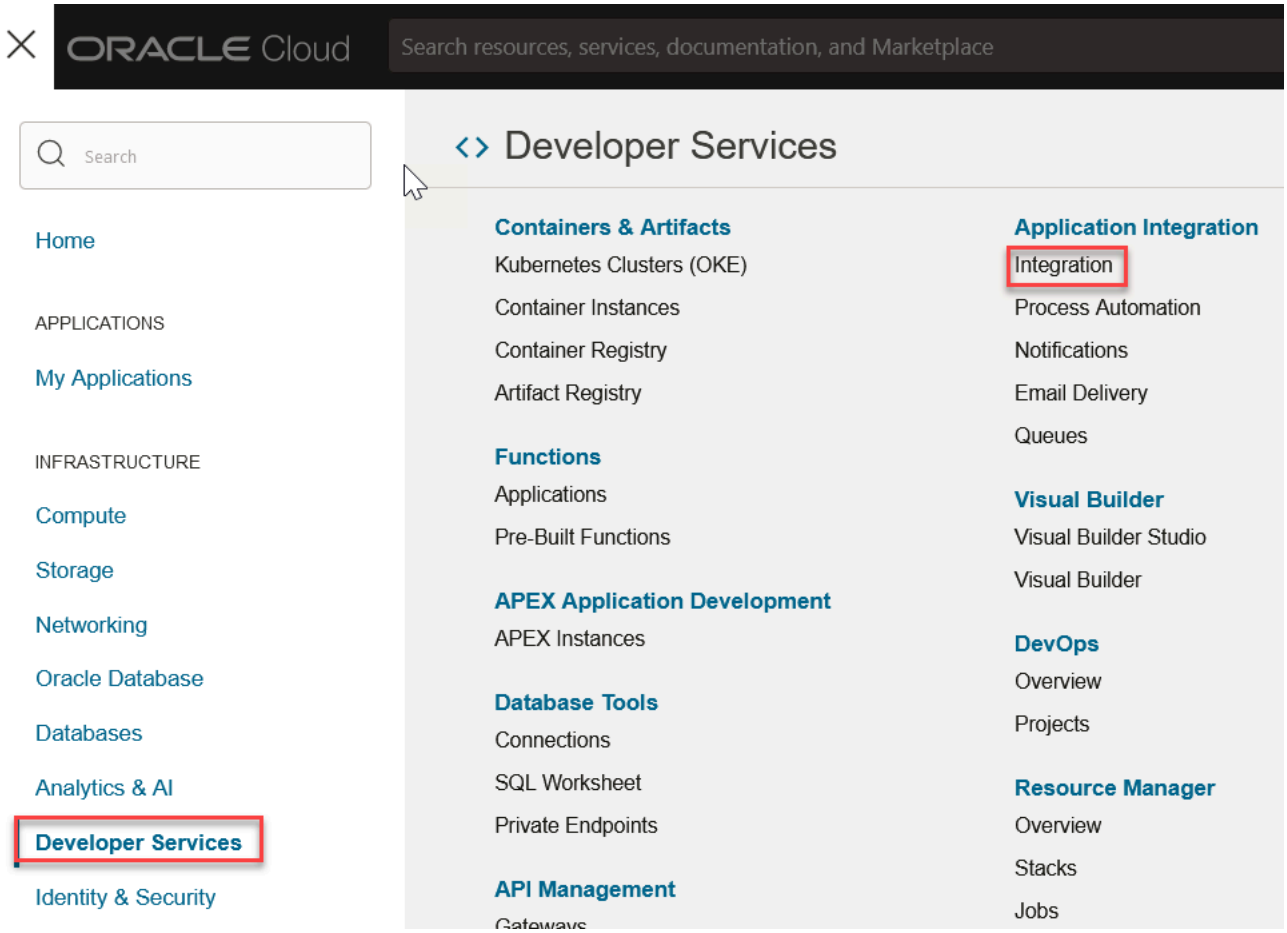
   If you need to identify the Fusion IAM domain associated with your Oracle Fusion Cloud Applications, do these tasks:

   a. Sign in to the default identity domain. You must have administrator privileges.
   b. Click **Navigator menu** > **My Applications** > **Fusion Applications**.
   c. On the Overview page, select the environment to identify the associated IAM identity domain.
   d. On the Environment Details page, click **Integrations**.
   e. Under **Integrations** section, click the URL against the **Associated Identity Domain**.
   f. On the Overview page, note the domain name. Select this domain name from the **Sign in with an identity domain** field.

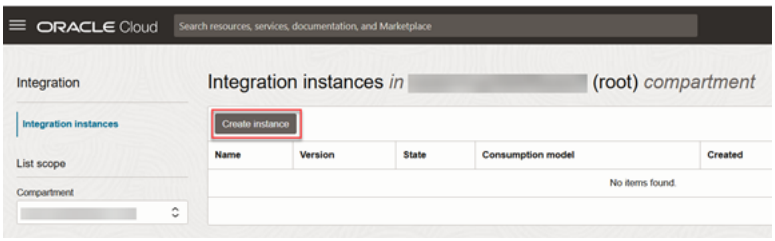


3. Open your Oracle cloud account URL in your browser. Here the tenant name is your cloud account name.
4. On the Oracle Cloud Account Sign In page, enter your cloud account user name and the password that you configured during the cloud account activation and click **Sign In**.
5. Click **Navigation menu** > **Developer Services**.

6. On the Developer Services page, click **Integration** under the **Application Integration** category.



7. From the **Compartment** list, select your desired compartment.
8. On the Integration instances page, click **Create instance**.



9. On the Create instance dialog box, enter these values:
   a. In the **Name** field, enter a name for the instance.
   b. Select the edition: **Standard** or **Enterprise**.
   c. Select the shape: **Development** or **Production**.
   d. Select the license type: **Subscribe to a new Oracle integration license** or **Existing Oracle Fusion Middleware license (BYOL)**.
   e. Click **Create**.

Your Oracle Integration instance, created under the Fusion Applications identity domain and displayed under Integration instances, offers several key features:

- All the Fusion Applications users are automatically available to be assigned to this integration instance.
- Single Sign-On (SSO) is enabled beween Fusion Applications and Oracle Integrations.
- You can configure Oracle Integration to call Fusion Applications APIs using OAuth.

Configure Fusion Applications Identity Domain Federated with an External IdP for OCI Extensions

All Fusion Applications users are available in Fusion Applications identity domain automatically through synchronization. When these users attempt to sign-in to an OCI extension service, they are redirected automatically to Fusion Applications through the Oracle-delivered SSO.

However, you have the option to configure your OCI extension service to further refine access to just those users created locally in the Fusion Applications identity domain or external users that originate from an external IdP directly federated with the Fusion Applications identity domain.

This section explains how to federate the Fusion Applications identity domain with an external IdP such as Microsoft Entra ID. Using the Oracle Cloud Infrastructure Console, you can configure a policy and rules for an external IdP that's federated with the Fusion Applications identity domain. You can then associate the policy with an OCI extension service such as Oracle Visual Builder. Note that only a single policy can be associated with the OCI extension service in a domain.

