

Oracle® Application Framework

Personalization Guide

Release 11*i*

Part No. B25439-02

June 2006

Oracle Application Framework Personalization Guide, Release 11i

Part No. B25439-02

Copyright © 2000, 2006, Oracle. All rights reserved.

Primary Author: Siu Chang

Contributing Author: David Weld

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

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

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

U.S. GOVERNMENT RIGHTS

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

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

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

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

Contents

Send Us Your Comments

Preface

1 Customization Primer

Customization Overview	1-1
Configuration	1-2
Personalization	1-3
Extensibility	1-6
Interoperability	1-7

2 Personalizing OA Framework Applications

Branding	2-1
Customizing Look-and-Feel (CLAF)	2-4
Icons	2-21
Responsibilities and Menus	2-22
Messages	2-22
Lookup Codes	2-22
Style Sheets.	2-23

3 Personalizing Your Pages and Portlets

Overview.	3-1
Personalization Levels	3-2
OA Personalization Framework Features.	3-5
Profile Options Used By OA Personalization Framework	3-7

4 Administrative-Level Personalizations

Overview.	4-1
Creating Admin-Level Personalizations	4-2
Using the Page Layout Personalization Page	4-5
Using the Page Hierarchy Personalization Page.	4-12
Examples: Creating New Items Using the Personalization UI.	4-17
Function-Level Personalizations	4-20
Known Issues.	4-20

Related Information	4-21
-------------------------------	------

5 Administrative-Level Personalization User Interface

Overview.	5-1
Focused Page Hierarchy Personalization Page	5-2
Choose Personalization Context Page	5-4
Personalize Page	5-6
Create Query Page.	5-13
Page Hierarchy Personalization Page: Expanded Search Region	5-14
Update Layout Page	5-17
Reorder Contents Page.	5-17
Add Content Page	5-19
Create Content Page	5-20
Create Item Page	5-20
Update Item Page	5-34
Manage Personalization Levels Page	5-36
Extract Translation Files Page	5-37
Upload Translations Page	5-38
Translate Page	5-39
Choose Languages Page	5-40
Personalize Views Page	5-41
Create View, Update View, and Duplicate View Pages	5-42
Rename Columns / Totaling Page	5-49

6 User-Level Personalization User Interface

Overview.	6-1
Personalizing the Views Panel of a Search Page.	6-1
Personalizing a Table Region at the User Level	6-2
Personalize Views Page	6-3
Create View, Update View, and Duplicate View Pages	6-4
Rename Columns / Totaling Page	6-6
Personalizing an LOV Choice List.	6-7
Known Issues.	6-7

7 Portlet-Level Personalization User Interface

Overview.	7-1
Personalizing a Portlet	7-1
Create View and Update View Pages	7-1
Rename Columns / Totaling	7-3
Known Issues.	7-4

8 Translating Personalizations

Overview.	8-1
Translating Personalizations Stored in the AK Repository	8-1

Translating Personalizations Stored in MDS	8-3
OA Extension Translation Toolset	8-8
XLIFF Extractor	8-8
XLIFF Importer	8-13
Mapping a Language Code to a Language-Territory Code	8-14
Known Issues	8-16
Related Information	8-16

9 Deploying Personalizations

Overview	9-1
Deploying Personalizations Using the Functional Administrator Responsibility	9-1
Deploying Personalizations Using the Import/Export Command Line Tools	9-3
Export Tool	9-4
Import Tool	9-9
Deploying Personalizations in the AK Repository	9-12
AKLOAD	9-13
Known Issues	9-15
Related Information	9-15

10 Migrating AK Personalizations

Overview	10-1
Oracle-Seeded User-Level Personalizations	10-2
Migrating Customer Personalizations	10-2
Known Issues	10-4
Related Information	10-4

A Personalization Considerations

Overview	A-2
Attachments	A-2
Auto-Repeating Layout	A-3
Bound Values	A-3
Branding	A-3
Buttons (Global)	A-3
Charts and Graphs	A-3
Concurrent Processing	A-4
Configurable Pages	A-4
Custom HTML	A-5
Date Picker	A-5
Declarative Page Flow	A-5
Dialog Pages	A-5
Dynamic User Interface	A-5
Flexfields	A-5
Forms / OA Framework Page Integration	A-5
formValue	A-5

HGrid	A-5
Hide/Show	A-6
Images in Your Pages	A-6
Locator Element: Breadcrumbs	A-6
Mobile Applications	A-6
Page Contents Bottom Line	A-6
Page Footer	A-6
Printable Page	A-6
Processing Page	A-6
Rich Text Editor	A-7
Save Model (Warn About Changes)	A-7
Shuttle	A-7
Standard Web Widgets	A-7
SubTab Navigation	A-7
Switchers (Application, Context and Table Content)	A-7
Tables - Advanced	A-7
Tables - Classic	A-10
Tree	A-11
Workflow Worklist	A-11

B OA Framework Personalization and Customization Known Issues

Overview	B-1
Personalization/Extensibility	B-1
Translation	B-2
Customizing Look-and-Feel (CLAF)	B-2

C OA Framework Personalization Caveats

Overview	C-1
Personalization Caveats - for Customers	C-1

D Personalization FAQ

Overview	D-1
Customizing Look-and-Feel (CLAF)	D-1

Index

Send Us Your Comments

Oracle Application Framework Personalization Guide, Release 11i

Part No. B25439-02

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document. Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

Note: Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the new Applications Release Online Documentation CD available on Oracle Metalink and www.oracle.com. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: appsdoc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and enquire about our Oracle University offerings. A list of Oracle offices is available on our website at www.oracle.com.

Preface

Intended Audience

Welcome to Release 11i of the *Oracle Application Framework Personalization Guide*.

This documentation assumes familiarity with Oracle E-Business Suite applications. It is written for the consultants and administrators who oversee the functional requirements of these applications and deploy the functionality to their users.

See Related Documents on page x for more Oracle Applications product information.

TTY Access to Oracle Support Services

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

Documentation Accessibility

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

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

Structure

- 1 Customization Primer
- 2 Personalizing OA Framework Applications
- 3 Personalizing Your Pages and Portlets
- 4 Administrative-Level Personalizations
- 5 Administrative-Level Personalization User Interface
- 6 User-Level Personalization User Interface
- 7 Portlet-Level Personalization User Interface
- 8 Translating Personalizations
- 9 Deploying Personalizations
- 10 Migrating AK Personalizations
- A Personalization Considerations
- B OA Framework Personalization and Customization Known Issues
- C OA Framework Personalization Caveats
- D Personalization FAQ

Related Documents

You can choose from many sources of information, including online documentation, training, and support services, to increase your knowledge and understanding of Oracle Applications.

If this guide refers you to other Oracle Applications documentation, use only the Release 11*i* versions of those guides, including:

- Oracle Applications Component Reference
- Oracle Applications User's Guide
- Oracle Applications System Administrator's Guide
- Oracle Applications Developer's Guide
- Oracle Application Framework Developer's Guide (available on Oracle MetaLink [<http://metalink.oracle.com/>])

Tip: For information about upcoming Oracle Application Framework instructor-led training, please refer to Oracle University's course offerings [http://education.oracle.com/web_prod-plq-dad/plsql/show_desc.redirect?redir_type=3&p_org_id=&lang=].

Do Not Use Database Tools to Modify Oracle Applications Data

Oracle STRONGLY RECOMMENDS that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle Applications data unless otherwise instructed.

Oracle provides powerful tools you can use to create, store, change, retrieve, and maintain information in an Oracle database. But if you use Oracle tools such as SQL*Plus to modify Oracle Applications data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data.

Because Oracle Applications tables are interrelated, any change you make using an Oracle Applications form can update many tables at once. But when you modify Oracle Applications data using anything other than Oracle Applications, you may change a row

in one table without making corresponding changes in related tables. If your tables get out of synchronization with each other, you risk retrieving erroneous information and you risk unpredictable results throughout Oracle Applications.

When you use Oracle Applications to modify your data, Oracle Applications automatically checks that your changes are valid. Oracle Applications also keeps track of who changes information. If you enter information into database tables using database tools, you may store invalid information. You also lose the ability to track who has changed your information because SQL*Plus and other database tools do not keep a record of changes.

Customization Primer

This chapter covers the following topics:

- Customization Overview
- Configuration
- Personalization
- Extensibility
- Interoperability

Customization Overview

Oracle Applications has a layered architecture, where each layer encapsulates the maximum reusable set of features without creating dependencies on higher layers. Such architecture enhances reusability of functionality and makes possible global customizations. The task of customizing an Oracle Application can fall into one of a few categories:

- Configuration, page 1-2: using pre-built features to fine-tune the application to match the business and deployment practices of a particular customer. Configuration examples:
 - Setup of a chart of accounts.
 - Setup of business groups or organizations.
 - Setup of logging and auditing profiles.
- Personalization, page 1-3: declaratively tailoring the user interface (UI) look-and-feel, layout or visibility of page content to suite a business need or a user preference. Personalization examples:
 - Tailor the color scheme of the UI.
 - Tailor the order in which table columns are displayed.
 - Tailor a query result.
- Extensibility, page 1-6: extending the functionality of an application, beyond what can be done through personalization. Extensibility examples:
 - Add new functional flows.
 - Extend or override existing functional flows.
 - Extend or override existing business logic.

- Interoperability, page 1-7: interfacing Oracle Applications with third party applications and service providers. Interoperability examples:
 - Interface with a single sign on server.
 - Interface with a credit rating service.
 - Interface with a legacy application.

These customization categories aren't always clear cut. Certainly, in some cases, customization tasks can even span a couple of categories. This book is designed to give a high level perspective of the various customization categories, and discusses only those categories that OA Framework facilitates. References to other resources are provided, where other customization categories are discussed in more detail.

Configuration

Configurations exist almost in every layer and every application. Broadly, configurations can be classified into three classes, as follows:

Deployment Topology Configurations

Deployment topology configurations map closely to the hardware topography of a deployment and are mostly done through technology stack configuration parameters. Examples:

- Setting up the number of Java Virtual Machines (VM) to run on each middle-tier server.
- Setting up the number of database connections.
- Setting up the JServ parameters.

Configurations under this category are documented in greater detail in each technology stack layer's respective administration manuals, which include the following:

- *Oracle Applications AD Procedures Guide*
- *Oracle Applications AD Utilities Reference Guide*
- *Install Oracle Applications*
- *Upgrading Oracle Applications*

Global Functionality Configurations

Global functionality configurations cut across application families and are mostly done through shared technologies such as AOL (Applications Object Library), TCA (Trading Community Architecture), Tasks, Notes, etc. Examples:

- Setting up the multi-org hierarchy.
- Setting up the various party business relationships.
- Setting up various Profiles and Responsibilities.

Configurations under this category are documented in greater detail in the respective layer's implementation and administration manuals, which include the following:

- *Multiple Organizations in Oracle Applications*
- *Multiple Reporting Currencies in Oracle Applications*

- *Oracle Applications Flexfields Guide*
- *Oracle Applications System Administrator's Guide*
- *Oracle Self-Service Web Applications Implementation Manual*
- *Oracle Workflow Guide*

Application or Functional Area Configurations

Configurations associated with a particular functional area (such as accounting) or application. Examples:

- Setting up General Ledger chart of accounts.
- Setting up employee benefit packages.
- Setting up an online catalog.

Configurations under this category are documented in a greater detail in a respective application's implementation manual, grouped under:

- *ERP Product Manuals*
- *CRM Product Manuals*

Personalization

OA Framework was designed with durable personalization capabilities. Durability of OA Framework personalization is largely attributed to the declarative architecture and the object-oriented approach underlying the implementation. Declarative UI component definitions are stored in the form of meta-data in a database repository. Personalizations are translated into offsets from the base meta-data definition and stored separately. At runtime, the applicable personalizations meta-data is uploaded from the repository and layered over the base meta-data definition to produce the net effect. Product upgrades and patches affect only the base meta-data definition, so customer personalizations continue to function properly as applicable.

Personalization Levels

The built-in personalization UI facilitates a variety of personalization features at a number of different levels within the following user groups:

Oracle's In-House E-Business Suite Developer

- **Seeded Function Level** - like the Function Level available to Administrators (see the following section), but personalizations made at this level can only be changed or deleted by Oracle.
- **Seeded User Level** - like the User Level available to End Users (see the following section), but personalizations made at this level can only be changed or deleted by Oracle. (Also referred to as "Oracle-seeded user-level" personalizations.)
- Other seeded levels - Oracle Applications Developers can create and ship personalizations at any of the Administrator personalization levels discussed in the following section, but these are not protected against change and deletion by Administrators at the customer site.

For additional information, refer to Chapter 4: Implementing Specific UI Features: Personalizable Pages in the *OA Framework Developer's Guide*.

Oracle Applications Administrator

- **Function Level** - the customer administrator can define functions and use them as context for granular level personalizations. For example, you can create a function-level personalization to "hide the salary field, if the user is updating an employee record, but not when the user is creating a new employee".
- **Industry Level** - the customer administrator can use the delivered set of predefined industry categories to define personalizations according to vertical market distinctions.

Seeded personalizations may be provided at this level, but customer administrators can also create their own admin-level Industry personalizations.
- **Localization Level** - the customer administrator can use locales as context for personalizations such as "showing a different address field label based on country settings".
- **Site Level** - the customer administrator can introduce global personalizations that affect all users with access to the given application component, such as "setting the number of rows shown in a table".
- **Organization Level** - the customer administrator can introduce personalizations that affect all users belonging to a particular organization or business unit with access to the application component. Example: "sort notifications by age for one organization and by urgency for another".
- **Responsibility Level** - the customer administrator can introduce personalizations that affect all users of a particular responsibility with access to the application component. Example: "show a trend graph for the sales manager responsibility".
- **Seeded User Level** - like the User Level available to End Users (see below), but personalizations made at this level are visible to all users and can only be changed or deleted by the customer administrator. (Also referred to as "admin-seeded user-level" personalizations.)
- Refer to Personalizing Your Pages and Portlets, page 3-1 for additional information.

Oracle Applications User

- Application Users can save personalized views of a query results region and retrieve them at a later time. User level personalizations aren't seen by other users.
- Refer to Personalizing Your Pages and Portlets, page 3-1 for additional information.

Available Personalizations

The following administrator and end user personalizations are available:

Administrator Personalizations

- Change number of rows displayed in a table.
- Change product branding (image).
- Change region header icon.

- Hide or show regions and items.
- Change layout order of regions and items within the boundaries of the parent region.
- Include or exclude descriptive flexfield segments.
- Define up to three sorting levels for tabulated data.
- Filter (restrict querying of) tabular data.
- Change item labels and region headers.
- Change required state of non-mandatory items.
- Update allowed state for updateable items.
- Enable totals for table columns, when applicable.
- Alter the item cascading style sheet (CSS) - to personalize the look and feel of an item.
- Set a default value for an item.
- Define tips (in line instructions and usage help) for associated items.
- Add new items to an existing region. Typically, as part of an extensibility project, where new items are limited to specific styles.
- All administrator personalizations are visible to the end user.
- System Personalizations - in addition to the above, the following are some cross application personalizations facilitated by both OA Framework and Application Object Library:
 - Branding
 - Style sheets
 - Images
 - Responsibilities
 - Menus
 - Messages
 - Lookup Codes
 - Pre-packaged Flexfields
 - Customizing Look and Feel

End User Personalizations

Unlike Administrators, Users can create and save several personalized views that can be retrieved conveniently at a later time. That said, end-user personalized views are limited in scope to Query regions with search results tables. For these regions, end-users can personalize any of the following features:

- Change the number of rows displayed in a table.
- Hide or show regions and items (results table columns are a popular example).
- Change the layout order of regions and items within the boundaries of the parent region (order of results table columns are a popular example).
- Define up to three sorting levels for tabulated data.

- Filter (restrict query) tabular data.
- Change item labels and region headers.
- Enable totals for table columns, when applicable.

Extensibility

OA Framework was designed with durable extensibility capabilities. Durability of OA Framework extensibility is largely attributed to the declarative architecture and the object-oriented approach underlying the implementation. The JDeveloper wizards and built-in personalization UI make it easier to extend Oracle Applications. In addition, Oracle customers can take advantage of the extensibility offered by Flexfields (*Oracle Applications Flexfields Guide*) and Oracle Workflow (*Oracle Workflow Guide*).

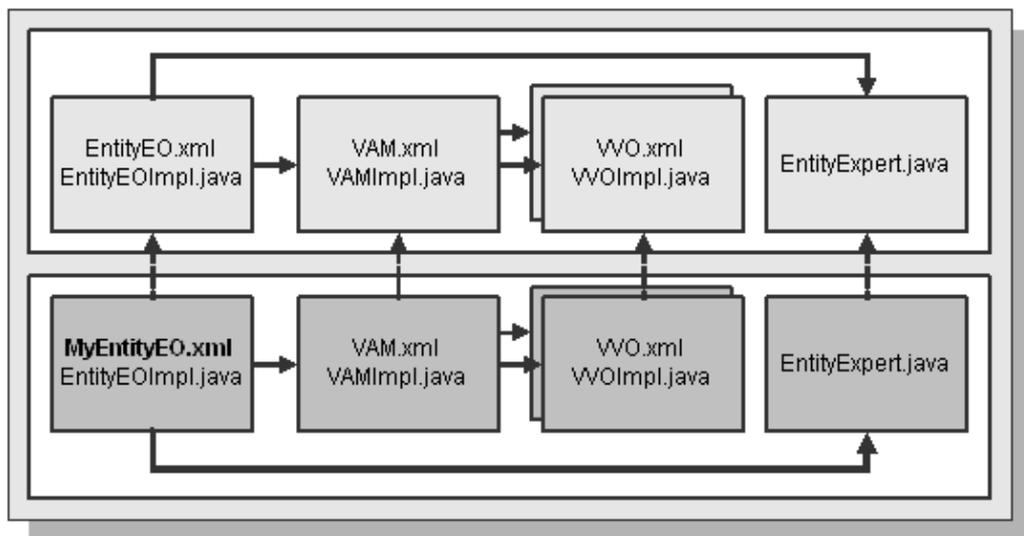
OA Framework extensibility is geared to enable customers to add new functionality and override or extend existing business logic beyond what can be accomplished via personalization. This includes the following extensibility scenarios:

- Adding new pages or complete flows.
- Adding a new attribute (i.e. field) to a prepackaged page.
- Extending attribute defaulting logic.
- Extending validation logic.