Prerequisite

Before you proceed with the configuration, make sure that Fusion Applications identity domain is federated with Microsoft Entra ID. For more information, see *SSO Between OCI and Microsoft Azure* in OCI documentation.

Caution: Don't edit the Fusion Applications IdP and the default IdP policy.

Create a Policy and Rules for the External IdP

Create a policy and rules that will apply for the OCI extension application and then associate the policy with the application.

1. On the OCI UI, click Identity and Security, **Domains**.
2. In the **List scope** section, select the root user account in the **Compartment** drop-down list.
3. On the Domains page, select the domain name that's suffixed with current domain. This is the domain that's provisioned with Fusion Applications.
4. On the domain Overview page, click **Security** on the left pane.
5. On the Terms of use documents page, click **IdP policies** on the left pane.
6. On the Identity provider (IdP) policies page, click **Create IdP policy**.
7. Enter a name for the policy and click **Add policy**.
8. Click **Add IdP rule** to assign the rule to the relevant identity providers and set the conditions.
9. On the Add identity provider rule dialog box, enter a name for the rule and specify these additional details.
    a. In the **Assign identity providers** drop-down list, select the identity provider that you want to associate with this rule.
    b. Specify the required conditions.

c. Click **Add IdP rule**.

10. Click **Next**.
11. Click **Add apps** to search for and add the applications and services that'll use this rule.

Assign a Role to the User for Accessing the OCI Extension Application

To access the OCI extension application, the user associated with the Identity Provider must be assigned the required role.

1. On the OCI UI, go to the current domain overview page.
2. Click **Oracle Cloud Services** on the left pane.
3. Select the Oracle Visual Builder application from the list.
4. On the application details page, click **Application roles** in the left pane.
5. Expand the role that you want, using the down arrow.
6. Click **Manage** to view the list of assigned users.
7. Select a user from the list of **Available Users** and click **Assign**.

Verify Access to the Oracle Visual Builder Application

1. Access the Oracle Visual Builder Cloud Service instance. The URL is available in the Primary audience field in the section **Configure application APIs that need to be OAuth protected** on the Oracle Visual Builder Cloud Service overview page.
2. Sign in using the credentials of the identity provider user to whom you assigned the role earlier. You can view the home page of the Oracle Visual Builder Cloud Service application.

Manage Users in the Fusion Applications Identity Domain

User synchronization between Oracle Fusion Applications and Oracle IAM identity domain is configured by default. All users created in Fusion Applications through Security Console are synchronized frequently with the IAM identity domain. This synchronization enables assigning Fusion Applications users to Fusion Applications identity domain application roles, groups, OCI service instances, and application extensions in that identity domain.

In addition to the users federated and synchronized from Fusion Applications, you can directly *create users* in the IAM identity domain or federate users from third-party identity providers. You can assign these users to a *group* and assign that group to application roles. Individual users can also be assigned to application roles. If the IAM identity domain is *federated* with an external IdP such as Microsoft Azure, then the list of users associated with that external IdP are also maintained in the IAM identity domain.

Caution: Although the IAM identity domain lists all users together, never edit the details of users that are synchronized from Fusion Applications or from other federated third-party identity providers because the user data may become out of sync and interfere with the user assignment.

Caution: Don't edit the Fusion Applications synchronized groups.

Extension Use Cases with the Fusion Applications Identity Domain

The use cases listed here describe how to use the Fusion Applications identity domain to build extensions in Fusion Applications and describe how users created in Fusion Applications and any other third-party identity providers are synchronized in the identity domain environment.

Additionally, it also describes how local Fusion Applications identity domain users can be used to segregate access to customer applications deployed in the identity domain.

The use cases also cover scenarios about integrating additional external services into the Fusion Applications developed using Visual Builder Studio. Additionally, the use cases describe federating Fusion Applications identity domain with an external IdP to grant access to non-Fusion users for OCI services used in SaaS Extensions.

To understand more about building extensions using the Fusion Applications identity domain, go through the specific use cases listed here:

- *Extending Fusion Applications using VBCS: OAuth Configurations*
- *Extending Fusion Applications using VBCS: SSO Configurations*
- *Extending Fusion Applications using VBCS: MFA Configurations*
- *How to integrate external services into Fusion Application Visual Builder Studio (VBS) pages (or Fusion Application extensions)*
- *Oracle Fusion Cloud Applications SSO with Microsoft Entra ID*
- *Oracle Fusion Cloud Applications identity domain federation for SaaS Extension (OCI Services)*
- *Subscription of PaaS services in Fusion Cloud Applications Identity Domain*
- *Extending Fusion Applications using VBCS: Identity Domain Subscription*

For extensions that use an external identity domain that requires access to Fusion Applications API using JWT token, see the following blog:

How to Access Fusion Applications by Visual Builder in Different Identity Domain Using OAuth

Configure OAuth to Access Oracle Fusion Cloud Applications REST APIs

Oracle Fusion Cloud Applications expose REST APIs through two resource platforms: Fusion Applications and Spectra services, each serving different purposes.

Fusion Applications provide REST APIs for Fusion application business functionality, while Spectra provides service-oriented REST APIs for platform services. Although both platforms use Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) for OAuth-based authentication and authorization, they support different OAuth grant types, client types, scopes, and configuration procedures.

OAuth in Oracle Fusion Cloud Applications uses an access token, issued as a JSON Web Token (JWT), to grant access to protected resources. The access token is issued by the OCI Identity and Access Management identity domain

associated with your Fusion Applications instance, which acts as the authorization server. All Fusion Applications users are automatically available in the associated identity domain for OAuth-based authentication to access protected REST resources.

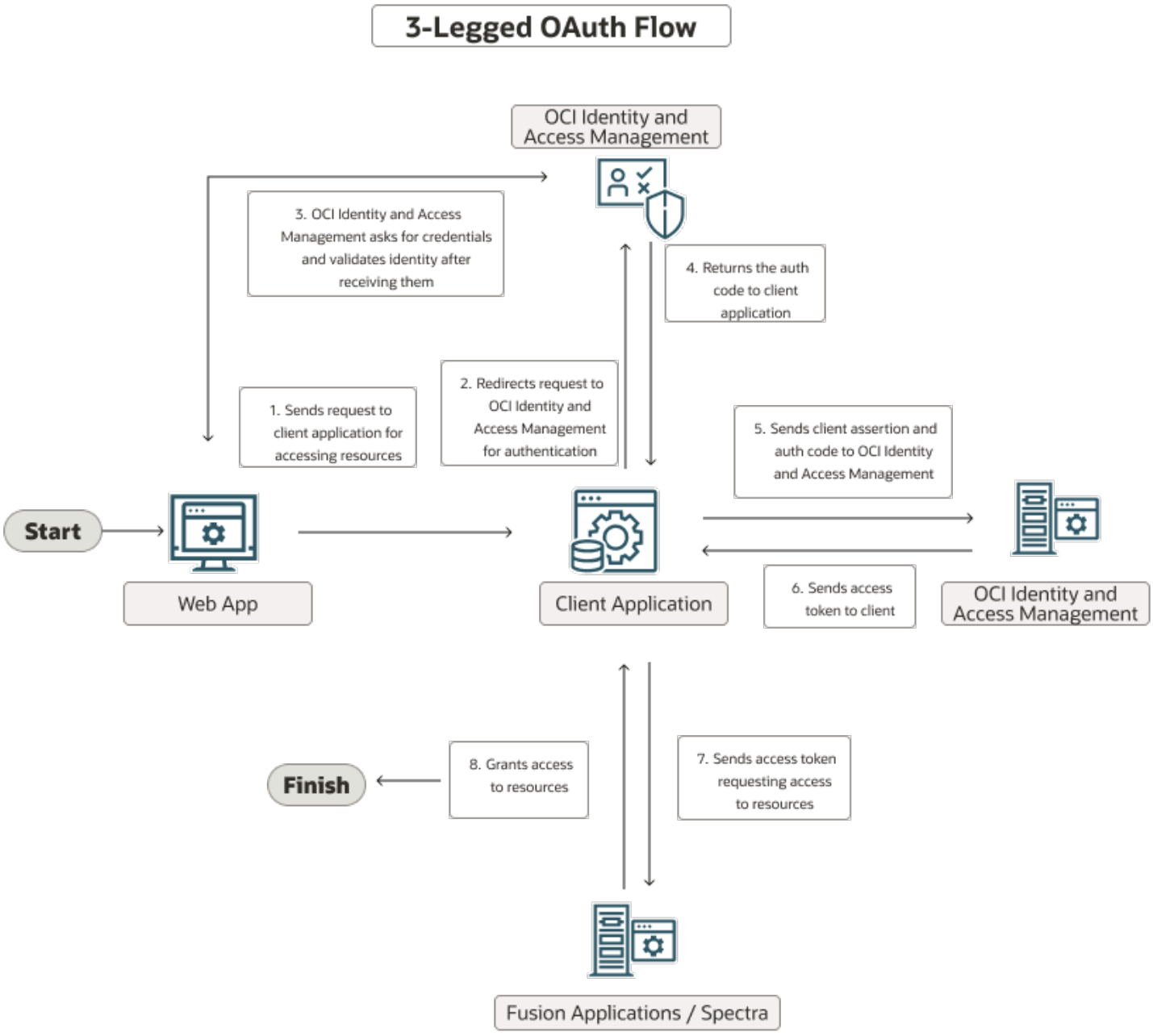
The following sections explain the supported OAuth flows and their differences, describe the available OAuth grant types and their applicability to Fusion Applications and Spectra resources, explain how to identify the appropriate OAuth scopes, and provide step-by-step procedures for configuring the supported grant types.

Difference Between 2-Legged OAuth and 3-Legged OAuth

Both the 2-legged and 3-legged flows result in an access token that the client application uses to access the REST resources. However, your choice between them depends on how you want to authorize access to the REST resources.

A 3-legged OAuth flow is an interactive authorization flow in which a user authenticates with the identity provider and grants consent to the client application to access protected resources on their behalf. If the user is already authenticated, they might not be prompted to sign in again. The OAuth client must be registered with the authorization server (in this case the identity domain). After successful authentication and authorization, the authorization server issues an authorization code to the client application. The client application exchanges this authorization code along with its preassigned client assertion to obtain the access token. Finally, the client sends the access token to the resource server to access the REST resources.

The following illustration explains the flow.



A 2-legged OAuth flow is an authorization flow in which the client application requests an access token directly from the authorization server without requiring user interaction during token acquisition. Depending on the integration scenario and the supported grant type, the client application can either obtain an access token on behalf of a user by presenting a signed JWT user assertion or authenticate itself using the Client Credentials grant type. For details about client and user assertions, see *Client/User JWT Assertion*.

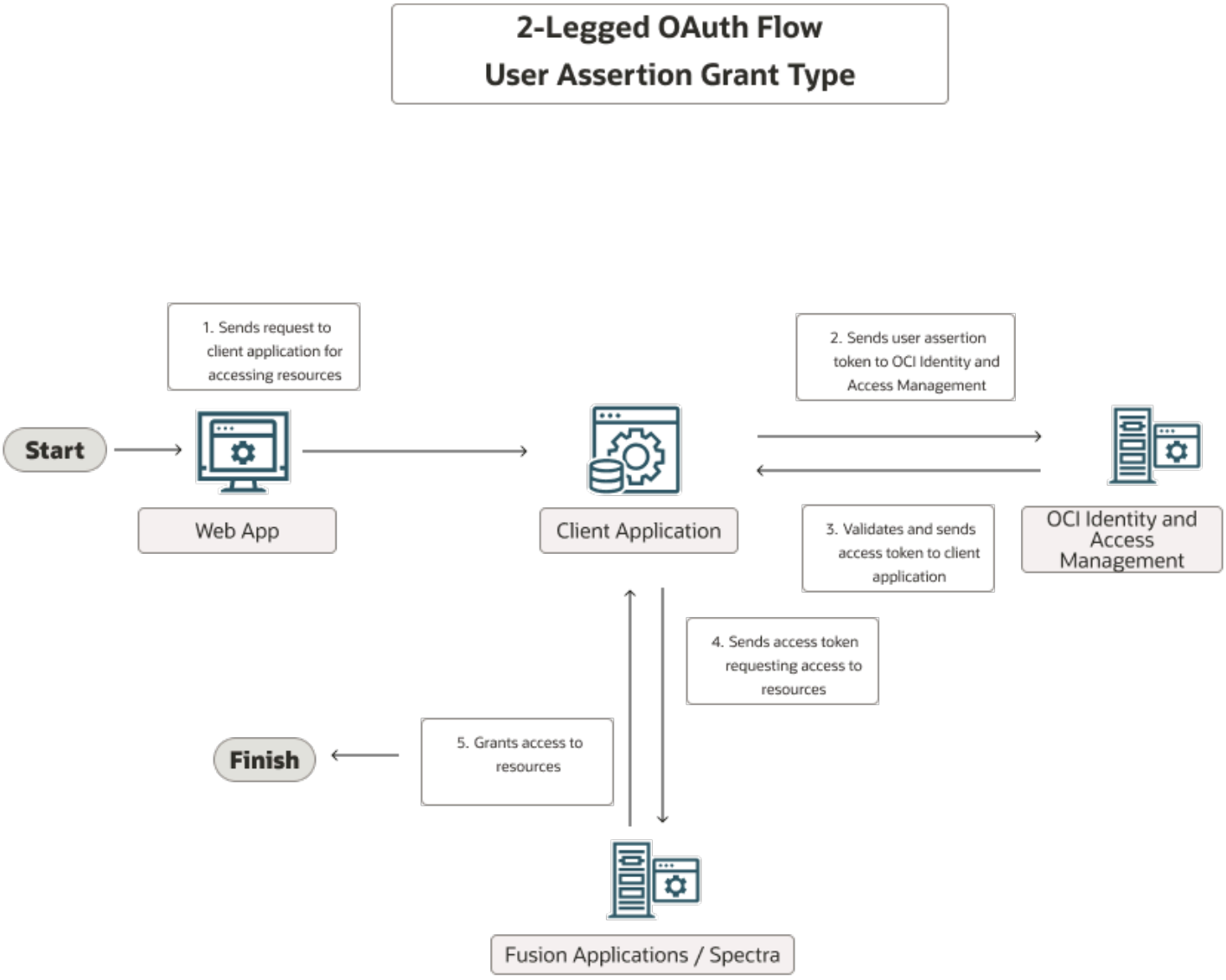
Scenario 1: If you want an application (the OAuth client) to access the REST resources on behalf of a user without collecting the user's credentials at runtime, you can use the JWT User Assertion grant type. In this configuration, the client obtains a user assertion from a trusted identity provider and presents it to the authorization server to request an access token. After successful validation, the authorization server issues an access token that represents the user