To understand how extensibility works, one must understand how OA Framework applications are built. Please refer to the following sections in the *OA Framework Developers Guide*:

- Chapter 2: OA Framework Essentials: **Anatomy of an OA Framework Page**
- Chapter 3: Building an OA Framework Application: **Implementing the Model**
- Chapter 3: Building an OA Framework Application: **Implementing the View**

Extensibility is often observed in the UI, but implementation is mostly centered on the underlying business objects. The following diagram depicts the BC4J objects involved when extending an OA Framework application.



The first row of the diagram represents an exhaustive list of all possible objects a developer might create when creating an entity object. The first box illustrates that when creating an entity object, two files are generated: the meta-data definition XML file and the actual implementation Java class file. Entity Objects handle attribute level and record level validations. These validations often need to use other View Objects, called Validation View Objects (VVO). Validation View Objects are grouped under a Validation Application Module (VAM). Like Entity Objects, creating a VVO and a VAM generates a meta-data definition XML file and an implementation java class file for each object.

Finally, the Entity Object sometimes relies on a helping class to offer, among other services, a validation service optimized for usage by other Entity Objects. This helper class is called the Entity Expert and is linked to the Entity Object through an Entity Object property.

Note: The diagram illustrates a case in which all objects are extended, which is not always the case. In most of the situations, you may be satisfied with extending just a portion of these objects.

Caution: Never edit the base definition of an object or make a copy of a base object. Always extend the relevant object and use BC4J substitutions to reference the extended object.

For example, you may be satisfied with extending the Entity Expert to override a validation method such as `isSupplierValid`. In such a case, it is not wise to reference the extended Entity Expert (`MyEntityExpert`) directly from the base Entity Object (`EntityEO.XML`), as such an approach does not survive upgrades.

A durable approach requires extending the base Entity Object (using the JDeveloper Wizards) and updating the entity expert property on the extended Entity Object to point to the extended Entity Expert.

Interoperability

Implementing Oracle Applications for established customers sometimes involves interfacing with legacy applications or third party services. Interoperability scenarios can be classified into three levels:

- **Deployment Wide** - these involve cross application services that can be interfaced transparently from the application. Example: integration with a single-signon server.
- **Application Specific** - these involve special interoperability features that the application is directly aware of. Example: integration between Oracle iPayment and credit card processors. These are documented in the respective application implementation manuals.
- **Function Specific** - these involve industry standards for publishing a variety of interfaces used to interoperate with third party applications and services. Such industry standards include Web Services, Enterprise Java Beans, MIME, and so on.

In addition to the above, the Oracle Applications technology stack is consolidating around the OA Framework technology stack. In the interim, some CRM applications are not migrated from the JTT technology stack. Instead, OA Framework has created interoperability solutions that allow for these technology stacks to coexist and facilitate a smooth user experience upon transition between technology stacks.

Personalizing OA Framework Applications

This chapter covers the following topics:

- Branding
- Customizing Look-and-Feel (CLAF)
- Icons
- Responsibilities and Menus
- Messages
- Lookup Codes
- Style Sheets

Branding

For a full discussion of Branding, refer to the Browser Look-and-Feel (BLAF) guidelines on Branding on the Oracle Technology Network [<http://otn.oracle.com/tech/blaf/specs/branding.html>] (OTN).

Basic Branding

Basic branding includes the display of corporate ("Oracle") and product brand names. If you want to personalize the branding displayed on your OA Framework pages, you must first determine what branding size is set for the Branding Size profile option:

- Profile Option Name: FND: Branding Size
- Profile Option Internal Name: FND_BRANDING_SIZE
- Profile Option Level: SITE ONLY
- Profile Option Values: Regular, Medium, or Small
- Profile Option Default Value: Small
- Description: This profile controls the size of the global buttons and the corporate and product branding displayed at the top of an OA Framework page.

A value of `Regular` sets the corporate and product branding images to the largest size, as shown in the following example, and renders the global buttons with corresponding icons and text links.

Figure 2-1: FND: Branding Size profile option value set to Regular (corresponds to BLAF "Large" layout)



A value of `Medium` sets the corporate and product branding images to a lower profile, as shown in the following example, and displays the global buttons as text links only.

Figure 2-2: FND: Branding Size profile option value set to Medium (corresponds to BLAF "Medium" layout)



A value of `Small` sets the corporate image to the largest size, but displays the product branding as text drawn from one of several possible sources, as shown in the following example. The global buttons are displayed as text links only.

Figure 2-3: FND: Branding Size profile option value set to Small (corresponds to BLAF "Regular" layout)



Personalizing the Corporate Branding Image

Every page contains a `corporateBranding` page element that has an Oracle corporate image file assigned to it. That image file is `/OA_MEDIA/FNDSSCORP.gif` and contains the Oracle corporate image. You can change the corporate branding image that appears on your page(s) in one of two ways:

- To *globally* override the Oracle corporate branding image with your own corporate branding image, set the profile option **Corporate Branding Image for Oracle Applications** (`FND_CORPORATE_BRANDING_IMAGE`) to the name of an image file (GIF) that contains your corporate image. For example, `CustomerImage.gif`. The image file `CustomerImage.gif` should be placed under the `$OA_MEDIA` directory. This profile option should be set at the Site level and does not have a default value.
- Or, create an administrative-level personalization, page 5-6 for a specific page by setting the **Image URI** property on the **corporateBranding** page element to a different image file. This updates the image only for the specific usage on that page.

Warning: Do not attempt to globally override the corporate branding image by renaming your custom corporate branding image file as

/OA_MEDIA/FNDSSCORP.gif. This results in distortion of your corporate image. When you render a page, OA Framework checks whether a value is set for the profile option **Corporate Branding Image for Oracle Applications**. If no value is set, it renders /OA_MEDIA/FNDSSCORP.gif as the corporate branding image and sets the image area size to 134 x 23 (which is the actual size of the image in FNDSSCORP.gif). If this profile option has a value set, it renders the content of the specified image file without specifying an image area size since the image size is as yet, unknown.

Personalizing Medium or Regular Product Branding

Every page contains a **productBranding** page element that has an image file assigned to it. The image file contains a regular-sized version of the branding image. All branding image files reside in the **OA_MEDIA** directory. For each branding image file in **OA_MEDIA**, there is also a corresponding *brandingimage_MED.gif* file that contains a medium-sized version of the same image. OA Framework renders one of these two branding images based on whether the branding size profile is set to Regular or Medium.

If the **FND: Branding Size** (FND_BRANDING_SIZE) profile option is set to **Regular** or **Medium**, you can change the product branding image in one of two ways:

- You can replace the appropriate image file with a file of the same name that contains an updated image. This updates all instances of that image globally.
- You can create an administrative-level personalization, page 5-6 for the page by setting the **Image URI** property on the **productBranding** page element to a different image file. This updates the image for the specific usage on that page.

Personalizing Small Product Branding

For small branding, when the **FND: Branding Size** profile option is set to **Small**, the product branding is derived in one of three possible ways:

- **Default behavior** - If your OA Framework page is launched from the Navigator in the E-Business Suite Personal Home Page, OA Framework automatically sets the small branding text for you based on the current selected responsibility and page link. This ensures consistency between the options presented in the Navigator, and the branding text displayed on the application page.

Note: The default branding is set only if the FND:Framework Compatability Mode profile option value is set to **11.5.10**.

- **Declarative branding behavior** - A developer may set small product branding declaratively on the page itself and override the default branding behavior. If this is the case, you can personalize the product branding text by creating an administrative-level personalization, page 5-6 for the page that sets the Text property on the **productBranding: formattedText** page element to the product name you want to display.
- **OAPB override behavior** - A developer may also define a Small Branding form function and call it using the OAPB URL parameter with the page to specify the product branding text. When a developer specifies an OAPB parameter programmatically, it overrides both the default and declarative branding behavior, if any. The only way to personalize the product branding text in this case is by logging

into Oracle Applications and updating the **User Function Name** value for that small branding function, using the Form Functions screen.

In-Context Branding

In-Context branding includes contextual information rendered below corporate and product branding information, as shown in the following example.

Figure 2-4: Example of In-Context Branding



To personalize in-context branding, create an administrative-level personalization, page 5-6 for the page that sets the Text property on the **productBranding: formattedText** page element to the product name you want to display. Then set the Text property on the **inContextBranding: formattedText** page element to the context you wish to display. For example, the Text property on the **inContextBranding: formattedText** page element displayed in Figure 2-4 is, "Customer Sun Microsystems - Menlo Park".

Customizing Look-and-Feel (CLAF)

Oracle E-Business Suite applications are currently shipped with Oracle's corporate Browser Look-and-Feel (BLAF), which supplies a rich set of components for web-based applications. Although OA Personalization Framework provides you with the ability to change the look of a page by changing user interface (UI) component properties, adding UI components, and so on, it does not allow you to create and apply an entirely new Look-and-Feel to an application.

The new Customizing Look-and-Feel (CLAF) feature addresses this issue by providing a self-service based UI to modify the Look-and-Feel of an application. The feature enhances OA Personalization Framework by allowing personalization administrators to:

- Create a new Look-and-Feel (LAF).
- Register the new LAF.
- Apply the LAF at the Application, Responsibility, Site, Organization or User levels.
- Update a custom LAF.

UIX currently provides OA Framework with the following LAFs, which can be used directly in web applications:

- **Browser Look-and-Feel (BLAF)** - the default LAF that conforms to Oracle's corporate UI guidelines.
- **Minimal Look-and-Feel (MLAF)** - a simplified version of BLAF with a lightweight UI.
- **Base Look-and-Feel** - the root of all LAF implementations.
- **Simple Look-and-Feel (SLAF)** - a LAF built on top of Base Look-and-Feel to offer more customization features, and to serve as an illustration of how to create LAF extensions on top of a base LAF.

You can build custom LAFs by extending Base LAF, Simple LAF, or another custom LAF.

Note: You cannot extend BLAF or MLAF. See *Deprecated - Custom Style Sheets in OA Framework Prior to 11.5.10.2CU*, page 2-30 if you want to make modifications to the styles in BLAF.

Infrastructure of a Look-and-Feel

A Look-and-Feel is defined by three major components: style sheets (XSS), page 2-5, icons, page 2-5, and renderers, page 2-5.

Style Sheets

A style sheet document (.xss extension) lists the styles for the Look-and-Feel. Styles control the color and font of HTML components. For a complete discussion of style sheets and styles, please refer to the Style Sheets topic, page 2-23 in this chapter. Style sheets are located in /OA_HTML/cabo/styles.

See also the list of global styles, page 2-18 provided by UIX later in this chapter.

Icons

Some web beans are composed of one or more icons that control the web bean's Look-and-Feel. Icons are identified by a name. For example, the Hide/Show web bean consists of the "disclosed" icon. Icons are present in the LAF configuration file. For additional information, refer to the Icons topic, page 2-21 in this chapter.

Renderers

A renderer controls how a web bean lays out its children and generates HTML. If the layout of the LAF you wish to create is different from the standard layout provided by Oracle, you will have to write new renderers for those components that differ in their layout. You define renderers declaratively as templates (.uit extension). Following is an example template definition for a sidebar component:

```
<!-- Template used by sample LAF for side bar. -->
<templateDefinition
  xmlns="http://xmlns.oracle.com/uix/ui"
  xmlns:ui="http://xmlns.oracle.com/uix/ui"
  xmlns:html="http://www.w3.org/TR/REC-html40"
  targetNamespace="http://www.example.org/demo/templates"
  localName="sidebar">
  <content>
  ...
  </content>
</templateDefinition>
```

You should have one template renderer for each component that has a custom layout. The section between the <content> and </content> tags contains the desired UI layout and any references to other UI components. The UI layout is represented in uiXML and HTML.

Important: You can use a template renderer to customize the layout of some - but not all - components. Following is a list of the components with customizable layouts:

- BorderLayout
- breadCrumbs
- bulletedList

- button
- contentContainer
- flowLayout
- footer
- globalButtonBar
- globalHeader
- header
- hideShow
- link
- messageComponentLayout
- pageButtonBar
- pageLayout
- printablePageLayout
- rowLayout
- sideBar
- sideNav
- stackLayout
- subTabBar
- subTabLayout
- tabBar
- tableLayout
- tip
- train

Look-and-Feel Extension (Custom Skins)

When you define a new Look-and-Feel, you can also alter the layout of its inherited Look-and-Feel, thereby creating what is called a custom skin. To complete the creation of a custom skin, you must register any custom renderers, page 2-17, custom facet renders, page 2-6, and custom icons, page 2-17 for that skin, on your custom Look-and-Feel using the Customizing Look-and-Feel UI.

Note: A facet is an optimized variation of a LAF, usually created for a specific output medium. For example, a printable version of a page can be implemented as a facet by excluding superfluous navigation and personalization controls that are not necessary for printed output.

OA Framework will provide a set of skins or web page layout themes that extend the base and simple LAFs from UIX. You can use these skins to change the way Oracle E-Business Suite application pages lay out their content. These skins may be applied at the Site, Responsibility, Application, Organization or User level by setting the profile

option **Oracle Applications Look and Feel** (APPS_LOOK_AND_FEEL). The skins that OA Framework will provide are under development.

For more information, see Custom Renderer and Custom Icon Standards, page 2-17.

Example

The following code is an example of content in a LAF extension XML that defines a new skin. This example represents a LAF identified by a family called `customlaf`. Since it extends the `simple.desktop` LAF, it inherits all the styles from the `UIX Simple Look-and-Feel (SLAF)`. This custom LAF overrides the renderer for page layout by providing its own template-based renderer for page layout. It also provides a custom printable facet page layout renderer (which is initiated to render the page when you run the page in printable page mode) and some custom icons.

```
<lookAndFeel xmlns="http://xmlns.oracle.com/uix/ui/laf"
             xmlns:ui="http://xmlns.oracle.com/uix/ui"
             id="customlaf.desktop"
             family="customlaf"
             extends="simple.desktop"
             styleSheetName="customlaf-desktop.xss">

  <!-- Custom Renderers -->
  <renderers>
    <!-- Register a custom pageLayout Renderer -->
    <renderer name="ui:pageLayout">
      <template name="pageLayout.uit"/>
    </renderer>
  </renderers>

  <!-- Facet custom Renderers -->
  <renderers facets="printable">
    <!-- Register a custom pageLayout Renderer -->
    <renderer name="ui:pageLayout">
      <template name="printablePageLayout.uit"/>
    </renderer>
  </renderers>

  <!-- Custom Icons -->
  <icons>
    <!-- Provide some icon -->
    <icon name="ui:tabBarStart">
      <contextImage uri="images/laf/customlaf/tbs.gif"
                   width="8" height="26"/>
    </icon>
  </icons>
```

Customizing Look-and-Feel (CLAF) User Interface

The Customizing Look-and-Feel feature provides a self-service user interface that allows you to create a new Look-and-Feel, as well as update an existing Look-and-Feel. It does not, however, provide a user interface to create custom template renderers. You must first create/write any custom template renderers that you require before you can create the look and feel.

Prerequisites

- You should have a good knowledge of `UIX` web beans and be able to identify which specific `UIX` component maps to a given component in your web site.

- You should have a good understanding of UIX XSS infrastructure and be able to identify what color and font styles are associated with each component. Refer to the Style Sheets topic, page 2-23 and the Global Styles table, page 2-18 for additional information.
- You must ensure that you have write permission on the file system where OA Personalization Framework is running. If write permission is not granted before you attempt to create or update a Look-and-Feel, the UI will throw an exception.
- You need to write a template renderer only if a component has a layout and children that are different from that of the Browser Look-and-Feel (BLAF). If you need to register a new template renderer, follow these steps before proceeding to the Accessing the CLAF UI section, page 2-8 following:
 1. Name your custom template renderer file as `<webBeanType>.uit`. For example: `sideBar.uit`, `tabBar.uit`, or `pageLayout.uit`.
 2. Create a folder named `<lookandFeelId>` under `$HTML_TOP/cabo/templates`, where `<lookandFeelId>` would be a LAF name, such as `custom-desktop`.
 3. Place the `<webBeanType>.uit` file in the folder `$HTML_TOP/cabo/templates/<lookandFeelId>`. For example to register a new template renderer for `tabBar`, move `tabBar.uit` so that resides as `$HTML_TOP/cabo/templates/custom-desktop/tabBar.uit`, where `custom-desktop` is the `<lookandFeelId>` folder.
 4. Continue with the Accessing the CLAF UI, page 2-8 and Creating a New LAF, page 2-9 sections to create new LAF.

Important: To ensure that your template renderer registers properly, make sure the new LAF you create has the same LAF name as the `<lookandFeelId>` folder you created in the second step.

To update an already registered template renderer, first replace the old `<webBeanType>.uit` file with your new modified version of the `<webBeanType>.uit` file. Next, proceed to the Accessing the CLAF UI, page 2-8 and Updating a LAF, page 2-16 sections. Make sure you replace `<webBeanType>.uit` in the appropriate `<lookandFeelId>` folder.

Accessing the CLAF UI

The Customizing Look-and-Feel user interface can be accessed in one of two ways:

- Using the Oracle Applications Users window, add the responsibility **FND_CLAF_ADMINISTRATOR** to the user who should be given permission to create new LAFs. Make sure you bounce your web server after saving the change so that the setting takes effect. A Customizing Look and Feel Administrator responsibility should now be available for the user. After selecting that responsibility, the user should then select **Customizing Look and Feel Administrator** to navigate to the first page of the CLAF UI.
- In the Page Hierarchy Personalization page, page 4-12 of the Admin Personalization UI, the Page Hierarchy HGrid displays a column called Customize Look and Feel. A **Customize Look and Feel** (pen) icon appears in the column if the following two conditions are met:
 - The application page's current Look-and-Feel is customizable. For example if the current Look-and-Feel is BLAF, an icon will never appear because BLAF is not customizable.

- The component is LAF customizable, page 2-20. This means that only a component associated with a style or icon can have the **Customize Look and Feel** icon enabled. If a personalization administrator does not have access to the Customizing Look and Feel Administrator responsibility when he or she selects the icon, OA Framework returns an error.

When you select the Customize Look and Feel icon in the HGrid for a specific page element, you navigate to the second page of the CLAF UI (Customize Styles and Icons, page 2-11) for that page element.

Creating a New LAF

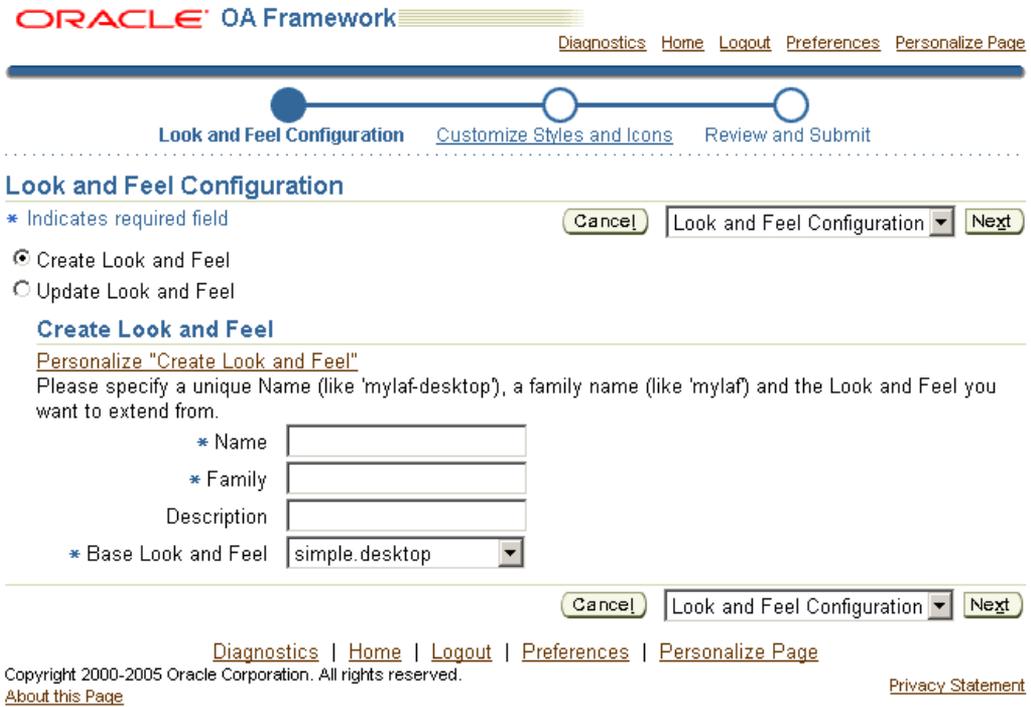
Aside from creating custom template renderers yourself, the CLAF UI provides all the other features needed to create a new look and feel. The UI allows you to:

- Specify the new Look-and-Feel identifier or name. The identifier must be unique, all lower case, and must follow the standard `<lookAndFeelFamilyName>-<device>` naming convention, where `<device>` is either `desktop` or `pda`.
- Specify the family under which the Look-and-Feel belongs (in all lower case). Multiple LAFs for different devices can exist under the same family. For example, **myclaf-desktop** and **myclaf-pda** both exist under the **myclaf** family.
- Specify the base Look-and-Feel that your new custom Look-and-Feel shall extend. All styles, icons, and renderers are inherited from this base look and feel, and can be overridden.
- Modify the styles inherited from the base Look-and-Feel to suit the colors and fonts of the new Look-and-Feel you want to create.
- Modify the icons inherited from the base Look-and-Feel to suit the new look and feel you want to create.
- Get immediate feedback on the changes being made to styles and icons using the preview page.
- Register any custom template renderers you define for any web beans.

The following steps outline how to create a new custom LAF or skin using the CLAF user interface:

1. Evaluate a good sampling of pages in the web site for which you want to create a new Look-and-Feel or skin. Determine whether the overall layout and component order as specified in BLAF is sufficient for the new skin. If it is, then you do not need to create a new LAF, but simply need to modify certain styles or icons. If it is not, that is, the layout and component order differs from that of BLAF, you will need to define renderers for each of those components.
2. Log in as a user with access to the Customizing Look and Feel Administrator responsibility. Select the menu option **Customizing Look and Feel Administrator** to initiate the CLAF UI.
3. In the Look and Feel Configuration page, select the **Create Look and Feel** radio button.

Figure 2-5: Creating a LAF in the Look and Feel Configuration page



4. Specify values for the following parameters, then select **Next**:

- **Name** - (Required) A unique name for the LAF, in *all lower case*, following the naming standard `<lookAndFeelFamilyName>-<device>`.

Caution: The name of your new LAF must not contain upper case letters. Although you can enter a name with upper case letters, the OA Personalization Framework converts them to lower case in the filename when saving the LAF. If you try to update this LAF later, the system will not be able to find it, and will generate an error.

OA Framework automatically generates an internal `.xml` file and an `.xss` file using this name. For example, if you name your LAF `custom-desktop`, OA Framework automatically generates the Look-and-Feel configuration file `$HTML_TOP/cabo/lafs/custom-desktop.xml`, and the stylesheet file `$HTML_TOP/cabo/styles/custom-desktop.xss`.

- **Family** - (Required) The family under which this LAF is being created. Specify the family as all lower case. For example, the `custom` LAF family can have two LAFs named `custom-desktop` and `custom-pda`.
- **Description** - (Optional) Enter text to describe the LAF that you are creating.
- **Base Look and Feel** - (Required) The base LAF that the LAF you are creating will extend. The set of existing LAFs in your environment is displayed in the poplist.

Note: You cannot extend the Oracle corporate BLAF (Browser Look-and-Feel) or the MLAF (Minimal Look-and-Feel).

5. In the Customize Styles and Icons page that appears, the set of *global* named styles is displayed by default. The **Component** poplist also displays `global` by default to indicate that the page is currently displaying the global named styles, page 2-18 that affect more than one component. You can also choose the Selectors or Icons sub tabs to display any global selectors or global icons.

Figure 2-6: Customize Styles and Icons page.

ORACLE Customizing Look and Feel Administrator Diagnostics Home Logout Preferences Personalize Page

[Look and Feel Configuration](#) **Customize Styles and Icons** [Review and Submit](#)

Customize Styles and Icons

Component

[Personalize "Component"](#)
 Select the component to customize. Choose 'global' to customize the set of global styles and icons. If a web bean is not present in the list, then it has no specific styles or icons exposed for customization.

Component

Styles and Icons

[Personalize "Styles and Icons"](#)

[Personalize Header](#)
 Select Show to modify included styles and properties. Select Clear if you do not want a style to inherit values from the base stylesheet. [Personalize Advanced Table: \(NamedStylesTableRN\)](#)

 1-10 of 45

Details	Style	Description	Clear	Delete
<input type="button" value="Show"/>	CenterTextAlign	Sets the text-align property to "center" regardless of the reading direction.	<input type="checkbox"/>	<input type="button" value="Delete"/>
<input type="button" value="Show"/>	RightTextAlign	Sets the text-align property to "right" regardless of the reading direction.	<input type="checkbox"/>	<input type="button" value="Delete"/>
<input type="button" value="Hide"/>	LeftTextAlign	Sets the text-align property to "left" regardless of the reading direction.	<input type="checkbox"/>	<input type="button" value="Delete"/>

Properties

Type	Current Values	New Values	Delete
property	Name text-align Value left	Name <input type="text" value="text-align"/> Value <input type="text"/>	<input type="button" value="Delete"/>
<input type="button" value="Add Property"/>			

<input type="button" value="Show"/>	EndTextAlign	Sets the text-align property to "right" for left-to-right reading direction and "left" for right-to-left reading direction.	<input type="checkbox"/>	<input type="button" value="Delete"/>
<input type="button" value="Show"/>	StartTextAlign	Sets the text-align property to "left" for left-to-right reading direction and "right" for right-to-left reading direction.	<input type="checkbox"/>	<input type="button" value="Delete"/>
<input type="button" value="Show"/>	LightAccentBorder	The lightest border color in the accent color ramp.	<input type="checkbox"/>	<input type="button" value="Delete"/>
<input type="button" value="Show"/>	MediumAccentBorder	A slightly lighter border color in the accent color ramp.	<input type="checkbox"/>	<input type="button" value="Delete"/>
<input type="button" value="Show"/>	VeryDarkAccentBorder	The darkest border color in the accent color ramp.	<input type="checkbox"/>	<input type="button" value="Delete"/>
<input type="button" value="Hide"/>	DarkAccentBorder	The primary border color in the accent color ramp.	<input type="checkbox"/>	<input type="button" value="Delete"/>

Properties

Type	Current Values	New Values	Delete
property	Name background-color Value #CCCCCC <input type="checkbox"/>	Name <input type="text" value="background-color"/> Value <input type="text"/>	<input type="button" value="Delete"/>
<input type="button" value="Add Property"/>			

<input type="button" value="Show"/>	LightBorder	The lightest border color in the core color ramp.	<input type="checkbox"/>	<input type="button" value="Delete"/>
-------------------------------------	-------------	---	--------------------------	---------------------------------------

 1-10 of 45

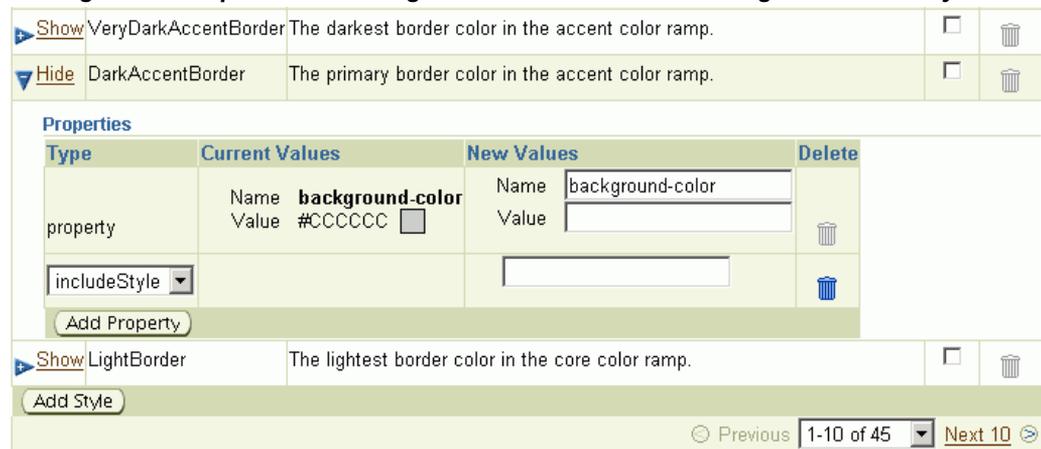
- Modify named styles or selectors - In the Named Styles or Selectors sub tabs, identify the style you want to change. Styles control color or font properties. You can add a new custom style by selecting **Add Style** or delete a custom style by selecting the **Delete** icon.

Select **Show** to expand the detail region for a style or selector to make any of the following modifications:

- Add a new property or included style to a style or selector by choosing **Add Property** and using the **Type** poplist to specify the type of property to add.
- Change the value of any existing property.
- Delete any property or included style from a style or selector .

Note: You cannot directly update an included style. To replace an included style, delete the existing one, then add a new one using the **Add Property** button.

Figure 2-7: Expanded detail region of the DarkAccentBorder global named style.



- Modify icons (icons, page 2-17 may either be text-based or GIF image-based). Select the **Icons** sub-tab and identify the icon you want to modify. Select **Show** to expand the detail region for the icon to make any of the following modifications:
 - Alter the icon's properties, such as height or width.
 - Replace the existing icon with a different icon (such as replacing a text icon with an image icon).

Figure 2-8: Expanded detail region of the global required icon.

ORACLE Customizing Look and Feel Administrator

[Diagnostics](#) [Home](#) [Logout](#) [Preferences](#) [Personalize Page](#)

[Look and Feel Configuration](#) **Customize Styles and Icons** [Review and Submit](#)

[Cancel](#) [Back](#) [Customize Styles and Icons](#) [Next](#)

Component

Personalize "Component"

Select the component to customize. Choose 'global' to customize the set of global styles and icons. If a web bean is not present in the list, then it has no specific styles or icons exposed for customization.

Component

Styles and Icons

[Named Styles](#) [Selectors](#) **Icons**

Personalize Header: [\(IconsRN\)](#)

Select Show to modify included properties. [Personalize Advanced Table: \(IconsRN.IconsTableRN\)](#)

Details	Name	Description
Show	warningAnchor	The warningAnchor icon is a version of the warning icon that is displayed when the icon is used as a link to additional information about the warning.
Show	warning	The warning icon is used by the inline message components (eg. messagePrompt, messageTextInput, etc...) to indicate that an warning message is being displayed to the user.
Hide	required	The required icon is used by the inline message components (eg. messagePrompt, messageTextInput, etc...) to indicate that a value must be specified.
Current Values		New Values
Text-based Icon. Properties not available.		Enter the icon name (like myIcon.gif) for the URI. All icons are picked from \$HTML_TOP/cabo/images/mylaf folder.
	URI	<input type="text"/>
	Height	<input type="text"/>
	Width	<input type="text"/>
Show	infoAnchor	The infoAnchor icon is a version of the info icon that is displayed when the icon is used as a link to additional information.
Show	info	The info icon is used by the inline message components (eg. messagePrompt, messageTextInput, etc...) to indicate that an informational message is being displayed to the user.
Show	errorAnchor	The errorAnchor icon is a version of the error icon that is displayed when the icon is used as a link to additional information about the error.
Show	error	The error icon is used by the inline message components (eg. messagePrompt, messageTextInput, etc...) to indicate that an error has occurred.

[Named Styles](#) [Selectors](#) **Icons**

[Cancel](#) [Back](#) [Customize Styles and Icons](#) [Next](#)

[Diagnostics](#) | [Home](#) | [Logout](#) | [Preferences](#) | [Personalize Page](#)

Copyright 2000-2005 Oracle Corporation. All rights reserved. [Privacy Statement](#)

[About this Page](#)

- If you wish to add or modify named styles, selectors or icons that are specific to a component, use the **Component** poplist to select the component you wish to customize. Once you select a component, the page refreshes with a preview of the component beneath the **Component** poplist.

Figure 2-9: Customize Styles and Icons page with a preview of the contentContainer component

ORACLE Diagnostics Home Logout Preferences Personalize Page

[Look and Feel Configuration](#) **Customize Styles and Icons** [Review and Submit](#)

Customize Styles and Icons Cancel Back Customize Styles and Icons Next

Component

[Personalize "Component"](#)

Select the component to customize. Choose 'global' to customize the set of global styles and icons. If a web bean is not present in the list, then it has no specific styles or icons exposed for customization.

Component

Component: contentContainer

[Personalize Header: \(DisplayCompRN\)](#)
[Personalize Table Layout: \(ButtonsTableLayout\)](#) Refresh

[Personalize Header: \(PageLayoutRN\)](#)

Container Text

Text A

Text B

Styles and Icons

[Personalize "Styles and Icons"](#)

Named Styles [Selectors](#) [Icons](#)

[Personalize Header](#)

Select Show to modify included styles and properties. Select Reset if you do not want a style to inherit values from the base stylesheet. [Personalize Advanced Table: \(NamedStylesTableRN\)](#)

Details	Style	Description	Reset	Delete
Show	ContentContainerLightBackground		<input type="checkbox"/>	
Hide	ContentContainerMediumBackground		<input type="checkbox"/>	

Properties

Type	Current Values	New Values	Delete
includeStyle	MediumAccentBackground		

[Add Property](#)

Show	ContentContainerDarkBackground		<input type="checkbox"/>	
Show	ContentContainerContentBody		<input type="checkbox"/>	
Show	ContentContainerBody		<input type="checkbox"/>	
Show	ContentContainerContent	This named style can be used to specify common properties that are shared by the OraContentContainerContentDark, OraContentContainerContentMedium, OraContentContainerContentLight and OraContentContainerContentTransparent style classes.	<input type="checkbox"/>	
Show	ContentContainer	This named style can be used to specify common properties that are shared by the OraContentContainerDark, OraContentContainerMedium, OraContentContainerLight and OraContentContainerTransparent style classes.	<input type="checkbox"/>	

[Add Style](#)

Named Styles [Selectors](#) [Icons](#)

Cancel Back Customize Styles and Icons Next

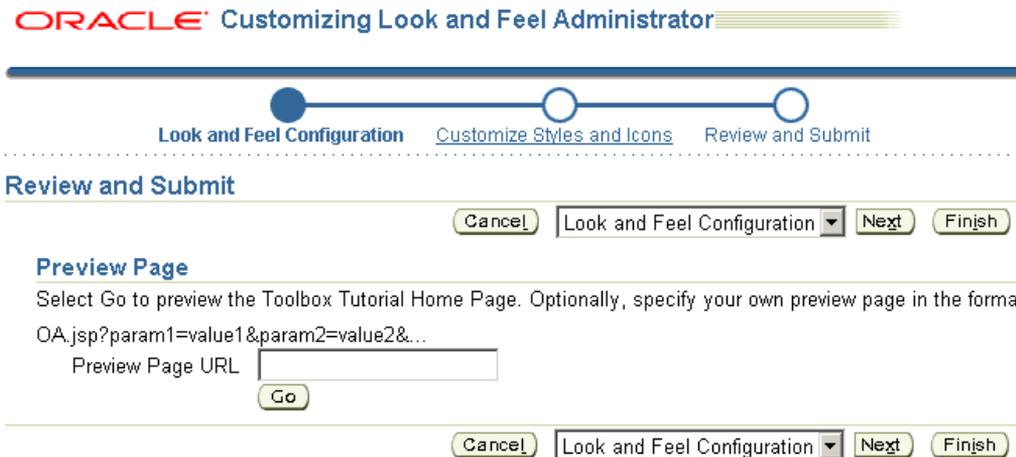
Repeat Step 6, page 2-13 if you wish to modify the component's named styles/selectors or Step 7, page 2-13 to modify the components named icons. Repeat this step for all the components you want to customize. Select Next when you are done.

Note: Any custom template renderers you defined are registered here.

9. In the Review and Submit page, select **Go** if you wish to preview your new custom LAF against the Toolbox Tutorial Home Page that is shipped with OA Framework 11.5.10.

Alternatively, you may specify your own page to preview, using the format `OA.jsp?param1=value1¶m2=value2&...` in the Page Preview URL field and selecting **Go**.