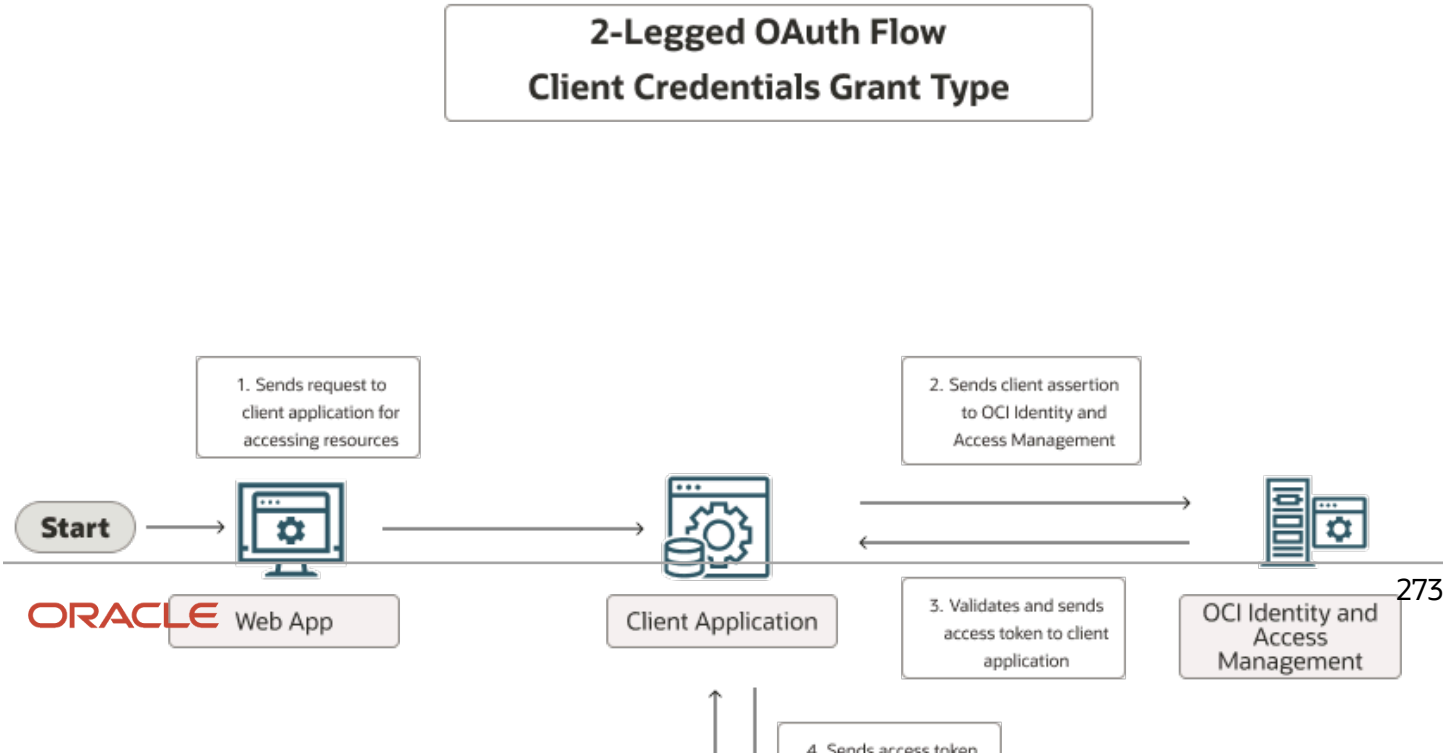
and can be used to access the REST resources on the resource server. In this case, there's no user interaction with the identity provider but the user identity is conveyed through the assertion.

Scenario 2: If you want an application to directly access the REST resources without a user context, you can use the Client Credentials grant type. In this configuration, the client application uses a client assertion to authenticate itself to the authorization server and request an access token. The application then uses the access token to access the REST resources on the resource server. This flow is typically used for service-to-service integrations where user context isn't required.

> **Note:** The Client Credentials grant type can be used only for Spectra services.

The following illustrations will help you understand the differences between these two grant types used in a 2-legged OAuth flow.


2-Legged OAuth Flow
User Assertion Grant Type
1. Sends request to client application for accessing resources
2. Sends user assertion token to OCI Identity and Access Management
Start
Web App
Client Application
OCI Identity and Access Management
3. Validates and sends access token to client application
4. Sends access token requesting access to resources
Finish
5. Grants access to resources
Fusion Applications / Spectra


2-Legged OAuth Flow
Client Credentials Grant Type
1. Sends request to client application for accessing resources
2. Sends client assertion to OCI Identity and Access Management
Start
Web App
Client Application
OCI Identity and Access Management
3. Validates and sends access token to client application

Before You Proceed Further

Before configuring OAuth, ensure that you're familiar with OCI Identity and Access Management and have the necessary privileges to create and manage applications in the identity domain associated with your Fusion Applications instance. To learn more, see *Overview of OCI Identity and Access Management*. Also, make sure that you or other required users have the necessary privileges to access the Identity and Access Management Admin Console to navigate to the identity domain. For details, see *Managing Users*.

At a high level, configuring OAuth involves the following steps:

1. Identify whether you're accessing Fusion Applications or Spectra services, and select the appropriate OAuth grant type for your integration requirements.
2. Identify the identity domain where you'll create the OAuth client application. The identity domain acts as the OAuth authorization server.
3. Configure the OAuth client application in the selected identity domain based on the selected OAuth grant type.
4. Select the appropriate OAuth scope for the REST API that you want to access.
5. Authenticate and obtain an access token from the authorization server.
6. Access the REST resources using the access token.

Identify Your Resource Platform, Supported Grant Types, and OAuth Flows

Configuring OAuth involves identifying the REST API endpoint and the resource platform, selecting a supported OAuth grant type, and selecting the appropriate OAuth flow to obtain an access token. Because the supported grant types, OAuth flows, client types, and scopes differ between Fusion Applications and Spectra services, identify the resource platform before you begin the configuration. Use the following table to identify the appropriate OAuth configuration for your integration.

REST API Endpoint Pattern	Resource Platform	Supported Grant Types and OAuth Flows
`/api/`... Example: `/api/boss/data/objects/ora/commonAppsInfra/objects/v1/setIdSets`	Spectra	• Authorization Code (3-legged) • Authorization Code with PKCE (3-legged) • JWT User Assertion (2-legged) • Client credentials (2-legged)
Any other format Example: `/crmRestApi/resources/11.13.18.05/accounts` Or `/hcmRestApi/scim/Users` Or `/oam/services/rest/access/api/v1/audit`	Fusion Applications	• Authorization Code (3-legged) • Authorization Code with PKCE (3-legged) • JWT User Assertion (2-legged)

For detailed information about the supported grant types, see

- *Access Grant Types* in Oracle Cloud Infrastructure Documentation.
- *Access Grant Types* in REST API for Oracle Identity Cloud Service.

Identify the Client Scope for Your REST Resource

A resource scope defines whether users have access to the application's resources. The scope is defined by the resource application administrator. When a client requests authorization, the returned access token contains this information, including information on whether it's read-only or has create, read, update, and delete (CRUD) access.

To find the scope, go to your domain and look for the Oracle provisioned resource application, such as Fusion Applications. On the service details page, go to the section **OAuth Configuration**, then **Configure application APIs that need to be OAuth Protected**, and then **Scopes**. You'll find the resource scope listed in that section.

While configuring OAuth for your application, you'll need to provide the client scope. The scope would vary based on the type of REST API you're allowing access to. Let's understand how the scope varies by REST API type and where to find the appropriate scope.

> **Note:** The scope examples here are only to explain the OAuth configuration process and shouldn't be treated as the definitive scope values for your integration. Scope values can vary depending on the application, service, and REST API. To identify the correct scope, look for the OAuth configuration of the corresponding application in the identity domain or the product documentation.

REST Resource Type	Finding Scope Definition

For Spectra REST APIs where the resource path is in the format `/api/`..., for example, `/api/boss/data/objects/ora/hcmHrCore/employment/v1/workers`, look for the scope in the Oracle BOSS Cloud service in the identity domain.	1. Sign in to the OCI Identity and Access Management console, navigate to the list of registered identity domains and click the domain associated with Fusion Applications.
	2. From the menu in the sidebar, click Oracle Cloud Services and select, for example, **Oracle BOSS Cloud (Spectra)** service.
	3. To find the primary audience name and the scope, scroll further down to the **Configure application APIs that need to be OAuth protected** section. You'd see something like:
	`Primary audience: urn:opc:resource:fusion:<podname>:boss`
	`Scope: /`
	The client scope is the concatenation of the primary audience name `urn:opc:resource:fusion:<podname>:boss` and the resource scope `/`.
	The complete client scope definition would appear as `urn:opc:resource:fusion:<podname>:boss/`.

For the Fusion Applications REST APIs where the resource path is in any other format, for example, `crmRestApi/resources/latest/accounts`, look for the scope in the Fusion Applications service in the identity domain.	1. Sign in to the OCI Identity and Access Management console, navigate to the list of registered identity domains and click the domain associated with Fusion Applications.
	2. From the menu in the sidebar, click Oracle Cloud Services and select, for example, **Oracle Applications Cloud (Fusion)** service.
	3. To find the primary audience name and the scope, scroll further down to the **Configure application APIs that need to be OAuth protected** section. You'd see something like:
	`urn:opc:resource:fa:instanceid=${fa-instance-id}`
	`or`
	`urn:opc:resource:faaas:fa:${pod-system-name}`
	The client scope is the concatenation of the primary audience name and the resource scope, for example `urn:opc:resource:fa:instanceid=<fa-instance-id>` and `urn:opc:resource:consumer::all`.

Based on the type of the REST API you're using, select the scope and proceed with the OAuth configuration.

Configure OAuth Using Authorization Code Grant Type (Supported for Fusion Applications and Spectra)

In the case of an Authorization Code (3-legged OAuth) configuration, the involved parties are: the user, the application, the identity provider, the authorization server, and the resource server. For the user to be authorized, the application must be registered with the authorization server (in this case Fusion Applications identity domain). Therefore, you'll

need to create a confidential application that can be accessed by multiple users but can keep the OAuth credentials confidential and protected. Let's look at how to do this in the application.

> **CAUTION:** Never change the name of the OAuth token issuer.

> **Note:** This procedure focuses only on the fields and data entry points essential for the configuration. You can ignore and skip any other fields on the UI that aren't mentioned here. Where relevant, instructions to skip the fields are included in the procedure.

1. Register your client and resource applications with the authorization server.
    a. Sign in to the Fusion Applications identity domain as an Administrator.
    b. From the list of identity domains, click Integrated applications and click **Add application**.
    c. On the Add application page, select **Confidential Application** and click **Launch workflow**.
    d. On the Add Confidential Application page, enter a name for the application that you're adding.
    e. Skip all other fields and click **Submit**.
    f. In the OAuth Configuration section, select Edit OAuth and select **Configure this application as a client now** to proceed with the client configuration settings. Because you're creating a client application, only this option is relevant.
    g. In the Authorization section, select the **Authorization code** checkbox to specify the grant type. This is the only grant type required for this flow.
    h. In the **Redirect URL** field, enter the URL of your custom application that requires access to the Fusion Applications REST resources.

        > **Note:** The URL path might vary depending on the type of application. See the documentation of your custom application for the exact information regarding the redirect URL used for OAuth configuration.

    i. In the Client type section, select **Trusted**. A trusted client is allowed to request access tokens using a signed JWT assertion.
    j. Click **Import Certificate** and select the public signing certificate that matches the private key used to sign the JWT assertion.

        > **Note:** While importing the certificate, set the certificate alias to match the Key Identifier (kid) value used in the JWT assertion header.

    k. Skip the next few fields and go to the Token issuance policy section.
    l. In Authorized resources, select **specific** so that the user is granted access only to the specific resources.

2. Add resources for your product.
   a. Turn on **Add resources** to include the resources that are accessible to the users.
   b. In the Resources subsection, click **Add scope**.

      > **Tip:** A scope defines the level of access to the application's resources. The scope is defined by the resource application administrator. To find the scope, go to your domain and look for the Oracle provisioned resource application, such as Fusion Applications. On the service details page, go to the section **OAuth Configuration**, and then **Configure application APIs that need to be OAuth Protected**, and then **Scopes**. You'll find the resource scope listed in that section.

   c. On the Add scope dialog box, select the applications containing the resources, and click **Add**. The resources are added. The scope defined by your application administrator appears in the format: `<resource primary audience name><resource scope name>`.
   d. Review the added resources. If you don't need any resource, select and click **Remove** to exclude it from the scope.
3. Complete the application creation and activate the application.
   a. Go to the Actions menu and select **Activate**. A confirmation message appears asking for a confirmation to activate the application.
   b. Click **Activate application**. The confidential application is activated and ready for use.

      Also, on the OAuth Configuration page, look for the Client ID and make a note of it because you'll need it for requesting the access token.
4. Request an access token. You can do so using a client application such as Postman or a cURL command. Let's look at both these options.

> **Note:** The procedures use the client assertion. Oracle recommends using client assertion instead of client secret because client assertions provide stronger authentication, reduce the risk associated with shared secrets, and simplify secure key management in enterprise environments. The process of obtaining the client assertion token is outside the scope of this document. It's assumed that you already have a valid signed JWT client assertion.