Figure 2-10: Review and Submit page



10. If you are satisfied with the previewed changes, select **Finish** to save your changes and create/register your new custom LAF, otherwise select **Back** to return to the previous pages in the CLAF flow to make additional modifications.
11. To run your Oracle E-Business Suite applications pages with your custom LAF, log in to Oracle Applications and set the profile option **Oracle Applications Look and Feel** (APPS_LOOK_AND_FEEL) to the custom LAF that you created, as it appears in the poplist.
12. Bounce your web server, then run your application pages to see the new LAF take effect.

Updating a LAF

You can also use the CLAF UI to update an existing Look-and-Feel. The UI allows you to:

- Modify any styles or icons defined in the Look-and-Feel.
- Change any renderers registered with the Look-and-Feel.

The following steps outline how to update a LAF or skin using the CLAF user interface:

1. Access the CLAF UI using one of these two methods:
 - Log in as a user with access to the Customizing Look and Feel Administrator responsibility, and select the menu option **Customizing Look and Feel Administrator** to initiate the CLAF UI.
 - In the Page Hierarchy Personalization page, page 4-12 of the Admin Personalization UI, select the **Customize Look and Feel** (pen) icon in the Customize Look and Feel column of the Page Hierarchy HGrid. You can access the CLAF UI using this method if the appropriate conditions, page 2-8 are met.
2. In the Look and Feel Configuration page, select the **Update Look and Feel** radio button.

Figure 2-11: Updating a LAF in the Look and Feel Configuration page

ORACLE Customizing Look and Feel Administrator

[Diagnostics](#) [Home](#) [Logout](#) [Preferences](#) [Personalize Page](#)

Look and Feel Configuration Customize Styles and Icons Review and Submit

Look and Feel Configuration

* Indicates required field Look and Feel Configuration

Create Look and Feel

Update Look and Feel

Update Look and Feel

[Personalize "Update Look and Feel"](#)

Select the Look and Feel you want to update. Optionally add or modify styles for specific browser(s) and/or locale(s). Use comma for delimiting the list of browser and locales.

* Look And Feel Name Look and Feel Configuration

[Diagnostics](#) | [Home](#) | [Logout](#) | [Preferences](#) | [Personalize Page](#)

Copyright 2000-2005 Oracle Corporation. All rights reserved. [Privacy Statement](#)
[About this Page](#)

3. Specify the name of the Look-and-Feel that you want to update, then click **Next**.
4. Refer to Steps 5 and on, page 2-11 in the Creating a New LAF section, page 2-9 for the remaining steps that are also common to updating a LAF.

Custom Renderer and Custom Icon Standards

The following naming and directory standards must be followed:

- All custom template renderer files should be named as `<webBeanType>.uit`. For example: `sideBar.uit`, `tabBar.uit`, or `pageLayout.uit`.
- All template renderer files should reside under `$HTML_TOP/cabo/templates/<lookandFeelId>/` where `<lookandFeelId>` would be a LAF name, such as `custom-desktop`.

- All custom icon files should reside under \$HTML_TOP/cabo/images/<lookandFeelId>/ where <lookandFeelId> would be a LAF name, such as custom-desktop.

Global Styles

The following tables list description of the global styles that are provided by UIX.

Table 2-1: Global alignment styles provided by UIX

Alignment Styles	Description
CenterTextAlign	Sets the text-align property to "center" regardless of the reading direction.
RightTextAlign	Sets the text-align property to "right" regardless of the reading direction.
LeftTextAlign	Sets the text-align property to "left" regardless of the reading direction.
EndTextAlign	Sets the text-align property to "right" for left-to-right reading direction and "left" for right-to-left reading direction.
StartTextAlign	Sets the text-align property to "left" for left-to-right reading direction and "right" for right-to-left reading direction.

Table 2-2: Global color styles provided by UIX

Color Style	Description
LightAccentBorder	The lightest border color in the accent color ramp.
MediumAccentBorder	A slightly lighter border color in the accent color ramp.
VeryDarkAccentBorder	The darkest border color in the accent color ramp.
DarkAccentBorder	The primary border color in the accent color ramp.
LightBorder	The lightest border color in the core color ramp.
MediumBorder	A slightly lighter border color in the core color ramp.
VeryDarkBorder	The darkest border color in the core color ramp.
DarkBorder	The primary border color in the core color ramp.
DisabledLinkForeground	The default foreground color for disabled links.
VisitedLinkForeground	The default foreground color for visited links.
ActiveLinkForeground	The default foreground color for active links.
LinkForeground	The default foreground color for inactive, unvisited links.
ErrorTextForeground	The foreground color for error text (red).
SelectedTextForeground	The foreground color for selected/highlighted text, or text which is drawn on a dark background.
TextForeground	The default text foreground color (black). See the Color Styles section, page 2-28 in the Style Sheets topic, page 2-23 for more information.

Color Style	Description
LightAccentForeground	The lightest foreground color in the accent color ramp. This value is computed relative to the DarkAccentForeground color.
MediumAccentForeground	A slightly lighter foreground color in the accent color ramp. This value is computed relative to the DarkAccentForeground color.
VeryDarkAccentForeground	The darkest foreground color in the accent color ramp. This value is computed relative to the DarkAccentForeground color.
DarkAccentForeground	The primary foreground color in the accent color ramp.
LightForeground	The lightest foreground color in the core color ramp. This value is computed relative to the DarkForeground color.
MediumForeground	A slightly lighter foreground color in the core color ramp. This value is computed relative to the DarkForeground color.
VeryDarkForeground	The darkest foreground color in the core color ramp. This value is computed relative to the DarkForeground color.
DarkForeground	The primary foreground color in the core color ramp.
TextBackground	The default text background color (white). See the Color Styles section, page 2-28 in the Style Sheets topic, page 2-23 for more information.
LightAccentBackground	The lightest background color in the accent color ramp. This value is computed relative to the DarkAccentBackground color.
MediumAccentBackground	A slightly lighter background color in the accent color ramp. This value is computed relative to the DarkAccentBackground color.
VeryDarkAccentBackground	The darkest background color in the accent color ramp. This value is computed relative to the DarkAccentBackground color.
DarkAccentBackground	The primary background color in the accent color ramp. Also known as the Accent background color. See the Color Styles section, page 2-28 in the Style Sheets topic, page 2-23 for more information.
LightBackground	The lightest background color in the core color ramp. This value is computed relative to the DarkBackground color.
MediumBackground	A slightly lighter background color in the core color ramp. This value is computed relative to the DarkBackground color.
VeryDarkBackground	The darkest background color in the core color ramp. This value is computed relative to the DarkBackground color.
DarkBackground	The primary background color in the core color ramp. Also known as the Core background color. See the Color Styles section, page 2-28 in the Style Sheets topic, page 2-23 for more information.

Table 2-3: Global font styles provided by UIX

Font Style	Description
VeryLargeFont	<p>A very large version of the default font.</p> <p>See the Font Size Styles section, page 2-27 and the Font Size Styles for Microsoft Windows Explorer section, page 2-28 in the Style Sheets topic, page 2-23 for more information.</p>
LargeFont	<p>A large version of the default font.</p> <p>See the Font Size Styles section, page 2-27 and the Font Size Styles for Microsoft Windows Explorer section, page 2-28 in the Style Sheets topic, page 2-23 for more information.</p>
MediumFont	<p>A version of the default font which is slightly larger than the default size. This is used for medium sized text, such as level 2 headers.</p> <p>See the Font Size Styles section, page 2-27 and the Font Size Styles for Microsoft Windows Explorer section, page 2-28 in the Style Sheets topic, page 2-23 for more information.</p>
SmallFont	<p>A small version of the default font. This style is used for text which is slightly smaller than the default, such as breadCrumb links.</p> <p>See the Font Size Styles section, page 2-27 and the Font Size Styles for Microsoft Windows Explorer section, page 2-28 in the Style Sheets topic, page 2-23 for more information.</p>
VerySmallFont	<p>A very small version of the default font. This style is used for very small text such as inline messages, and copyright and privacy messages.</p> <p>See the Font Size Styles section, page 2-27 and the Font Size Styles for Microsoft Windows Explorer section, page 2-28 in the Style Sheets topic, page 2-23 for more information.</p>
DefaultBoldFont	<p>A bold version of the default font.</p>
DefaultFont	<p>Specifies the default font for the Look-and-Feel. This style defines both the default font family (as specified by the DefaultFontFamily named style) and the default font size.</p> <p>See the Font Size Styles section, page 2-27 and the Font Size Styles for Microsoft Windows Explorer section, page 2-28 in the Style Sheets topic, page 2-23 for more information.</p>
DefaultFontFamily	<p>Specifies the default font family list ("font-family" property) for the Look-and-Feel. See the Font Styles section, page 2-26 in the Style Sheets topic, page 2-23 for more information.</p>

Customizable Components

The following is a list of components whose Look-and-Feel may be customized using the CLAF UI:

- applicationSwitcher
- breadCrumbs x train
- button
- colorField
- contentContainer

- dateField
- footer
- globalButton
- globalHeader
- hGrid
- header
- hideShow
- lovInput
- messageBox
- separator
- sidebar
- sortableHeader
- tabBar
- table
- train

Note: If a side navigation menu is created by adding functions with prompts to an **HTML Sub Tab** menu instead of an **HTML sideBar** menu, any look-and-feel changes using the CLAF tool on the sideBar component will not be supported on the menu.

Known Issues

See a summary of key CLAF issues, page B-2 with suggested workarounds if available.

Personalization FAQ

The Personalization FAQ includes a number of entries that cover Customizing Look-and-Feel, page D-1 issues.

Icons

For a detailed discussion of icons, refer to the BLAF Icon Specifications [<http://otn.oracle.com/tech/blaf/specs/icons.html>] on the Oracle Technology Network (OTN).

If you need to develop new icons to support your customizations, refer to Icons in BLAF Applications [<http://otn.oracle.com/tech/blaf/specs/icons.html>] and the BLAF Icon Naming Strategy [http://otn.oracle.com/tech/blaf/specs/icons/blaficon_namingstrategy.html] on OTN.

You may also want to look at the Icon Repository for BLAF [<http://otn.oracle.com/tech/blaf/specs/iconList.html>] on OTN to find the base templates that were used to create icons.

Icons are stored as gif image files in the **OA_MEDIA** directory. You can change the image of an icon in one of two ways:

- You can replace the appropriate image file with a file of the same name that contains an updated icon. This updates all instances of that icon globally.
- You can create an administrative-level personalization, page 5-6 for the page by setting the Image URI property on the **image** page element to a different image file. This updates the image for that specific usage of the icon.

Responsibilities and Menus

For a detailed discussion of menus, responsibilities and other Applications security issues, refer to the "Managing Oracle Applications Security" chapter of the *Oracle Applications System Administrator's Guide*.

For a detailed discussion of menus in OA Framework pages, refer to "Menus" in the "Implementing the View" section of the *OA Framework Developer's Guide*.

Using the Oracle Applications Responsibilities screen, you can modify seeded responsibilities or define new ones to create the menu structure and business flows that match your business roles and requirements.

Menus serve two major purposes in Oracle Applications:

- Oracle Applications menus are a navigation tool for the user.
The menus are fully independent of the underlying code, whether it is OA Framework-based or Forms-based. In fact, the same page can be available from multiple menus, where the navigation paths can all vary substantially. You can use the Oracle Applications Menus screen to modify or define new menus.
- Oracle Applications menus are part of the Oracle Applications security model.
A menu must be assigned to a responsibility using the Oracle Applications Responsibilities screen, and a responsibility must be assigned to a valid user using the Oracle Applications Users screen. To see a given menu, you must be logged into Oracle Applications as the appropriate user.

Messages

For a detailed discussion of the Message Dictionary, refer to the *Oracle Applications Developer's Guide*. Also, the *Oracle Applications User Interface Standards for Forms-Based Products* guide has a nice summary of using messages, and contains useful general rules.

You can change messages for OA Framework-based applications in the same way that you change messages for Forms-based applications. If you wish to personalize an error message or other long text, such as a long tip message, you can edit the message in which the error message or long text is stored. To identify the name of the message used for a long tip on a given page, navigate to the Personalize page, page 5-6 in the Admin-level Personalization UI, page 4-1 to identify the values set for the Tip Message Name and Tip Message Appl Short Name properties.

Lookup Codes

Lookup codes which define the display values primarily seen in poplists are handled in OA Framework-based applications the same way that they are handled in Forms-based applications. You can use the Oracle Applications Lookups page to update or define new lookups and lookup codes. For further information, refer to the online help for the

Lookups page. The lookups data is stored in the FND_LOOKUPS table, and you may extract the lookups from there.

Style Sheets

Prior to Release 11.5.10.2CU, page 2-30, OA Personalization Framework used custom style sheets (.xss files) to specify and manage the visual styles of the Look-and-Feel of applications built with OA Framework. One of the goals of custom style sheets was to allow presentation styles, such as fonts and colors, to be separated from the HTML content to which the styles are applied. This enables you to maintain a consistent Look-and-Feel for the application, yet allow you to manage the customizations of styles for different target audiences.

The custom style sheets employ XML Style Sheets (XSS) language, a language that is based on Cascading Style Sheets (CSS), but is designed for easy extraction of style information.

With Release 11.5.10.2CU, OA Framework still uses style sheets, but the style sheets are now associated with a specific Look-and-Feel that you can create or update using the Customizing Look-and-Feel (CLAF) user interface, page 2-4. Creating a custom Look-and-Feel is also referred to as creating a new *skin*. With the new CLAF UI, you no longer need to create or modify custom style sheets directly to alter the styles in your Look-and-Feel. Instead, you use the CLAF UI to create or modify a custom Look-and-Feel (LAF) and modify the styles for that custom LAF.

The following table compares the old custom.xss model with the new CLAF UI model of behavior:

Customizing custom.xss	Customizing Using the CLAF UI
Applicable only over BLAF.	Applicable over any LAF (other than BLAF and minimal).
Allows only stylesheet customization.	Allows stylesheet, icon and layout (renderer) customization.
Cannot define a new skin (custom Look-and-Feel).	Can define a new skin.
Uses the old, deprecated form of style sheet inheritance: blaf.xss -> oa.xss -> custom.xss	Uses the new form of Look-and-Feel inheritance, where one Look-and-Feel can extend another Look-and-Feel; style sheets do not extend other stylesheets.
Desupported by UIX.	Supported by UIX.
Must be used only for minor style (color and font) modifications on top of BLAF.	-

Although you no longer have to work with XSS style sheets directly, the content of this document provides important information about the concepts of XSS that you should know before you start modifying styles in the CLAF UI.

The BLAF (Browser Look-and-Feel) style sheet (blaf.xss) defines Oracle's corporate Look-and-Feel for HTML applications.

Note: If you want to make minor style changes (to font or color) that override BLAF, you must do so by using custom.xss. See [Deprecated](#)

- Custom Style Sheets in OA Framework Prior to 11.5.10.2CU, page 2-30 for details.

Introduction to XSS

An XSS (stylesheet) document consists of a set of style sheets, each of which defines a set of visual styles to be applied to the Look-and-Feel of a web page. Styles control two key aspects of HTML components - color and font. Following is an example of a base style sheet document:

```
<stylesheetDocument xmlns="http://xmlns.oracle.com/uix/style"
  version="2.0">
<!-- Style sheet for our Custom UI -->
<stylesheet>
  <!-- Change the dark color ramp to blue -->
  <style name="DarkBackground">
    <property name="background-color">#00619C</property>
  </style>

  <!-- Override global button text color/font -->
  <style selector=".OraGlobalButtonText">
    <property name="color">black</property>
    <property name="font-weight">bold</property>
    <property name="font-size">80%</property>
    <property name="text-decoration">none</property>
  </style>
</stylesheet>
</stylesheetDocument>
```

Style Sheets

Each style sheet is defined with a `<stylesheet>` element. You can designate which end-user environment to apply a style sheet to by assigning attributes, also called "variants", to the `<stylesheet>` element. XSS supports the following five variants:

- Locale (e.g. `locales="ja"` or `"en_US"`)
- Reading direction (e.g. `direction="rtl"` or `"ltr"`)
- Browser (e.g. `browsers="Netscape"` or `"Internet Explorer"`)
- Browser major version (e.g. `versions="4"` or `"5"`)
- Operating system (e.g. `platforms="Windows"` or `"Linux"`)

A style sheet that contains no variant attributes is known as a base style sheet. The following example shows a style sheet that defines styles for Internet Explorer users in the Japanese locale:

```
<stylesheet locales="JA" browsers="ie">...
</stylesheet>
```

Styles

Each `<stylesheet>` element contains one or more `<style>` elements. In general, a style element is identified by a name and a set of one or more properties as shown below:

```

<style name="styleA">
  <property name="prop1" value="value1" />
  <property name="prop2" />
</style>

```

A property is identified by a name and a value. It defines a facet for the style (for example, font size for a particular font style) and is a CSS property. Generally, an XSS document would have about 100-250 style definitions. Each XSS style definition can use from 1 to 5 CSS properties. There are about 100 valid CSS properties available in all. These properties are listed in the W3C CSS Reference [http://www.w3schools.com/css/css_reference.asp].

Styles are classified into two types:

- Named styles, page 2-25 - which are named bundles of properties that may also include other named styles and is used to define some of the basic units of reusability for the Look-and-Feel.
- Selectors, page 2-25 - which include properties, and/or named styles and is used by a specific web bean to control its Look-and-Feel.

Named Styles

Often, many style definitions share common properties, such as a base font or standard background color. In XSS, you can define a *named style* for such common properties, which can then be referenced by other named styles or selectors in the XSS document. This ability to include a named style allows you to make and maintain simple and concise customizations to the style sheets.

For example, the "DefaultFont" named style below defines the font and font size for two other styles, each of which reference the "DefaultFont" using the `<includeStyle>` element.

```

<stylesheet>
  <style name="DefaultFont">
    <property name="font-family">Arial</property>
    <property name="font-size">10pt</property>
  </style>
  <style selector=".OraFieldText">
    <includeStyle name="DefaultFont">
      <property name="color">#000000</property>
    </style>
  <style name="LinkFont">
    <includeStyle name="DefaultFont">
      <property name="color">#663300</property>
    </style>
</stylesheet>

```

Selectors

A `<style>` element can also be identified by a *selector*, which associates the style with a particular web bean or set of web beans in a HTML document. Selectors are defined by a list of properties and can also reference other named styles using the `<includeStyle>` element. The following example shows the style definition for the HTML OraFieldText element:

```

<stylesheet>
  <style selector=".OraFieldText">
    <includeStyle name="DefaultFont">
      <property name="color">#000000</property>
    </style>
  </stylesheet>

```

In this example, the selector ".OraFieldText" indicates that the properties defined by this style should be applied to any HTML element with a class attribute value of OraFieldText. The style itself is a black Default Font (Arial 10pt) text.