Using Postman	Using cURL

a. In Postman, create a new request.
b. On the Authorization tab, select **OAuth 2.0** as the Authorization Type.
c. In the Configure New Token section, enter the following client configuration:
- Token Name: Enter a generic name for the token.
- Grant Type: Set the grant type to **Authorization Code**.
- Callback URL: Enter the Redirect URL value that's entered in the identity domain.
Note: The URL value must match what's configured in the confidential application.
- Auth URL: The IDCS URL appended with `/oauth2/`
	a. Open a browser and send a request to obtain the authorization code using this URL:
	`https://<identity domain URL>/oauth2/v1/authorize?response_type=code&client_id=<clientid>&redirect_uri=<redirect url>&scope=<scope>`
	The browser gets redirected to the <redirect URL> specified in the confidential application. In the URL path, you can find the authorization code as a value in the `code` query parameter. Copy the authorization code for use later in step c.
	b. Launch the command prompt.
	c. Enter the cURL command below, replacing the text in angle brackets (< >) with the appropriate values:
	`curl --request POST "https://<identity domain URL>/oauth2/v1/token" -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" -d "grant_type=authorization_code&code=<authorization_code>&redirect_uri=<redirect_assertion-type:jwt-bearer&client_assertion=<client_assertion_JWT>"`
	d. You can expect the response in a format as shown here:
	`Status: 200`
	`"access_token":"eyJ4NXQiOiI4Wk. . ."`
	`"token":"Bearer",`
	`"expires_in":3600`

Using Postman

`v1/authorize` is used as the Auth URL. You can get the URLs from `https://idcs-<idcs-id>.identity.oraclecloud.c known/idcs-configuration`.

- Access Token URL: The IDCS URL appended with `/oauth2/v1/token` is used as the Access Token URL. You can get the URLs from `https://idcs-<idcs-id>.identity.oraclecloud.c known/idcs-configuration`.
- Client ID: Enter the Client ID that was generated after creating the confidential application.
- Scope: Mention the scope defined in the confidential application.
- Client Authentication: Select **Send client credentials in body**.

d. In the Advanced section, go to the **Token Request** fields and add the client assertion details.

Key	Value	Send In
`client_ assertic type`	`urn:iet: assertic type:jw bearer`	Request body
`client_ assertic`	The signed client assertion JWT	Request body

e. Click **Get New Access Token**. Postman passes the information to the identity domain for authentication and receives the access token.

Using cURL

e. Copy the value displayed for `access_token`. This is the token you'll need to request access to the resources.

5. Access the resource using the access token.

   In Postman, click **Use Token** to use the token (auto-fill the Token field) and access the resources from the resource server. Or, if you used the cURL command in the previous step, you could continue with the command prompt to send the REST request and get access to the resources. You'd need to structure your request as shown here:

```
curl --request GET "https://<REST API endpoint URL>"
 -H "Content-Type:<as needed by your REST endpoint, such as, application/json>"
 -H "Authorization: Bearer <access_token>"
```

   Note that the Authorization header is set to Bearer token and you'd provide the access token retrieved earlier as its value.

Configure OAuth Using Authorization Code with PKCE Grant Type (Supported for Fusion Applications and Spectra)

Use the Authorization Code with PKCE (Proof Key for Code Exchange) grant type to enable OAuth 2.0 authentication for public clients, such as mobile applications, that can't securely store a client secret. PKCE enhances the Authorization Code flow by protecting the authorization code from interception during the authorization process.

The Authorization Code with PKCE flow involves four parties: the user, the client application, the authorization server, and the resource server. During the OAuth flow, the user authenticates with the identity provider. After successful authentication, the authorization server issues an authorization code to the client application. The client application then sends the authorization code and the PKCE code verifier to the authorization server. After validating the request, the authorization server issues an access token to the client application to access protected resources on behalf of the user. Let's look at how to configure this flow in the application.

> **CAUTION:** Never change the name of the OAuth token issuer.

> **Note:** This procedure focuses only on the fields and data entry points essential for the configuration. You can ignore and skip any other fields on the UI that aren't mentioned here. Where relevant, instructions to skip the fields are included in the procedure.

1. Register your client and resource applications with the authorization server.
    a. Sign in to the Fusion Applications identity domain as an Administrator.
    b. From the list of identity domains, click Integrated applications and click **Add application**.
    c. On the Add application page, select **Mobile Application** and click **Launch workflow**.
    d. On the Add Mobile Application page, enter a name for the application that you're adding.
    e. Skip all other fields and click **Submit**.
    f. In the OAuth Configuration section, select Edit OAuth and select **Configure this application as a client now** to proceed with the client configuration settings. Because you're creating a client application, only this option is relevant.
    g. In the Authorization section, select the **Authorization code** checkbox to specify the grant type. This is the only grant type required for this flow.
    h. In the **Redirect URL** field, enter the URL of your custom application that requires access to the Fusion Applications REST resources.

        > **Note:** The URL path might vary depending on the type of application. See the documentation of your custom application for the exact information regarding the redirect URL used for OAuth configuration.

    i. Skip the next few fields and go to the Token issuance policy section.
    j. In Authorized resources, select **specific** so that the user is granted access only to the specific resources.

2. Add resources for your product.
   a. Turn on Add resources to include the resources that are accessible to the users.
   b. In the Resources subsection, click Add scope.

      Tip: A scope defines the level of access you need to use the application's resources. The scope is defined by the resource application administrator. To find the scope, go to your domain and look for the Oracle provisioned resource application, such as Fusion Applications. On the service details page, go to the section **OAuth Configuration**, and then **Configure application APIs that need to be OAuth Protected**, and then **Scopes**. You'll find the resource scope listed in that section.

   c. On the Add scope dialog box, select the applications containing the resources, and click **Add**. The resources are added. The scope defined by your application administrator appears in the format: `<resource primary audience name><resource scope name>`.
   d. Review the added resources. If you don't need any resource, select and click **Remove** to exclude it from the scope.
3. Complete application creation and activate the application.
   a. Go to the Actions menu and select **Activate**. A confirmation message appears asking for a confirmation to activate the application.
   b. Select **Activate application**. The mobile application is activated and ready for use.

      Also, on the OAuth Configuration page, look for the Client ID and make a note of it because you'll need it for requesting the access token.
4. Request an access token. You can do so using a client application such as Postman or a cURL command. Let's look at both these options.

Using Postman

a. In Postman, create a new request.
b. On the Authorization tab, select **OAuth 2.0** as the Authorization Type.
c. In the Configure New Token section, enter the following client configuration:
- Token Name: Enter a generic name for the token.
- Grant Type: Set the grant type to **Authorization Code (With PKCE)**.
- Callback URL: Enter the Redirect URL value that's entered in the identity domain.

Note: The URL value must match what's configured in the confidential application.

- Auth URL: The IDCS URL appended with `/oauth2/v1/authorize` is used as the Auth URL. You can get the URLs from `https://idcs-<idcs-id>.identity.oraclecloud.c` `known/idcs-configuration`.
- Access Token URL: The IDCS URL appended with `/oauth2/v1/token` is used as the Access

Using cURL

a. Before you construct the cURL request, generate the `code_challenge` from a `code_verifier`. You'll need to use the `code_challenge` value in the request. Generating the `code_challenge` from a `code_verifier` is the standard mechanism defined by the OAuth 2.0 PKCE specification. To generate the `code_challenge` from a `code_verifier`, follow these steps:
- Encode the `code_verifier` as UTF-8 bytes.
- Compute the `SHA-256` hash of those bytes.
- Base64URL-encode the hash (replacing + with -, / with _, and removing any trailing = padding).