Reusing Styles and Properties

As mentioned above, you can use the `<includeStyle>` element in your style definition to include *all* the properties from another style in your style definition. You can also use the `<includeProperty>` element to include *only a specific property* of another style in your style definition. For example, the following styleA is defined by a property called prop1, all the properties from styleB, and a property called prop3, which is derived from property prop2 of styleC.

```

<style name="styleA">
  <property name="prop1" value="value1" />
  <includeStyle name="styleB" />
  <includeProperty name="styleC" propertyName="prop2"
    localPropertyName="prop3" />
</style>

```

Global Versus Component-Specific Styles

All styles and selectors can be further grouped as global styles or component-specific styles:

- **Global styles** - affect more than one component. Examples of global styles include DefaultFont and LightBackground.
- **Component-specific styles** - affect only one component. An example of a component-specific style is OraContentContainerHeaderText which controls the font properties of the header text of the content container web bean. A set of twenty-one web beans in OA Framework have custom styles defined for them that can be further customized.

Most customizations that you make to a style sheet involve the set of forty-five Global named styles that control about ninety percent of the Look-and-Feel of the application.

Font Styles

One of the most common customizations you may wish to make to the look and feel of the Oracle Applications user interface is to modify the default font. Although a stylesheet may define many different styles that affect font properties, all these styles refer to a single named style that defines the default font family. As a result, if you wish to change the font family properties for all the styles defined in a style sheet, you need to make only a single style override to the DefaultFontFamily style.

Style	Properties	Description
DefaultFontFamily	font-family = Arial, Helvetica, Geneva, and sans-serif	Specifies the default font family list for the Look-and-Feel.

For example, suppose you want to change the default font family for all styles to include only the CSS "serif" font family. In the Customizing Look and Feel UI, page 2-4, you would update your Look-and-Feel by modifying the global **DefaultFontFamily** style such that the value of its **font-family** property is changed to `serif`.

Font Size Styles

Another common customization you may wish to make to the Look-and-Feel of the user interface is to alter the default font size. Almost all the font size styles defined in a style sheet reference a common named style that defines the default font size:

Style	Properties	Description
DefaultFont	includeStyle = DefaultFontFamily font-size = 10pt	Specifies the default font and default font size for the Look-and-Feel.

To change the default font size, in the Customizing Look-and-Feel UI, page 2-4, you would modify the **DefaultFont** style such that the value of its **font-size** property is changed to something like `12pt`.

Style sheets may also make use of several other font sizes. A smaller font size may be used for in line messages, while a larger font size is used for headers, and so on. Each of the different font sizes may be defined in style sheet, as shown:

Style	Properties	Description
VerySmallFont	includeStyle = DefaultFont font-size = -2pt	A very small version of the default font. This style is used for very small text such as inline messages, and copyright and privacy messages.
SmallFont	includeStyle = DefaultFont font-size = -1pt	A small version of the default font. This style is used for text which is slightly smaller than the default, such as breadcrumb links.
MediumFont	includeStyle = DefaultFont font-size = +1pt	A version of the default font which is slightly larger than the default size. This is used for medium sized text, such as level 2 headers.
LargeFont	includeStyle = DefaultFont font-size = +3pt	A large version of the default font. This is used for large text, such as level 1 headers.
VeryLargeFont	includeStyle = DefaultFont font-size = +6pt	A very large version of the default font.

Each of these font size definitions includes the `DefaultFont`, using the `<includeStyle>` element, and then specifies a point size increment or decrement relative to the default font size. By using this XML Style Sheet Language technique, it is possible for you to change all font sizes defined in `blaf.xss` with a single override of the `DefaultFont` style.

Font Size Styles for Microsoft Internet Explorer

The `DefaultFont` style may be defined twice, once in a generic style sheet as described previously, and once in an environment-specific style sheet. The latter style sheet overrides the default font size specifically for the Microsoft Internet Explorer (IE) browser, to take advantage of some IE-specific functionality.

In particular, IE provides a text zooming feature that allows you to scale text to a larger or smaller size, via the **View > Text Size** menu. This functionality does not work with sizes specified in point units, but does work with sizes specified using the CSS "absolute size" keywords. Hence, the IE-specific style sheet uses the "x-small" keyword for its default font size, which is rendered as 10 point text by default. The size is scalable, so the other font sizes defined for IE are specified as a percentage of the default font size.

Style	Properties	Description
<code>DefaultFont</code>	<code>font-size = x-small</code>	Overrides the default font size specifically for the Microsoft Internet Explorer (IE) browser.
<code>VerySmallFont</code>	<code>font-size = 67%</code>	A very small version of the default font specifically for IE.

To change the default font size for IE, use the *Customize Look-and-Feel UI*, page 2-4 to update the Look-and-Feel for the IE browser. Modify the **DefaultFont** style such that the value of its **font-size** property is changed to something like **small**.

Color Styles

The Browser Look-and-Feel makes use of four colors that you may potentially want to change:

- Text foreground color - Color used for almost all text. The default is black.
- Text background color - Color used for almost all content. The default is white.
- Core background color - This is the default blue color that appears throughout the user interface. For example, the color is used as the background color for the selected link in the level one tab bar, as well as the background color of the global header and footer. A style sheet may derive a color ramp that includes lighter and darker shades based on the core background color. For example, a Look-and-Feel may have a light blue foreground color for a selected link in a level one tab bar.
- Accent background color - This is the default tan color that appears through the user interface. For example, the color is used as the background color for the unselected links in the level one tab bar. As with the core background color, style sheets may derive a color ramp based on the accent background color. For example, the dark brown foreground color used for hypertext links, as well as the lighter yellow colors used for action button backgrounds may be variations of the accent background color.

As is the case with fonts, style sheets may use the <includeStyle> element to share the set of color-related named styles with other defined styles. So if you wish to change the colors in the user interface, you need only override these four named styles using the Customize Look-and-Feel UI, page 2-4. These four color styles are defined as follows:

Style	Properties	Description
TextForeground	color = #000000	The default text foreground color (black).
TextBackground	background-color = #ffffff	The default text background color (white).
DarkBackground	background-color = #336699	The primary background color in the core color ramp. Also known as the Core background color.
DarkAccentBackground	background-color = #cccc99	The primary background color in the accent color ramp. Also known as the Accent background color.

The "DarkBackground" and "DarkAccentBackground" styles define the primary colors in the core and accent background color ramps respectively.

If you decide to customize the colors for the user interface, you should try to maintain the contrasts between the text foreground and background colors, as well as between the core and accent colors.

In general, try to select colors from the web safe color palette, as these colors have the most consistent results across the widest range of browsers and platforms. The web safe color palette is a set of 216 colors, where each red, green, or blue component of that color is a multiple of 51 in decimal (0, 51, 102, 153, 204, or 255) or #33 for hexadecimal values (#00, #33, #66, #99, #cc, or #ff).

Also consider, that when you select a new core or accent background color, blaf.xss derives a ramp of lighter and darker shades from that color. As a result, selecting very dark or very light color values may result in less distinction between various darker or lighter shades in the color ramp.

Deprecated - Style Sheet Derivations and Inheritance Hierarchy

Warning: This section is maintained for reference purposes only, because styles in a Look-and-Feel are no longer inherited in this way starting with OA Framework 11.5.10.2CU.

In XSS, with the use of variants, you can define new style sheets that are based on a parent or base style sheet. The derived style sheets implicitly extend the parent style sheet as well as inherit all the styles defined by the parent style sheet. A derived style sheet may also override styles inherited from ancestors.

If the same style is defined in multiple style sheets and is applicable to the current end-user environment, a set of rules apply to determine which style takes precedence. The rules are based on the variants listed below, shown in order of highest to lowest precedence:

- Locale
- Reading direction

- Browser
- Operating system
- Browser version

For example, suppose you have the following style sheet:

```
<stylesheet locales="JA" browsers="ie">
...
</styleSheet>
```

Without precedence rules, either of the following two style sheets could be its parent:

- ```
<!--Style sheet for Japanese-->
<stylesheet locales="JA">
...
</styleSheet>
```
- ```
<!--Style sheet for Internet Explorer-->
<stylesheet browsers="ie">
...
</styleSheet>
```

With precedence rules, the complex inheritance hierarchy is simplified so that the derived style sheet has only a single parent. In this example, the Japanese for Internet Explorer style sheet inherits styles from the parent Japanese style sheet, which in turn inherits styles from its parent Internet Explorer style sheet.

Deprecated - Custom Style Sheets in OA Framework Prior to 11.5.10.2CU

Warning: This section is maintained for reference purposes only, and if you want to override a few minor styles in BLAF.

In OA Framework, style definitions for the Look-and-Feel are defined by the *blaf.xss* style sheet document.

Prior to OA Framework 11.5.10.2CU, specific Oracle Applications products could modify or extend those default styles to address product-specific requirements for their audience. These product-specific styles were defined in the *oa.xss* style sheet document, which included *blaf.xss*.

As a customer, you could also modify existing styles in *blaf.xss* or *oa.xss*, or add new styles to suit your needs. Rather than directly modify *blaf.xss* or *oa.xss*, you placed all your style sheet customizations in a style sheet document called *custom.xss*, located in *OA_HTML/cabo/styles*.

Following is the content of the file. Only those sections of the file designated for customizations should be edited.

```

<?xml versions="1.0"?>
<stylesheetDocument xmlns="http://...">
<import href="oa.xss"/>

<stylesheet>
<!--Please start your customizations of the base-->
<!--stylesheet here-->

<!--Please end your customizations of the base-->
<!--stylesheet here-->

</stylesheet>

<!--Please start browser or platform specific style-->
<!--sheet customizations here-->

<!--Please end browser or platform specific style-->
<!--sheet customizations here-->
</stylesheetDocument>

```

OA Framework always called custom.xss as its main style sheet. The document custom.xss used the <import> element to include all the styles defined in oa.xss, as illustrated above. The document oa.xss, in turn, used the <import> element to include all the styles defined in blaf.xss.

With this hierarchy, you were able to make customizations to "base" styles without altering base style definitions. In addition, if Oracle ever updated oa.xss or blaf.xss, your customizations to custom.xss were preserved and automatically reflected any updates made to the named styles that you included in your custom styles.

All style sheet documents resided under OA_HTML/cabo/styles.

Caution: The derivation of styles through the described inheritance of style sheets (blaf.xss -> oa.xss -> custom.xss) is deprecated as of OA Framework 11.5.10.2CU. OA Framework now derives styles from the inheritance of Look-and-Feel (LAF) definitions where a Look-and-Feel can extend another Look-and-Feel. Refer to the Customizing Look-and-Feel section, page 2-4 for more details about the current behavior.

CSS Style Lookup

You can use an interactive user interface called the Item Style and Style Type Selection page to preview a specific item style with a selected CSS Style applied to it. You should use this feature to simulate the Look-and-Feel of an item style before making the actual CSS style change to your pages with OA Personalization Framework, page 4-1.

Figure 2-12: Item Style and Style Type Selection page

ORACLE [Diagnostics](#) [Home](#) [Logout](#) [Preferences](#) [Personalize Page](#)

Item Style and Style Type Selection

[Personalize Region](#)

Please select an Item Style and Style Type and click the "Go" button to view the Item Style rendered with the chosen Style(s) applied. If no Style is selected, the Item Style is rendered with all Styles applied.

* Item Style

Style Type

Results of the Selection

The following table shows the Item Style selected with the Style(s) applied.

Style that has been applied	Item style with the selected style applied
BscTabNameText	Sample Label <input type="checkbox"/> Submit

[Diagnostics](#) | [Home](#) | [Logout](#) | [Preferences](#) | [Personalize Page](#)

Copyright 2000-2004 Oracle Corporation. All rights reserved. [About this Page](#) [Privacy Statement](#)

To access the Item Style and Style Type Selection page:

1. Sign on to Oracle Applications and select the System Administration responsibility.
2. Use the Oracle Applications Menus form to add the function **FWK_CSS_LOOKUP** to your System Administration menu.

To preview an Item Style with different CSS styles applied to it:

1. Sign on to Oracle E-Business Suite and select the System Administration responsibility.
2. Select **CSS Style Lookup** from the menu.
3. In the Item Style and Style Type Selection page, there is a required **Item Style** poplist and a **Style Type** poplist. Use the **Item Style** poplist to select an item style for which a CSS Style may be applied. Use the **Style Type** poplist to select a CSS Style to apply. The **Style Type** poplist lets you select from all the CSS styles that have a selector defined in custom.xss. If you select *All Styles*, the selected item renders with all the available CSS styles applied.
4. Choose **Go** to apply the CSS style(s). The results are displayed in a table with two columns. The first column lists the name of the CSS style that has been applied and the second column renders the item style with the CSS style applied to it.

Note: If the selected Item Style is Text Input or Static Styled Text, the results table displays three columns. The first column always lists the name of the CSS style that has been applied. For Text Input, the second column renders the Text Input with the applied CSS style, and the third column renders the Text Input as a multi-line Text Area with the applied CSS style. For Static Styled Text, the second column renders the Static Styled Text with the applied CSS style, and the third column renders the Static Styled Text as a URL with the applied CSS style.

Personalizing Your Pages and Portlets

This chapter covers the following topics:

- Overview
- Personalization Levels
- OA Personalization Framework Features
- Profile Options Used By OA Personalization Framework

Overview

OA Framework includes the OA Personalization Framework which allows you to personalize an Oracle E-Business Suite application page without modifying any underlying code. Personalizing the appearance of or the data displayed in an OA Framework-based page is easy and straightforward.

Important: To personalize your OA Framework-based applications, we recommend and support only the use of OA Personalization Framework.

Dynamically created pages without OA Extension metadata behind them are not personalizable with the OA Personalization Framework.

Caution: A page may contain logic that depends on the regions or items that you personalize. To avoid unintended consequences, you should always familiarize yourself with the page's logic and thoroughly test your personalization before deploying it to a production environment.

All personalizations you make through the OA Personalization Framework are added on top of the base product meta data at runtime. These personalizations never overwrite the existing base product UI and are therefore preserved during upgrades and patches, and can also be translated. This means you can create your personalizations on a test system first, extract your personalizations to a flat file, page 9-1, then load your finalized personalizations to your production system with little interruption.

With OA Personalization Framework, your personalizations are reflected immediately on the page.

To begin personalizing your pages and portlets, refer to the instructions in these chapters:

- Administrative-Level Personalizations, page 4-1
- User-Level Personalization User Interface, page 6-1

- Portlet-Level Personalization User Interface, page 7-1

Personalization Levels

OA Personalization Framework supports end-users as well as localization and customization teams in their efforts to tailor OA Framework-based applications to different users. OA Personalization Framework accomplishes this by allowing you to make personalizations at distinct levels so that you may target those personalizations to specific audiences.

There are different personalization levels available from the system administrative standpoint: Function, Industry, Localization, Site, Organization, Responsibility, and Admin-Seeded User. When you make personalizations at any of these levels, the personalizations are available only to the audience defined by that level. Since personalizations should only be made at these levels by a system administrator, these are collectively referred to as *administrative-level* (or admin-level) personalizations. Admin-level personalizations can be performed on any component in a page, including shared (extended) regions. Except for the Admin-Seeded User level, you can only have one set of personalizations per region per admin-level.

User-level or *portlet-level* personalizations, on the other hand, can be made only to certain tables in query regions or portlets. User or portlet-level personalizations can be made directly by an end-user and are visible only to that end-user, hence they are collectively referred to as *end user-level personalizations*.

All personalization levels are described in the following sections.

Function Level

A function in Oracle Applications is a token that is registered under a unique name for the purpose of assigning it to, or excluding it from, a responsibility. The OA Personalization Framework leverages the same infrastructure to drive the Personalization context at a feature or flow level. You can create personalizations for a region at the Function level, such that the personalizations are visible only when you display the region while the specific function is in context. For example, suppose you have an updatable region and you want that region to be updatable when accessed from FunctionA in the menu, but to be read-only when accessed from FunctionB in the menu. To accomplish this, you create a Function level personalization that makes all the fields read-only for FunctionB.

You can have only one set of personalizations per region per function; however, you can have as many functions as personalizations are needed, provided that these functions can be brought into context at runtime, such as by linking them to menus or passing them as parameters on the URL.

Industry Level

With the Industry personalization level, the OA Personalization Framework provides a way to tailor an application's user interface to suit a particular industry segment.

The Industry level includes a predefined list of vertical market categories, such as Healthcare - Provider and Utilities - Electricity. For each category, you can personalize the application interface to incorporate a unique combination of industry-specific terminology. For example, where one type of business might use

the term "business unit," another might use the term "store," and another might use "location" or "franchise."

Localization Level

Suppose you need to distribute your applications to a particular locale of users, where certain fields or buttons are hidden and labels need to be changed to accommodate that locale. You can do that by creating Localization level personalizations in the relevant regions before delivering your applications. All end-users for the specific locale will see the applied localization personalizations. For example, Oracle's localization teams would make country-specific localization-level personalizations in HR applications before delivering the applications to customers in a given country.

Site Level

A site refers to an installation of Oracle Framework-based applications. Personalizations that you make at the site level affect all users of the current installation. For example, as a system administrator, you might want to make a site level personalization where you change the table column labels to match your corporate standards.

Organization Level

An Org is an Organization or a business unit. Depending on the context, an Org can be a plant, a warehouse, a department, a division within a company or even a complete company. Personalizations that you make at the Org level affect all users of that Org. For example, you might make a personalization at the Org level to hide certain fields because they are not pertinent to the context of a particular Org.

Responsibility Level

A responsibility represents a specific level of authority within an application. Each responsibility lets you access specific functions or a set of product pages, menus, reports and data to fulfill your role in an application. When you make personalizations at the responsibility level, the changes are effective only for the users of a given responsibility. For example, you can personalize the Open Requisitions Line page for the "Office Supplies Purchasing Manager" responsibility to display only open requisition lines from a particular supplier.

Admin-Seeded User-Level

As an administrator you may want to create some personalizations that are available to all your users and allow your users to choose whether they want to use those personalizations. You can accomplish this by creating an "admin-seeded user-level" personalization, also known as an "admin-seeded" end user view. For example, you can create two personalized views of the Oracle Workflow Worklist. One view shows open workflow notifications and the other view shows FYI notifications. Each user can have access to both of these views.

You can only create "admin-seeded user-level" personalizations on tables (including hierarchy tables) in a Query region. You can also secure "admin-seeded user-level" to a specific function so that it is published only to a specific group of users. These personalizations then get seeded into the appropriate users' Personal Table Views page, so that individual users can choose which views to display. You can have more than one "admin-seeded user-level" personalization per region.

Note: Note that although both "admin-seeded user-level" personalizations and site-level personalizations are propagated to all users, the two personalization levels are different. An "admin-seeded user-level" personalization is saved as a personalized view that a user can choose to display from the Personal Table Views page, whereas a site-level personalization is a change that is made across the entire site that all users see automatically.

Before OA Framework 11.5.57, only Oracle-internal developers were able to create seeded user-level personalizations. Now, however, customers may also create seeded user-level customizations. To distinguish between seeded user-level personalizations that are shipped by Oracle and those that are defined by administrators at a customer site, the two types of personalizations are referred to as "Oracle-seeded user-level" personalizations (seeded by Oracle) and "admin-seeded user-level" personalizations (seeded by administrators at the customer site), respectively.

Note: An "Oracle-seeded user-level" personalization cannot be updated or deleted at the customer site; however, an administrator at a customer site who creates an "admin-seeded user-level" personalization can edit or delete that personalization.

Note: As of OA Framework 11.5.10, you can mark a seeded user-level personalization as a default view. End-user personalized views take highest precedence, followed by "admin-seeded user-level" personalizations, then "Oracle-seeded user-level" personalizations.

Portlet Level

Oracle Portal provides users with corporate and customized personal home pages accessible via web browsers. These home pages may contain corporate announcements, stock tickers, news headlines, and links to other web-based services. Oracle Portal may also connect to partner applications that share their user authorization and session management models with Oracle Portal. Oracle Applications is a partner application to Oracle Portal. Oracle Portal users can add links to their home pages to access Oracle Applications modules, and can display Applications information, such as Oracle Workflow notifications, directly on their home pages. These links are called portlets. You can personalize these portlets just as you can personalize the tables of a query region in an application. Any personalizations you make at the portlet level affect only the portlet used to display the region. You can have only one set of personalizations per region per portlet.

User Level

As an end-user, you can personalize certain tables in query regions and the personalizations would affect no one else. For example you can personalize the Requisitions History page to display only requisitions prepared by you. Each user can save multiple sets of personalizations per page region. A saved set of personalizations is also known as a personalized "view" and can be selected and applied from the "View Personalizations" list.

OA Personalization Framework Features

Cumulative Personalizations

The personalizations you make at the various levels are cumulative. Personalizations at the higher levels typically apply to broadly defined user populations, and those at lower levels apply to more narrowly defined groups. Personalizations made at lower levels are applied after those made at higher levels. The personalization levels are ordered as follows, from highest to lowest:

- Function (highest)
- Industry
- Localization
- Site
- Organization
- Responsibility
- Seeded-User
- Portlet (lowest)
- User (lowest)

For example, the personalizations made at the Portlet and User levels always overlay personalizations made at the Organization level.

Note: Personalizations made at the Portlet and User levels are mutually exclusive.

For Oracle-internal developers, Function is the highest level at which you can personalize, followed by Industry, Localization, Site, and so on.

The granular nature of each personalization is maintained throughout the layering of personalizations for the different levels. For example, suppose you make a set of Site level personalizations to a region by changing the label of four fields from a, b, c and d to w, x, y, and z. Now suppose you want to make additional changes to that same region, where those changes are available only for users of Org 2. The Organization-level change for Org 2 that you make includes hiding the second and third fields and changing the label of the last field to zz. A user of Org 2 would see, as a result of the cumulative personalizations, a region with two fields labeled w and zz.

Personalization Ownership

Personalizations made at an Admin level can only be modified, viewed, or deleted by an administrator at that respective level. Similarly, a personalized view made at the Portlet or User level can only be modified, viewed, or deleted from the portlet where the view was created or by the user who created the view.

Multiple Personalizations

At the User level, a particular region of an HTML page may have one or more sets of personalizations associated with it. Each set of personalizations is referred to as a personalized view of the region. You can identify a personalized view by assigning a

view name to it. OA Personalization Framework also identifies each personalized view by assigning a unique ID to it.

Default Views

Since you may define more than one personalized view at the User level, OA Personalization Framework lets you mark a specific view as the default view to apply to the region when you run the application.

Note: In the case of seeded personalized views, OA Personalization Framework applies a precedence rule to determine the default, if multiple defaults are defined (for example, if defaults exist at the User level, "Admin-seeded user level" and "Oracle-seeded user level"). The User-level default overrides any default set at any level. If no user-level view is marked as a default, then the "admin-seeded" default view takes precedence. Similarly, if no user-level or "admin-seeded user-level" views are marked as defaults, the "Oracle-seeded" view becomes the default.

In the case where personalizations are made on a region that extends another region, at the User, "Admin-seeded user" or "Oracle-seeded user" levels, the default is resolved horizontally across the layers of the region. All of the default User views take precedence, followed by the default admin-seeded views, then the default Oracle-seeded views.

Example

Suppose you have region A, which extends region B, which in turn extends region C, and the following views have been created and marked as "Default":

Region	Views Defined
A (extends B)	A.UserView - default A.Admin-seededUserView - default A.Oracle-seededUserView - default
B (extends C)	B.UserView - default B.Admin-seededUserView - default B.Oracle-seededUserView - default
C	C.UserView - default C.Admin-seededUserView - default C.Oracle-seededUserView - default

The final default view is resolved based on the following order of precedence:

```
A.UserView >
  B.UserView >
    C.UserView >
      A.Admin-seededUserView >
        B.Admin-seededUserView >
          C.Admin-seededUserView >
            A.Oracle-seededUserView >
              B.Oracle-seededUserView >
                C.Oracle-seededUserView
```

Personalization Caching

The personalizations that you make to a region do not cause any performance degradation in your application. Admin-level personalizations for a region are statically cached on the JVM (Java Virtual Machine) and user-level personalizations are cached on the session.

Profile Options Used By OA Personalization Framework

The following system profile options affect the behavior of the OA Personalization Framework.

Disable Self-service Personal (FND_DISABLE_OA_CUSTOMIZATIONS)

This is a system profile option specifically created for use by Oracle Support. You can set this profile option to `Yes` at the site or application level. If this system profile option is set to `Yes`, any personalizations you make, regardless of the level at which you make the personalizations, will not be applied. Only the original base definition of each OA Framework-based application page is ever displayed. Use this profile option to help Oracle support staff determine whether a problem arises from applied personalizations or from the base application. The default is `No`.

FND: Personalization Region Link Enabled (FND_PERSONALIZATION_REGION_LINK_ENABLED)

When you set this profile option to `Yes`, **Personalize Region** links appear above each region in a page. Each link takes the user first to the Choose Personalization Context page, page 5-4, then to the Page Hierarchy Personalization page, page 4-12 with focus on the region from which you selected the **Personalize Region** link. The default is null (not set).

Note: Note that enabling the **Personalize Region** links allows administrators to also personalize regions that are dynamically added to the page from custom code in the controller.

Personalize Self-service Defn (FND_CUSTOM_OA_DEFINITION)

You should set this profile option to `Yes` at the user level for an administrator. If you set this profile option to `Yes`, then when you log on as the Administrator, a global **Personalize Page** button appears on each OA Framework-based application page. When you select the global **Personalize Page** button on any page, the personalization user interface prompts you for the scope and administrative level at which you wish to create your personalizations before displaying the OA Personalization Framework UI. The default is `No`.

Xliff Export Root Path (FND_XLIFF_EXPORT_ROOT_PATH)

Use this profile option to specify the root path for extracting translated personalizations. The Xliff Export root path is used to generate the full path where the Xliff files are exported when administrators extract their translated personalizations from the Extract Translation Files page, page 5-37 in OA Personalization Framework. The default is null (no root path is set).

Xliff Import Root Path (FND_XLIFF_IMPORT_ROOT_PATH)

Use this profile option to set the root path for uploading translated personalizations. The Xliff Import root path is used to derive the full path from where the Xliff files are

uploaded when administrators use the Upload Translations page, page 5-38 in OA Personalization Framework to upload translated personalizations. The default is null (no root path is set).

Administrative-Level Personalizations

This chapter covers the following topics:

- Overview
- Creating Admin-Level Personalizations
- Function-Level Personalizations
- Known Issues
- Related Information

Overview

Administrative-level (admin-level) personalizations are performed by Oracle Application administrators who are often department-level functional consultants or administrators who understand, design and tailor the functional business flows of their organization. As an Oracle Applications Administrator, you can use OA Personalization Framework to personalize the pages of OA Framework-based applications at various personalization levels, page 1-4 without modifying any code. Refer to the Customization Primer, page 1-1 and the discussions of Personalization Levels, page 3-2 and OA Personalization Framework Features, page 3-5 for an overview of what you can accomplish.

Note: The term "Administrator" used throughout this document refers to the Oracle Applications Administrator described above and not to the System Administrator who configures and maintains the application system.

Note: If you use Netscape 4.7 or earlier or if you turn Partial Page Rendering (PPR) off by setting the **FND: Disable Partial Page Rendering** profile value to *Yes*, you may see additional **Go** buttons rendered throughout the OA Personalization Framework user interface. These **Go** buttons allow you to refresh parts of the page with changes that PPR would have otherwise taken care of.

Personalization Profile Options

Before you can personalize a region at a personalization level, you must set the **Personalize Self-service Defn** (FND_CUSTOM_OA_DEFINITION) profile option to *Yes* for the user you are signing in as. Every OA Framework-based application page contains a global **Personalize Page** link. This profile option enables the global **Personalize Page** link on every page so that you can personalize any page or region. The **Personalize Page**

link first navigates to the Choose Personalization Context page, page 5-4, then to the Page Hierarchy Personalization page, page 4-12 or to the Page Layout Personalization page, page 4-5, where you can drill down to the region you wish to personalize.

Additionally, you may set the **FND: Personalization Region Link Enabled** (FND_PERSONALIZATION_REGION_LINK_ENABLED) profile option to *Yes* or *Minimal*, to enable **Personalize Region** links for each region in a page. When this profile is enabled, **Personalize Region** links appear above each region, unless the region is a table, hierarchy grid, or table layout region, in which case the link appears below that region. Rather than use the global **Personalize Page** link on every page, you can use any of the **Personalize Region** links to first navigate to the Choose Personalization Context page, page 5-4, then to the Page Hierarchy Personalization page, page 4-12 with immediate focus on the region from which you selected the **Personalize Region** link.

Note: Enabling the "Personalize Region" links allows users to also personalize regions that are dynamically added to the page from custom code in the controller.

For more information about the FND: Personalization Region Link Enabled profile option, refer to the **Personalization** section in the OA Framework Profile Options appendix of the *Oracle Application Framework Developer's Guide*.

Tip: You may also want to set the **FND: Diagnostics** (FND_D IAGNOSTICS) profile option to *Yes* to render the "About this page" link at the bottom of each OA Framework-based page. The About this page link takes you to the "About" page which displays information about the current page, the page's context, and the environment. The Personalization subtab on the "About" page also identifies personalizations that are in effect on the current page for the current personalization context and provides a **Manage Personalization Level** button that lets you navigate to the Manage Personalization Levels page, page 5-36 directly to recover from errors that result from a personalization.

For more information about the FND: Diagnostics profile option, refer to the **Logging / Diagnostics** section in the OA Framework Profile Options appendix of the *Oracle Application Framework Developer's Guide*.

For more information about the "About" Page, refer to the **Discovering Page, Technology Stack and Session Information** section in the Testing and Debugging chapter of the *Oracle Application Framework Developer's Guide*.

Creating Admin-Level Personalizations

As an administrator, there are two ways in which you can initiate Admin-level personalizations. You can either search for the page using the Functional Administrator Home page, page 4-2 and personalize the page without running it, or run the actual page in Oracle E-Business Suite and initiate the Personalization UI from the global Personalize Page link, page 4-4 on the rendered page.

From the Functional Administrator Responsibility

To initiate OA Personalization Framework from the Functional Administrator tool in Oracle E-Business Suite:

1. Sign on to Oracle E-Business Suite and under the Functional Administrator responsibility, select the Functional Administration Home page. Refer to the *Oracle Applications System Administrator's Guide* for additional information about the Functional Administrator feature.
2. Select the **Personalization** tab to display the Application Catalog.
3. Specify an **Application** and optionally, a **Document Path**, to search for the page or pages you wish to personalize.
 - Note:** This search does not use a standard Query region -
 - The Document Path field has no support for wildcards.
 - If you specify a package in the Document Path field, the results will list all of the documents in the repository under that package and its subpackages.
4. Select the **pen** icon in the **Personalize Page** column for the page you want to personalize. This icon directs you to one of the following two personalization launch pages, depending on the page you are personalizing:
 - **Page Layout Personalization** - if the page you are personalizing is a configurable page, you navigate to this page.
 - Warning:** You should not personalize a configurable page by accessing it from the Functional Administrator responsibility. Your configurable page might contain controller code that is dependent on certain parameters that are available only when the page itself is launched from its intended flow, and the page might fail with unexpected errors. You can safely personalize your configurable page by accessing the Page Layout Personalization page using the global Personalize Page link, page 4-4 on the rendered configurable page.

For additional information, refer to the section on Configurable Pages, page A-4 in the Personalization Considerations appendix.
 - **Page Hierarchy Personalization** - if the page you are personalizing is not a configurable page, you navigate to this page. Like the Page Layout Personalization page, the Page Hierarchy Personalization is also a launch pad for personalization options, but it displays the entire structure of the page in a Hierarchy table (HGrid) rather than as a visual boxed layout. Refer to the Page Hierarchy Personalization page, page 4-12 for an example of this page and to the Page Hierarchy Task Flows section, page 4-14 for a list of the tasks that you can launch from this page.
5. Select the **pen** icon in the Manage Personalizations column to navigate to the Manage Personalization Levels page, page 5-36 where you can activate/inactive/delete specific personalizations or manage the translation of the personalizations made for the page in question.
6. If you wish to personalize the Standard Footer that appears in OA Framework-based pages, select the **Standard Footer** link under the **Personalizations Shortcuts**. This link takes you to the **Choose Personalization Context** page, where the Scope is set to **Region: Standard Footer**. When you choose Apply on this page, you navigate to

the Page Hierarchy Personalization page, where you can personalize the Standard Footer.

From the Personalize Page Global Link

To initiate OA Personalization Framework from the global Personalize Page link:

1. Sign on to Oracle E-Business Suite as an administrator and navigate to the application page that you wish to personalize.
2. Choose the **Personalize Page** global link.
3. Prior to OA Framework 11.5.10.2CU, this button first took you to the Choose Personalization Context page, page 5-4 where you could define the context for the personalizations you were about to make. Starting in OA Framework 11.5.10.2CU, this button now defaults to the current personalization context and takes you directly to one of the following two personalization launch pages, depending on your page of origin or your Accessibility mode:
 - **Page Layout Personalization** - if the page you are personalizing is a configurable page, you navigate to this page. A *configurable page* is a page designed with personalizable layout components called flexible layout regions and self-contained content components called flexible content regions. These flexible regions support a more graphical approach to rearranging the layout and visibility of page content.

The Page Layout Personalization page is a visual launch pad from which you can personalize a configurable page. It provides a boxed preview of the flexible layout structure within your page and displays controls that take you to different pages or flows where you specify and apply your actual personalizations. Refer to the Page Layout Personalization page, page 4-5 for an example of this page and to the Page Layout Task Flows section, page 4-5 for a list of the tasks that you can launch from this page.

- **Page Hierarchy Personalization** - if the page you are personalizing is not a configurable page or if you are running Oracle Applications in Accessibility mode (that is, the profile option **Self Service Accessibility Features** (ICX_ACCESSIBILITY_FEATURES) is set to Standard Accessibility), you navigate to this page. Like the Page Layout Personalization page, the Page Hierarchy Personalization is also a launch pad for personalization options, but it displays the entire structure of the page in a Hierarchy table (HGrid) rather than as a visual boxed layout. Refer to the Page Hierarchy Personalization page, page 4-12 for an example of this page and to the Page Hierarchy Task Flows section, page 4-14 for a list of the tasks that you can launch from this page.

Note: If there is an problem with the current personalization context, OA Personalization Framework will display the Choose Personalization Context page, showing the error message above the page.

From a Personalize Region Link

To initiate OA Personalization Framework from a Personalize Region link, page 4-2:

1. Select the **Personalize Region** link for the region you wish to personalize.

2. Prior to OA Framework 11.5.10.2CU, this button first took you to the Choose Personalization Context page, page 5-4 where you could define the context for the personalization you were about to make. Starting in OA Framework 11.5.10.2CU, this button now defaults to the current personalization context and takes you directly to the "complete" view of the Page Hierarchy Personalization page, page 4-12 with immediate focus on the region from which you selected the **Personalize Region** link.

Note: If there is a problem with the current personalization context, OA Personalization Framework will display the Choose Personalization Context page, showing the error message above the page.

Using the Page Layout Personalization Page

The Page Layout Personalization page provides a WYSIWYG user interface for personalizing a configurable page or region.

Note: If you personalize a non-configurable page (a page that does not have the necessary metadata to personalize its layout) or when you run your application in Accessibility mode (profile option **Self Service Accessibility Features (ICX_ACCESSIBILITY_FEATURES)** is set to `Standard Accessibility`), you will not see the Page Layout Personalization page, as shown below. Instead, you will see the Page Hierarchy Personalization page, page 4-12, where you can perform the same personalization functions, using a descriptive tabular user interface.

Note: With the Page Layout Personalization page, you can launch most of the personalization operations from the Layout mode or Content mode of the page. You can also personalize individual properties of a page element through the focused version of the Page Hierarchy Personalization page, page 5-2 that you navigate to by selecting the Personalize icon on a boxed region in the Page Layout Personalization page.

As of OA Framework 11.5.10.2CU, the Page Layout Personalization page has two modes, Layout, page 4-5 and Contents, page 4-8, which you can access from the subtabs on the page. These modes organize the personalization tasks that you can perform so that they correlate directly to the flexible layout or flexible content regions on the page.