For example, the following Python code generates the `code_challenge`:

```
import hashlib
import base64

def generate_code_challenge(code_verifier: str) -> str:
 digest = hashlib.sha256(code_verifier.encode("utf-8")).digest()
 return base64.urlsafe_b64encode(digest).decode("ascii").rstrip("=")

code_verifier = "dBjftJeZ4CVP-mB92K27uhbUJU1p1r_wW1gFWFOEjXk"
print(generate_code_challenge(code_verifier))
```

The code generates the `code_challenge` value. Use this value in the next step when you send the authorization request. Use the `code_verifier` value in step d when you send the token request. The authorization server validates the `code_verifier` against the `code_challenge` before issuing the access token.

b. Open a browser and send a request to obtain the authorization code using this URL:

```
https://<identity domain URL>/oauth2/v1/authorize?
response_type=code&client_id=<clientid>&redirect_uri=<redirect
```

Using Postman	Using cURL

Token URL. You can get the URLs from `https://idcs-<idcs-id>.identity.oraclecloud.c known/idcs-configuration`.
- Client ID: Enter the mobile application Client ID that was generated after creating the mobile application.
- Code Challenge Method: Select **SHA-256**.
- Scope: Mention the scope defined in the confidential application.
- Client Authentication: Select **Send client credentials in body**.
d. Select **Get New Access Token**. Postman passes the information to the identity domain for authentication and receives the access token.
	`url>&scope=<scope>&code_challenge_method=S256&code_challenge=<BASE64URL-ENCODE(SHA256(ASCII(code_verifier)))>`
	The browser gets redirected to the <redirect URL> specified in the confidential application. In the URL path, you can find the authorization code as a value in the `code` query parameter. Copy the authorization code for use later in step d.
	c. Launch the command prompt.
	d. Enter the cURL command below, replacing the text in angle brackets (< >) with the appropriate values:
	`curl --request POST "https://<identity domain URL>/oauth2/v1/token" -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" -d "grant_type=authorization_code&code=<authorization_code>&redirect_uri=<redirect_`
	e. You can expect the response in a format as shown here:
	`Status: 200 "access_token":"eyJ4NXQiOiI4Wk. . ." "token":"Bearer", "expires_in":3600`
	f. Copy the value displayed for `access_token`. This is the token you'll need to request access to the resources.

5. Access the resource using the access token.

   In Postman, click **Use Token** to use the token (auto-fill the Token field) and access the resources from the resource server. Or, if you used the cURL command in the previous step, you could continue with the command prompt to send the REST request and get access to the resources. You'd need to structure your request as shown here:

   ```
   curl --request GET "https://<REST API endpoint URL>"
    -H "Content-Type:<as needed by your REST endpoint, such as, application/json>"
    -H "Authorization: Bearer <access_token>"
   ```

   Note that the Authorization header is set to Bearer token and you'd provide the access token retrieved earlier as its value.

Configure OAuth Using User Assertion Grant Type (Supported for Fusion Applications and Spectra)

If you're setting up an application that's going to process requests on behalf of a user without requiring the user's credentials, but needs to use the user permissions in the requests, you can use the User Assertion grant type to set up a 2-legged OAuth flow. In this scenario, the client creates or obtains a signed JWT user assertion that contains the user identity and presents it to the authorization server for validation, without requiring any interaction from the user. Let's look at how to set it up in the application.

CAUTION: Never change the name of the OAuth token issuer.

Note: This procedure focuses only on the fields and data entry points essential for the configuration. You can ignore and skip any other fields on the UI that aren't mentioned here. Where relevant, instructions to skip the fields are included in the procedure.

1. Create an application that allows users to access the resources.
   a. Sign in to the identity domain as an Administrator.
   b. From the list of identity domains, click Integrated applications and click **Add application**.
   c. On the Add application page, select **Confidential Application** and click **Launch workflow**.
   d. On the Add Confidential Application page, enter a name for the application that you're adding.
   e. Skip all other fields and click **Submit**.
   f. In the OAuth Configuration section, select Edit OAuth and select **Configure this application as a client now** to proceed with the client configuration settings. Because you're creating a client application, only this option is relevant.
   g. In the Authorization section, select the **JWT assertion** checkbox to specify the grant type.
   h. In the Client type section, select **Trusted**. A trusted client is allowed to request access tokens using a signed JWT assertion.
   i. Select **Import Certificate** and select the public signing certificate that matches the private key used to sign the JWT assertion.

      > **Note:** While importing the certificate, set the certificate alias to match the Key Identifier (kid) value used in the JWT assertion header.

   j. Skip the next few fields and go to the Token issuance policy section.
   k. In Authorized resources, select **specific** so that the user is granted access only to the specific resources.
2. Add resources for your product.
   a. Turn on **Add resources** to include the resources that are accessible to the users.
   b. In the Resources subsection, click **Add scope**.
   c. On the Add scope dialog box, select the applications containing the resources, and click **Add**. The resources are added. You'll notice that the scope defined by your application administrator is available in the format: `<resource audience name><resource scope name>`.
   d. Review the added resources. If you don't need any resource, select and click **Remove** to exclude it from the scope.
3. Complete application creation and activate the application.
   a. Go to the Actions menu and select **Activate**. A confirmation message appears asking for a confirmation to activate the application.
   b. Click **Activate application**. The confidential application is activated and ready for use.
4. Request an access token. You can do so using a client application such as Postman or a cURL command. Let's look at both these options.

> **Note:** The procedures require a valid signed JWT user assertion and a valid signed JWT client assertion. The user assertion is used as the authorization grant to obtain an OAuth access token, while the client assertion is used to authenticate the client application. Oracle recommends using a client assertion instead of a client secret because client assertions provide stronger authentication, reduce the risks associated with shared secrets, and simplify secure key management in enterprise environments. The processes for obtaining the user assertion and client assertion are outside the scope of this document. It's assumed that you already have both assertions.

Using Postman	Using cURL
a. In Postman, create a new request. b. Set the request method to POST and enter the token endpoint URL: `https://<identity domain URL>/oauth2/v1/token`.	a. Launch the command prompt. b. Enter the cURL command below, replacing the text in angle brackets (< >) with the appropriate values: `curl --request POST "https://<identity domain URL>/oauth2/v1/token"`

Using Postman

c. On the Authorization tab, select **No Auth** as the Authorization Type.

d. On the Headers tab, select **Content-Type** as the key and **application/x-www-form-urlencoded** as its value.

e. On the Body tab, select **x-www-form-urlencoded** and add the required form fields along with the values:

Key	Value
`grant_type`	`urn:ietf:params:o type:jwt-bearer`
`asserti`	The user assertion token
`scope`	The client scope defined in the confidential application
`client_ asserti type`	`urn:ietf:params:o assertion-type:jwt-bearer`
`client_ asserti`	The signed client assertion JWT

f. Click **Get New Access Token**. Postman passes the information to the identity domain for authentication and receives the access token.