The **View Level** poplist that appears for both Layout and Contents mode, allows you to select a single personalization level from your specified context for which you wish to personalize the layout. Note that even though you can choose more than one personalization level in the Choose Personalization Context page, OA Personalization Framework only allows you to save a configurable page personalization at one level. The default value of the View Level poplist is the lowest applicable level specified in your context.

Layout Mode

The Layout mode of the Page Layout Personalization page renders boxes around each region on the page regardless of whether the region is a flexible layout or not. Control icons that allow you to configure a region, however, are displayed only for boxed flexible layout regions. The actual content of a boxed flexible layout region is not displayed. Note that in this particular example, the instruction text and region immediately below the

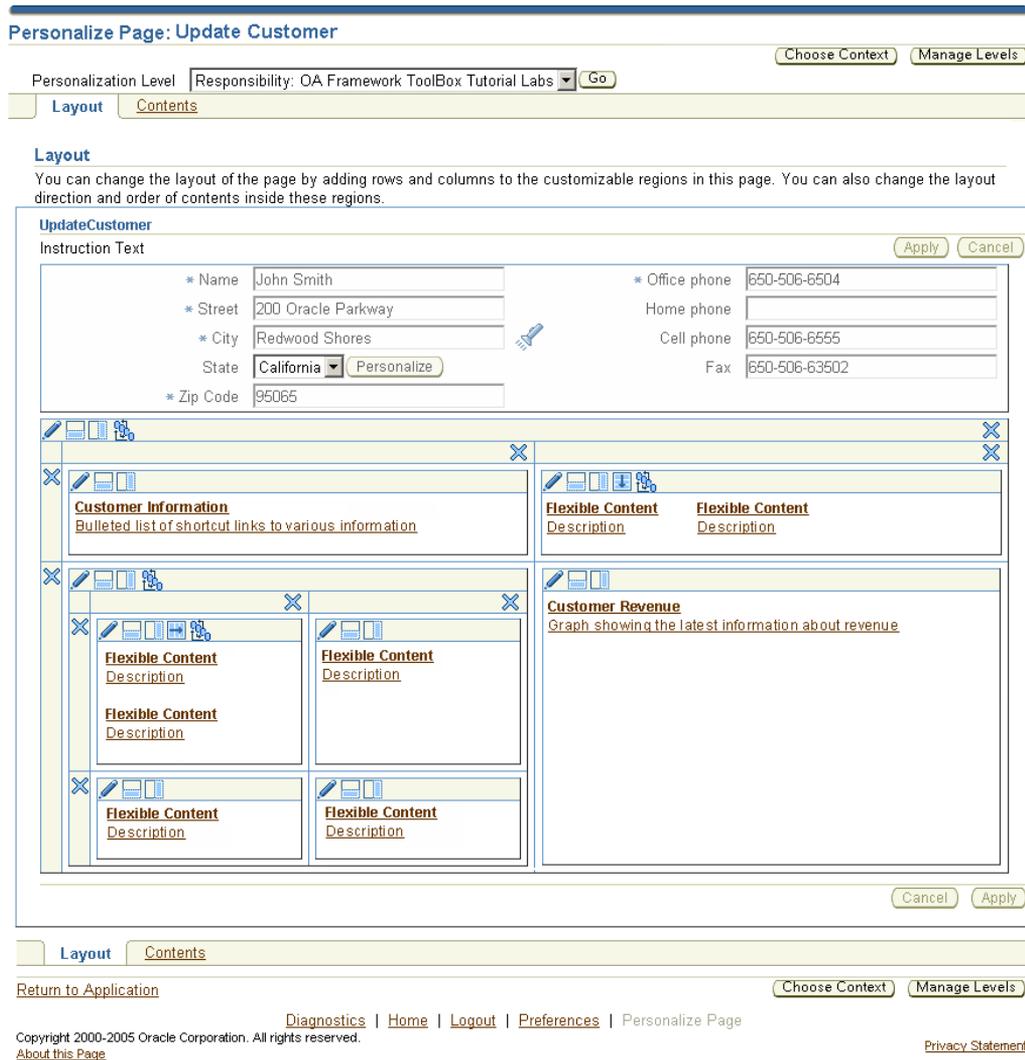
Update Customer header is not a flexible layout region, therefore its content is fully displayed.

In the Layout mode, flexible layout content is simply represented as a header title within each flexible layout box. This frees up screen space so that all nested levels of flexible layout regions can be represented on the page without further drilling.

The Layout Mode Task Flows section, page 4-6 below describes the personalizations you can accomplish in the Layout Mode.

If you wish to exit from the Page Layout Personalization page, select the **Return to Application** link at the bottom of the page to return to the original page.

Figure 4-1: Layout Mode of the Page Layout Personalization page for the Update Customer page.



Layout Mode Task Flows

The following table lists the personalization tasks you can accomplish from the Layout Mode of the Page Layout Personalization page.

Task	Launch Point from Layout Mode	Description
Personalize the properties (such as Prompt name, default value, and so on) of a specific page element		<ol style="list-style-type: none"> 1. Launches a focused version of the Page Hierarchy Personalization page, page 5-2 for the selected region. The Personalization Context shown at the top of the page lists the personalization context for the top-level object, which in the case of Oracle Applications, is the page layout. 2. Select the pen icon from the Personalize column to navigate to the Personalize page, page 5-6 for the page element. 3. In the Personalize page, you can edit the personalizable properties of the element at the personalization levels specified for the current context. 4. When you are done, select Apply to save your personalization or select Apply and Personalize Another to save your personalization and return to the focused Page Hierarchy Personalization page where you can select another element to personalize.
Alter the layout of a boxed region		Splits the current flexible layout region vertically into two peer layout regions, displayed as rows. This change is automatically committed.
Alter the layout of a boxed region		Splits the current flexible layout region horizontally into two peer layout regions, displayed as columns. This change is automatically committed.
Alter the layout of a boxed region		Changes the flexible content to a horizontal layout. Icon also updates to "Change to Vertical Layout". This change is automatically committed.
Alter the layout of a boxed region		Changes the flexible content to a vertical layout. Icon also updates to "Change to Horizontal Layout". This change is automatically committed.
Reorder the children of a boxed region		<p>Launches the Reorder Contents page, page 5-17 where you can reorder the children of the current boxed region. Choose "Apply" from the Reorder Contents page to save your changes.</p> <p>Note: Use this control for reordering only the immediate children of this region. To reorder contents below that level, select the region's Personalize icon, page 4-7 to access the focused version of the Page Hierarchy Personalization page, which contains a Reorder icon for each applicable item within the region.</p>

Task	Launch Point from Layout Mode	Description
Remove a row or column		Removes a row or a column from the page. The Remove Column icon appears on the right, above each column. The Remove Row icon appears to the left of the each row. Both Remove icons direct you to a warning page that asks you to confirm whether you really want to remove the row or column.
Change the personalization context	Choose Context button	Launches the Choose Personalization Context page, page 5-4. Choose Apply from the Choose Personalization Context page to save your changes.
Activate, Inactivate, or Delete existing personalizations	Manage Levels button	Launches the Manage Personalization Levels page, page 5-36 where you can select existing personalizations to activate, inactive or delete.
Translate personalizations	Manage Levels button	Launches the Manage Personalization Levels page, page 5-36 where you can select the Translate icon for a specific personalization to translate it to another language in the Translate page, page 5-39. Choose Apply from the Translate page to save your changes.
Extract or Upload translations of a personalization	Manage Levels button	Launches the Manage Personalization Levels page, page 5-36 where you can choose the Upload Translation button, page 5-38 or Extract Translation File button, page 5-37 to upload or extract the translations for a specific personalization, respectively. Choose Apply from the Upload Translations page or Extract Translation Files page to save your changes.

Contents Mode

The Contents mode of the Page Layout Personalization page displays all content in the page, including the content attached to flexible layout regions.

The Contents Mode Task Flows section, page 4-10 below describes the personalizations you can accomplish in the Contents Mode. You can add new content to the layout, create new items in an existing boxed content region, as well as update the personalizable properties of specific elements in the page. You can also remove or reorder contents from different regions on the page. To rearrange contents across different regions, you must first remove them from their current location and then add them to the new destination region.

If you wish to exit from the Page Layout Personalization page, select the **Return to Application** link at the bottom of the page to return to the original page.

Figure 4-2: Contents Mode of the Page Layout Personalization page for the Update Customer page.

Personalize Page: Update Customer Choose Context Manage Levels

Personalization Level: Responsibility: OA Framework ToolBox Tutorial Labs

Layout Contents

Contents

You can update the properties of different elements on this page. You can also add, create, remove or reorder contents from different regions on the page. To rearrange contents across different regions, you must first remove them from their current location and then add them inside the new destination region.

UpdateCustomer Apply Cancel

Instruction Text Apply Cancel

* Name: John Smith

* Street: 200 Oracle Parkway

* City: Redwood Shores

State: California

* Zip Code: 95065

* Office phone: 650-506-6504

Home phone:

Cell phone: 650-506-6555

Fax: 650-506-63502

Customer Information

- [GL Analysis](#)
- [Profit Margin](#)
- [Contribution Margin](#)
- [Expenses](#)
- [Manpower Budget](#)
- [Analysis](#)
- [Manpower Comparison](#)
- [Manpower Ratio](#)
- [Yahoo](#)
- [E-Trade](#)
- [Oracle](#)

Flavourite

- [Schedule Meeting](#)
- [Upload Documents](#)
- [Meeting Notification](#)
- [Add Attendees](#)

Shortcuts

- [Schedule Meeting](#)
- [Upload Documents](#)
- [Meeting Notification](#)
- [Add Attendees](#)

Quotes

Name

[Rio Volt SP250](#)

[TEAC Mini CD/ MP3 Player](#)

[Superdisk 240 Drive](#)

Orders

Name

[eMachines Intel 1.1GHz](#)

[IBM ThinkPad T23](#)

[Triniton ViewSonic](#)

Customer Revenue

Year	Actual (Thousands)	Target (Thousands)
1998	400	250
1999	450	350
2000	600	450

Some text contents to explain the data shown in Customer Revenue Chart.

Cancel Apply

Layout Contents

[Return to Application](#) Choose Context Manage Levels

Copyright 2000-2005 Oracle Corporation. All rights reserved. [Diagnostics](#) | [Home](#) | [Logout](#) | [Preferences](#) | Personalize Page [About this Page](#) [Privacy Statement](#)

Contents Mode Task Flows

The following table lists the personalization tasks you can accomplish from the Contents Mode of the Page Layout Personalization page.

Task	Launch Point from Contents Mode	Flow Description
Personalize the properties (such as Prompt name, default value, and so on) of a specific page element		<ol style="list-style-type: none"> 1. Launches a focused version of the Page Hierarchy Personalization page, page 5-2 for the selected region. The Personalization Context shown at the top of the page lists the personalization context for the top-level object, which in the case of Oracle Applications, is the page layout. 2. Select the pen icon from the Personalize column to navigate to the Personalize page, page 5-6 for the page element. 3. In the Personalize page, you can edit the personalizable properties of the element at the personalization levels specified for the current context. 4. When you are done, select Apply to save your personalization or select Apply and Personalize Another to save your personalization and return to the focused Page Hierarchy Personalization page where you can select another element to personalize.
Add content to a boxed region		<p>Launches the Add Content page, page 5-19, allowing you to select predefined resource content from a catalog in the database. Select "Apply" to save your changes when you are done. The new content is added as the last child of the current layout.</p>
Create a new item in a boxed region		<p>Launches the Create Item page, page 5-20 where you can create a new item for the content region. Select "Apply" to save your changes when you are done. The new item appears as the last item in the region. Use the Reorder Content icon to reorder the layout sequence of the new item. You can only create new items in boxed content regions or in the top page layout region.</p> <p>Note: Use this control only for creating immediate children of a given region. To create items below that level, select the region's Personalize icon, page 4-10 to access the focused version of the Page Hierarchy Personalization page, which contains a Create icon for each applicable item within the region.</p>

Task	Launch Point from Contents Mode	Flow Description
Reorder the children of a boxed region		<p>Launches the Reorder Contents page, page 5-17 where you can reorder the children of the boxed region. Select "Apply" to save your changes when you are done.</p> <p>Note: Use this control for reordering only the immediate children of this region. To reorder contents below that level, select the region's Personalize icon, page 4-10 to access the focused version of the Page Hierarchy Personalization page, which contains a Reorder icon for each applicable item within the region.</p>
Remove a boxed region		<p>Removes the boxed content. Directs you to warning page that asks you to confirm whether you really want to remove the flexible content. Although a removed content no longer renders in the configurable page, page 4-4, you can add it back using the Add Content icon, page 4-10.</p>
Create, Duplicate, Update or Delete an "admin-seeded user-level" personalization		<ol style="list-style-type: none"> 1. Select the Personalize pen icon to launch a focused version of the Page Hierarchy Personalization page, page 5-2 for a selected boxed region. 2. Use the Hierarchy Page HGrid to identify the query region for which you wish to create an "admin-seeded user-level" personalization. (You can create seeded user-level personalizations only for a table or a HGrid in a query region.) 3. Select the Seeded User Views icon to launch the Personalize Views page, page 5-41 where you can create, update, duplicate or delete "Admin-seeded user level" personalizations. 4. When you create, duplicate or update a seeded view, you navigate to the respective Create/Duplicate/Update Views page, page 5-42. Select Apply to save your changes when you are done.
Change the personalization context	Choose Context button	<p>Launches the Choose Personalization Context page, page 5-4. Choose "Apply" from the Choose Personalization Context page to save your changes.</p>
Activate, Inactivate, or Delete existing personalizations	Manage Levels button	<p>Launches the Manage Personalization Levels page, page 5-36 where you can select existing personalizations to activate, inactive or delete.</p>

Task	Launch Point from Contents Mode	Flow Description
Translate personalizations	Manage Levels button	Launches the Manage Personalization Levels page, page 5-36 where you can select the Translate icon for a specific personalization to translate it to another language in the Translate page, page 5-39. Choose Apply from the Translate page to save your changes.
Extract or Upload translations of a personalization	Manage Levels button	Launches the Manage Personalization Levels page, page 5-36 where you can choose the Upload Translation button, page 5-38 or Extract Translation File button, page 5-37 to upload or extract the translations for a specific personalization, respectively. Choose Apply from the Upload Translations page or Extract Translation Files page to save your changes.

Using the Page Hierarchy Personalization Page

In the Page Hierarchy Personalization user interface, the entire layout of a configurable page is displayed in a hierarchy table (HGrid). The Page Hierarchy HGrid displays nodes for all the structures that make up the page. The initial focus of the Page Hierarchy is at the page layout level. As in all HGrids, you can change the focus of the Page Hierarchy HGrid, as well as drill in or out of the various page structure nodes. The HGrid location indicator keeps track of where you are within the page structure.

Note: If the page you personalize is a configurable page, page 4-4 (a page that contains metadata to personalize its layout) and you are not running your page in Accessibility mode (profile option **Self Service Accessibility Features** (ICX_ACCESSIBILITY_FEATURES) is set to None), you will not see the Page Hierarchy Personalization page, as shown below. Instead, you will see the Page Layout Personalization page, page 4-5, where you can perform the same personalization operations, using a more graphical user interface.

Figure 4-3: Page Hierarchy Personalization page for the Update Customer page in Accessibility mode

Personalization Context

Scope: **Page: Configurable Page**
 Document Name: **/oracle/apps/dem/employee/webui/ConfigurablePG**
 Site: **Include**
 Organization: **Vision Operations**
 Responsibility: **OA Framework ToolBox Tutorial**

Personalization Structure

Simple View Complete View

TIP In simple view mode, some items or regions may not be visible in the page structure and the reorder or create icon may be disabled for certain nodes. Please switch to complete view mode if you need full personalization capabilities over all items and regions.

[Expand All](#) | [Collapse All](#)

Focus Name	Shown	User Personalizable	Personalize	Reorder	Add Content	Remove Content	Create Item	Update Item	Delete Item	Seeded User Views	Customize Look And Feel	Show in Complete View
Page Layout: Configurable Page	Yes											
Flexible Layout: test flexible layout	Yes	Yes										
Flexible Content: (flexibleContent2)	Yes	Yes										
Query: (QueryRN)	Yes											
simpleSearchPanel												
Table: Employees Table	Yes	Yes										
simpleSearchMappings												
Flexible Layout: Flex Region 1	Yes	Yes										
corporateBranding												

Personalization Context

The **Personalization Context** shown at the top of the page lists the context for the top-level object, which in the case of Oracle Applications, is the page layout. It identifies the scope of the personalizations you are about to make, as well as to what personalization level(s) and value(s) these personalizations apply.

Personalization Structure

The **Simple View** and **Complete View** radio buttons in the Personalization Structure region allow you to toggle between a Simple or Complete view of the page structure in the HGrid below. As of OA Framework 11.5.10.2CU, when you select the global **Personalize Page** button, you automatically navigate to the Simple view of the page structure hierarchy. The Simple view does not show some of the layout elements that create unnecessarily complex nesting within the page structure hierarchy. This simplified structure allows you to quickly locate the region or item you wish to personalize.

If you initiate personalizations by selecting a **Personalize Region** link, you always navigate to a Complete view of the page hierarchy, with focus on the region from which you selected the **Personalize Region** link.

Both Simple and Complete views remember their status when you switch from one type of view to another. For example, if you focus on a particular page element node in the Complete view, then switch to the Simple View, and finally return to the Complete view, the focus will be on the same node from which you first left the Complete view. There are, however, exceptions:

- When you navigate to the **Choose Context** page, then select **Apply** to return to the Page Hierarchy Personalization page, the view is reset to Simple and the focus of the HGrid is also reset.
- When you navigate to the **Choose Context** page, then select **Cancel** to return to the Page Hierarchy Personalization page, you always return back to the same view and focus.
- When you navigate to the **Manage Levels** page, then select **Return to Personalization Structure** to return to the Page Hierarchy Personalization page, the view is reset to Simple and the focus of the HGrid is also reset.

When you display the Simple view of the Page Hierarchy Personalization page, a column called **Show in Complete View** is displayed in the HGrid. If you select the icon in this column for a page element node, the page switches to the Complete view, with the focus on that page element node.