Using cURL

```
 -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8"
-d "grant_type=urn:ietf:params:oauth:grant-type:jwt-
bearer&assertion=<user_assertion_token>&scope=<scope>&client_assertion_type=urn:i
assertion-type:jwt-bearer&client_assertion=<client_assertion_JWT>"
```

c. You can expect the response in a format as shown here:

```
Status: 200

"access_token":"eyJ4NXQiOiI4Wk. . ."

"token":"Bearer",

"expires_in":3600
```

d. Copy the value displayed for `access_token`. This is the token you'll need to request access to the resources.

5. Access the resource on the resource server using the access token.

   In Postman, create a new request with Bearer Token as the authentication type and paste the obtained access token in the Token field. Or, if you used the cURL command in the previous step, you could continue with the command prompt to send the REST request and get access to the resources. You'd need to structure your request as shown here:

```
curl --request GET "https://<REST API endpoint URL>"
 -H "Content-Type:<as needed by your REST endpoint, such as application/json>"
 -H "Authorization: Bearer <access_token>"
```

   Note that the Authorization header is set to Bearer token and you'd provide the access token retrieved earlier as its value.

Configure OAuth Using Client Credentials Grant Type (Supported only for Spectra)

If you're setting up an application that processes requests without any user intervention and doesn't require user-delegated permissions, you can use the Client Credentials grant type to implement a 2-legged OAuth flow. In this

scenario, the application securely stores its private key and uses it to generate a signed client assertion, which the authorization server validates to authenticate the client without requiring any user interaction. This setup is typically recommended for client-server interactions that handle automated, application-level requests. Let's look at how to set it up in the application.

> **CAUTION:** Never change the name of the OAuth token issuer.

> **Note:** This procedure focuses only on the fields and data entry points essential for the configuration. You can ignore and skip any other fields on the UI that aren't mentioned here. Where relevant, instructions to skip the fields are included in the procedure.

1. Create an application that allows users to access the resources.
    a. Sign in to the identity domain as an Administrator.
    b. From the list of identity domains, select Integrated applications and select **Add application**.
    c. On the Add application page, select **Confidential Application** and select **Launch workflow**.
    d. On the Add Confidential Application page, enter a name for the application that you're adding.
    e. Skip all other fields and click **Submit**.
    f. In the OAuth Configuration section, select Edit OAuth and select **Configure this application as a client now** to proceed with the client configuration settings. Because you're creating a client application, only this option is relevant.
    g. In the Authorization section, select the **Client credentials** checkbox to specify the grant type.
    h. In the Client type section, select **Trusted**. A trusted client is allowed to request access tokens using a signed JWT assertion.
    i. Click **Import Certificate** and select the public signing certificate that matches the private key used to sign the JWT assertion.

        > **Note:** While importing the certificate, set the certificate alias to match the Key Identifier (kid) value used in the JWT assertion header.

    j. Skip the next few fields and go to the Token issuance policy section.
    k. In Authorized resources, select **specific** so that the user is granted access only to the specific resources.
2. Add resources for your product.
    a. Turn on **Add resources** to include the resources that are accessible to the users.
    b. In the Resources subsection, click **Add scope**.
    c. On the Add scope dialog box, select the applications containing the resources, and click **Add**. The resources are added. The scope defined by your application administrator appears in the format: `<resource primary audience><resource scope>`.
    d. Review the added resources. If you don't need any resource, select and click **Remove** to exclude it from the scope.
3. Complete application creation and activate the application.
    a. Go to the Actions menu and select **Activate**. A confirmation message appears asking for a confirmation to activate the application.
    b. Select **Activate application**. The confidential application is activated and ready for use.

4. Request an access token. You can do so using a client application such as Postman or a cURL command. Let's look at both these options.

> **Note:** The procedures use the client assertion. Oracle recommends using client assertion instead of client secret because client assertions provide stronger authentication, reduce the risk associated with shared secrets, and simplify secure key management in enterprise environments. The process of obtaining the client assertion token is outside the scope of this document. It's assumed that you already have a valid signed JWT client assertion.

Using Postman

a. In Postman, create a new request.
b. On the Authorization tab, select **OAuth 2.0** as the Authorization Type.
c. In the Configure New Token section, enter the following client configuration:
- Token Name: Enter a generic name for the token.
- Grant Type: Select **Client Credentials**.
- Access Token URL: The IDCS URL appended with `/oauth2/v1/token` is used as the Access Token URL. You can get the URLs from `https://idcs-<idcs-id>.identity.oraclecloud.c known/idcs-configuration`.
- Client ID: Enter the Client ID value that was generated after creating the confidential application.
- Scope: Mention the scope defined in the confidential application.
- Client Authentication: Select **Send client credentials in body**.

d. In the Advanced section, go to the **Token Request** fields and add the client assertion details.

Key	Value	Send In
`client_ assertio type`	`urn:iet: assertio type:jwt bearer`	Request body
`client_ assertio`	The signed client assertion JWT	Request body

e. Click **Get New Access Token**. Postman passes the information to the identity domain for

Using cURL

a. Launch the command prompt.
b. Enter the cURL command below, replacing the text in angle brackets (< >) with the appropriate values:

```
curl --request POST "https://<identity domain URL>/oauth2/v1/token"
 -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8"
 -d
 "grant_type=client_credentials&scope=<scope>&client_id=<client_id>&client_assert
assertion-type:jwt-bearer&client_assertion=<client_assertion_JWT>"
```

c. You can expect the response in a format as shown here:

```
Status: 200

"access_token":"eyJ4NXQiOiI4Wk. . ."

"token":"Bearer",

"expires_in":3600
```

d. Copy the value displayed for `access_token`. This is the token you'll need to request access to the resources.

Using Postman	Using cURL
authentication and receives the access token.	

5. Access the resource using the access token.

   In Postman, click **Use Token** to use the token (auto-fill the Token field) and access the resources from the resource server. Or, if you used the cURL command in the previous step, you could continue with the command prompt to send the REST request and get access to the resources. You'd need to structure your request as shown here:

   ```
   curl --request GET "https://<REST API endpoint URL>"
    -H "Content-Type:<as needed by your REST endpoint, such as application/json>"
    -H "Authorization: Bearer <access_token>"
   ```

   Note that the Authorization header is set to Bearer token and you'd provide the access token retrieved earlier as its value.

After you've completed the OAuth configuration, you can assign roles to the client application in Security Console for role-based access control. For details, see *Manage Roles in Custom OAuth Client Applications Using Application Extensions Page*.

Production to Test Considerations for Extensions with Fusion Applications Identity Domain

Production-to-Test refers to cloning your production environment onto your test environment. The process essentially copies all your data and configurations from the production to the target environment. It also copies all the users from the production to the target environment.

Here are a few things to keep in mind when executing this process for extensions secured by the IAM identity domain:

- When a Fusion Applications environment is live, the Fusion Applications users are automatically synced to the IAM identity domain using an automated process.
- During a product-to-test process, this sync process ensures that all users copied to the test or development environment are synchronized into the corresponding IAM identity domain. The time taken to synchronize all these users depends on the number of users cloned during the production-to-test process.
- Any locally created users in that IAM identity domain aren't affected by the sync process.