If the Page Hierarchy Personalization page is in the Simple view, and the children of a node do not reflect the actual hierarchy structure of the children (because some layout elements are not shown), the **Reorder** and **Create** icons will be disabled. To enable the **Reorder** and **Create** icons for the children of that node, select the **Show in Complete View** icon for that node, so that the correct hierarchy of the children is shown in the Complete view.

Personalization Structure HGrid

The **Shown** column in the HGrid indicates whether the page element is rendered after applying all existing personalizations for the current context. The **User Personalizable** column indicates whether the page element is user personalizable.

The Page Hierarchy Task Flows section, page 4-14 describes how to accomplish your personalization needs using the various launch controls on this HGrid.

If you wish to exit from the Page Hierarchy Personalization page, select the **Return to <Page_Title>** link at the bottom of the page to return to the original page.

Page Hierarchy Task Flows

The following table lists the personalization tasks you can accomplish from the Page Hierarchy Personalization page. Each task starts from a single launch point on the page. Note that all tasks take you to different pages within the Personalization user interface. From these launched pages, you specify and save your personalization changes. No changes are ever committed on the Page Hierarchy Personalization page itself.

Task	Launch Point from Page Hierarchy Personalization page	Flow Description
Search the Page Hierarchy HGrid for specific page elements	Search	<ol style="list-style-type: none"> Expand the Search region, page 5-14. Specify criteria to search for specific nodes based on a particular style or a particular title, prompt, or text and display these nodes in a flat table. Separately or additionally, check Personalized Only, to filter the hierarchy to display already-personalized nodes in a flat table.
Personalize the properties (such as Prompt name, default value, and so on) of a specific page element		<ol style="list-style-type: none"> Locate the page element you wish to personalize in the Page Hierarchy HGrid. Select the pen icon from the Personalize column to navigate to the Personalize page, page 5-6 for the element. In the Personalize page, you can edit the personalizable properties of the element at the different personalization levels specified for the current context. When you are done, select "Apply" to save your personalization and return to the previous page. <p>Note: Although a region may be personalizable, not all region items in a region are personalizable. Each region item in a region has an ADMIN_CUSTOMIZABLE property associated with it. If the property is set to false by the developer of the region, the region item is not personalizable at the personalization level. For example, a developer would very likely set ADMIN_CUSTOMIZABLE to false for the vertical spacer item in a table layout region to prevent disruption of the spacing in a table.</p>
Reorder the children of a region		<p>Launches the Reorder Contents page, page 5-17 where you can reorder the children of the current region.</p>
Add content to a layout region		<p>Launches the Add Content page, page 5-19, allowing you to select predefined resource content from a catalog. The new content is added as the last child of the current layout region. You can add content to a layout region only if the layout has no other layout children.</p> <p>Note: Add Content is available only if the page you are personalizing is a configurable page, page 4-4.</p>

Task	Launch Point from Page Hierarchy Personalization page	Flow Description
Remove content from a layout region		<p>Directs you to warning page that asks you to confirm whether you really want to remove the current region. Although a removed region no longer renders in the page, you can add it back using the Add Content icon.</p> <p>Note: Remove Content is available only if the page you are personalizing is a configurable page.</p>
Create a new item in a region		<p>Launches the Create Item page, page 5-20 where you can create a new child item in a region. The new item appears as the last item in the region. Use the Reorder Content icon to reorder the layout sequence of the new item. You can only create new items in boxed content regions or in the top page layout region.</p> <p>Note: You can only create a new item if you have your personalization context set to include the Site level.</p>
Update an item created with the Create Item page		<p>Launches the Update Item page, page 5-34 where you can update an item for the region. Note that you may only update an item that was previously created with the Create Item page, page 5-20.</p>
Delete an item from a region		<p>Directs you to warning page that asks you to confirm whether you really want to delete the current item. Deleted items are non-recoverable.</p> <p>Note: Delete Item is available only if the page element is an item that was created at the Site level using the Create Item page, page 5-20.</p>
Create, Duplicate, Update or Delete an "admin-seeded user-level" personalization		<ol style="list-style-type: none"> 1. Use the Hierarchy Page HGrid to identify the query region for which you wish to create an "admin-seeded user-level" personalization. (You can create seeded user-level personalizations only a table or HGrid in a query region.) 2. Select the Seeded User Views icon to launch the Personalize Views page, page 5-41. 3. In the Personalize Views page, where you select different buttons or icons to create, duplicate or update a seeded view, you navigate to the respective Create/Duplicate/Update Views page, page 5-42.
Customize Look and Feel		<p>Launches the Customizing Look-and-Feel user interface, page 2-8.</p>

Task	Launch Point from Page Hierarchy Personalization page	Flow Description
Change the personalization context	Choose Context button	Launches the Choose Personalization Context page, page 5-4.
Activate, Inactivate or Delete existing personalizations	Manage Levels button	Launches the Manage Personalization Levels page, page 5-36, where you can select existing personalizations to activate, inactive or delete.
Translate personalizations	Manage Levels button	Launches the Manage Personalization Levels page, page 5-36, where you can select the Translate icon, page 5-39 for a specific personalization to translate it to another language.
Extract or Upload translations of a personalization	Manage Levels button	Launches the Manage Personalization Levels page, page 5-36, where you can choose the Upload Translation File button, page 5-38 or Extract Translation File button, page 5-37 to upload or extract the translations for a specific personalization, respectively.

Examples: Creating New Items Using the Personalization UI

You can declaratively add new items or fields to regions using the Personalization UI.

Since you can only add new items to an OA Framework application page at the Admin-level, you need to first enable Admin-Level personalizations, page 4-1 for your application by setting the **Personalize Self-Service Defn** (FND_CUSTOM_OA_DEFINITION) profile option to **Yes**. From the Functional Administrator responsibility, page 4-2, or in the page you wish to personalize, select the Personalize Page global link, page 4-4 to navigate to the Create Item page, page 5-20 in the Personalization UI.

Example: Adding a URL Link to the Advanced Worklist Page

The Personalization UI provides you with the ability to add new items as well as include other sources of information to a personalizable region. This example illustrates how to create a URL link in the Advanced Worklist page for all users of the site that displays additional information. This example creates the URL link within the page layout level

Important: Be sure that in the page you wish to link to, you also create a URL link back to the source page.

1. Enable Admin-Level personalizations for your application by setting the **Personalize Self-service Defn** (FND_CUSTOM_OA_DEFINITION) profile option to **Yes** at the user level for the user who has access to the Workflow User Web Applications responsibility.
2. Log in to the Personal Home Page as that user and select the **Workflow User Web Applications** responsibility. Navigate to the Advanced Worklist.
3. Select the **Personalize Page** global link.

4. In the Choose Personalization Context page, page 5-4, set the Scope to the current page and check **Include** for Site. Leave all other fields blank. Choose **Apply**.
5. In the Personalize page, page 5-6, select the **Create Item** icon for the **Page Layout: (WFNTFWORKLISTFNPAGE)** page element.
6. In the Create Item page, page 5-20 that appears, set the **Item Style** as *Static Styled Text*. Set the following properties for the new Static Styled Text item as indicated:
 - **ID** - NtfAddInfo
 - **Prompt** - Additional Information
7. If you want to navigate to some arbitrary URL from this region item, you should specify the Destination URI property as: *protocol>://<machine>:<port>/...*, such as `http://www.abcompany.com/home/page1.htm`.

If you want to call an Oracle Applications function, set the **Destination Function** property to the function name, for example to call the PL/SQL General Preferences page, you would enter `ICX_USER_PREFERENCES` for the Destination Function.

Note: If you wish to call a secured function, please refer to the *Oracle Applications System Administrator's Guide - Security* for additional information.

8. Choose **Apply** on the Create Item page when you are satisfied with the changes. The new item, **Static Styled Text: Additional Information** is created after the **Stack Layout: Notifications Worklist Function** page element within the Page Layout level. If you choose **Return to Application**, the new Additional Information link renders after all the content for the Notifications Worklist Function.
9. If you wish to move the Additional Information link so that it renders above the Notifications Worklist Function content, choose the **Personalize Page** link to return to the Personalization UI and navigate back to the Personalize page. Select the **Reorder** icon for **Page Layout: (WFNTFWORKLISTFNPAGE)**. In the Reorder Contents page, select **Additional Information** from the Site list and select the up arrow to move the Additional Information page element so that it appears before Notifications Worklist Function. Choose **Apply** on this page. Then choose **Return to Application** on the Personalize page to see the change.
10. When you select the new Additional Information link that appears in the Advanced Worklist, you navigate to the URL or function that you specified in the Create Item Page.

Note: If you add a new item to a region and it does not appear on the page, it is likely that the region you added the item to does not have the **Add Indexed Children** property in OA Extension set to **True** in by the developer. If this occurs and adding the new item to a different region within the region hierarchy is not an option, contact Oracle Support for assistance.

Tip: : If you make a mistake when you create a new item, such that it causes the base page to display an error, you can sign on to Oracle Applications as the Functional Administrator responsibility, page 4-2 and remove the personalization using the Manage Personalization Levels page, page 5-36.

Example: Adding New Instruction Text to a Page

You can use OA Personalization Framework to add additional instructional text for your users in the form of tips. This example illustrates how to add a new tip to a page at the Site level.

1. Enable Admin-Level personalizations for your application by setting the **Personalize Self-service Defn** (FND_CUSTOM_OA_DEFINITION) profile option to *Yes* at the user level.
2. Log in to the Personal Home Page as that user and navigate to the page you wish to personalize.
3. Select the **Personalize Page** global link.
4. In the Choose Personalization Context page, page 5-4, set the **Scope** to the current page and check **Include** for Site. Leave all other fields blank. Choose **Apply**.
5. In the Personalize page, page 5-6, select the **Create Item** icon for the region or page element where you want to add the tip.
6. In the Create Item page that appears, set the **Item Style** as *Tip*, and set the following properties:
 - **ID** - specify an ID, such as *HRPageTip*.
 - **Text** - set the text that you want to display on the page, next to the tip. For example: **HR Page Tip**.
 - **Tip Message Appl Short Name** - specify the application short name of the tip. For example, *HR*.
 - **Tip Message Name** - specify a message name. For example, *HR_BIS_PERIOD*.

Choose **Apply** when you are done. Then choose **Return to Application** on the Personalize page to see the change.

7. The new tip is added to the end of the region in which you created the new item. If you wish to move the tip so that it renders elsewhere within the region, choose the **Personalize Page** link to return to the Personalization UI and navigate to the Personalize page. Select the **Reorder** icon for the page element under which the tip resides. In the Reorder Contents page, select the tip from the Site list and select the up or down arrow to reorder its position within the page element. Choose **Apply** on this page. Then choose **Return to Application** on the Personalize page to see the change.

Note: If you add a new item to a region and it does not appear on the page, it is likely that the region you added the item to does not have the **Add Indexed Children** property in OA Extension set to **True** in by the developer. If this occurs and adding the new item to a different region within the region hierarchy is not an option, contact Oracle Support for assistance.

Tip: If you make a mistake when you create a new item, such that it causes the base page to display an error, you can sign on to Oracle Applications as the Functional Administrator responsibility, page 4-2 and remove the personalization using the Manage Personalization Levels page, page 5-36.

Function-Level Personalizations

A function in Oracle Applications is a piece of application logic or functionality that is registered under a unique name for the purpose of assigning it to, or excluding it from, a responsibility. You can create standard personalizations for a region at the Function level so that the personalizations are effective only for users of a specific function. Once you create a function-level personalization, you can update it or delete it.

Note: Oracle may deliver predefined Function-level personalizations. Customers can view but not update or delete "Oracle-seeded" Function-level personalizations.

Function-level personalizations are the highest level of personalizations you can make. Any further personalizations you make to the same region at lower Admin-levels always override the personalizations you make at the Function level.

To maintain function security, the function for which you are personalizing the region must be included in the responsibility from where users launch the page containing the region.

Once you create a function-level personalization, you can pass the function name corresponding to the personalized region by specifying the function name on the URL using the parameter OAFunc. For example:

```
http://<server.com>:<portID>/OA_HTML/OA.jsp?OAFunc=<custom_function>&...
```

OAFunc can be used in two different ways:

- If OAFunc is used without akRegionCode/akRegionApplId, then the function corresponding to it is launched.
- If OAFunc is used in addition to akRegionCode/akRegionApplId or page, then it is used for setting the function context for that page. A function context should be set for the function personalization to take effect.

Example

Suppose you have the following URL that launches an Oracle Applications page that is defined as the **web_html_call** of function XYZ:

```
OA.jsp?OAFunc = XYZ
```

XYZ points to:

```
OA.jsp?page=oracle.apps.fnd.framework.webui.TestRegion
```

If you want a function-level personalization of ABC (defined using OA Personalization Framework) to apply to this page, you should change the **web_html_call** of function XYZ to:

```
OA.jsp?page=oracle.apps.fnd.framework.webui.TestRegion&OAFunc=ABC
```

Please refer to the "Personalizable Pages" topic in Chapter 4 of the *Oracle Application Framework Developer's Guide* for details of other ways you can pass a function name to a corresponding personalized region.

Known Issues

See a summary of key Personalization issues, page B-1 with suggested workarounds if available.

Related Information

- BLAF UI Guideline(s)
None
- Javadoc Files
None
- Sample Code
None

Administrative-Level Personalization User Interface

This chapter covers the following topics:

- Overview
- Focused Page Hierarchy Personalization Page
- Choose Personalization Context Page
- Personalize Page
- Create Query Page
- Page Hierarchy Personalization Page: Expanded Search Region
- Update Layout Page
- Reorder Contents Page
- Add Content Page
- Create Content Page
- Create Item Page
- Update Item Page
- Manage Personalization Levels Page
- Extract Translation Files Page
- Upload Translations Page
- Translate Page
- Choose Languages Page
- Personalize Views Page
- Create View, Update View, and Duplicate View Pages
- Rename Columns / Totaling Page

Overview

This chapter provides detailed descriptions of all the individual pages in the user interface for Admin-level personalizations.

Focused Page Hierarchy Personalization Page

You may launch the "focused" version of the Page Hierarchy Personalization page, page 4-12 by selecting the Personalize (pen) icon for a boxed region from the Page Layout Personalization page, page 4-5. The Personalization Structure on the focused page initially focuses on the structure of the selected region, from which you launch this page.

Figure 5-1: "Focused" version of the Page Hierarchy Personalization page for the Update Customer page.

Personalization Context

Scope **Page: Framework ToolBox Tutorial: Lab Solution**
 Document Name **/oracle/apps/fnd/framework/toolbox/labsolutions/webui/ConfigHomePG**
 Responsibility **OA Framework ToolBox Tutorial Labs**

Personalization Structure

Simple View Complete View
 TIP In simple view mode, some items or regions may not be visible in the page structure and the reorder or create icon may be disabled for certain nodes. Please switch to complete view mode if you need full personalization capabilities over all items and regions.

[Expand All](#) | [Collapse All](#)

[Flexible Layout: Notifications Layout](#) > [Flexible Layout: \(flexibleLayout 0 1128634073828\)](#) >

Focus Name	Shown	User Personalizable	Personalize	Reorder	Create Item	Update Item	Delete Item	Seeded User Views	Customize Look And Feel	Show in Complete View
Flexible Content: Notifications Content	Yes	Yes								
Header: Notifications	Yes									

[Return to Personalize Page](#)

[Diagnostics](#) | [Home](#) | [Logout](#) | [Preferences](#) | [Help](#) | Personalize Page

Copyright (c) 2005, Oracle. All rights reserved. [Privacy Statement](#)

The **Personalization Context** region at the top of the page lists the context for the top-level object, which in the case of Oracle Applications, is the page layout. It identifies the scope of the personalizations you are about to make, as well as to what personalization level(s) and value(s) these personalizations apply.

The **Simple View** and **Complete View** radio buttons above the HGrid allow you to toggle between a Simple or Complete view, page 4-13 of the page hierarchy in the HGrid.

The **Shown** column indicates whether the page element is rendered after applying all existing personalizations for the current context. The **User Personalizable** column indicates whether the page element is user personalizable.

You can launch a few tasks from this page, as described in the following table.

Task	Launch Point from Focused Page Hierarchy Personalization page	Flow Description
Personalize the properties (such as Prompt name, default value, and so on) of a specific page element		<ol style="list-style-type: none"> 1. Use the Page Hierarchy HGrid to locate the page element you wish to personalize. 2. Select the pen icon from the Personalize column to navigate to the Personalize page, page 5-6 for the element. 3. In the Personalize page, you can edit the personalizable properties of the element at the different personalization levels specified for the current context. 4. When you are done, select Apply to save your personalization. <p>Note: Although a region may be personalizable, not all region items in a region are personalizable. Each region item in a region has an ADMIN_CUSTOMIZABLE property associated with it. If the property is set to false by the developer of the region, the region item is not personalizable at the personalization level. For example, a developer would very likely set ADMIN_CUSTOMIZABLE to false for the vertical spacer item in a table layout region to prevent disruption of the spacing in a table.</p>
Reorder the children of a boxed region		Launches the Reorder Contents page, page 5-17, where you can reorder the children of the boxed region. Select "Apply" to save your changes when you are done.
Create a new item in a boxed region		Launches the Create Item page, page 5-20, where you can create a new item for the content region. Select "Apply" to save your changes when you are done. The new item appears as the last item in the region. Use the Reorder Content icon to reorder the layout sequence of the new item. You can only create new items in boxed content regions or in the top page layout region.

Task	Launch Point from Focused Page Hierarchy Personalization page	Flow Description
Create, Duplicate, Update or Delete an "admin-seeded user-level" personalization		<ol style="list-style-type: none"> 1. Use the Hierarchy Page HGrid to identify the query region for which you wish to create an "admin-seeded user-level" personalization. (You can create seeded user-level personalizations only for a table or HGrid in a query region.) 2. Select the Seeded User Views icon to launch the Personalize Views page, page 5-41. 3. In the Personalize Views page, where you select different buttons or icons to create, duplicate or update a seeded view, you navigate to the respective Create/Duplicate/Update Views page, page 5-42.
Customize Look-and-Feel		Launches the Customizing Look-and-Feel user interface, page 2-8.
Update an item created with the Create Item page		Launches the Update Item page, page 5-34 where you can update an item for the region. Note that you may only update an item that was previously created with the Create Item page, page 5-20.
Delete an item from a region		<p>Directs you to warning page that asks you to confirm whether you really want to delete the current item. Deleted items are non-recoverable.</p> <p>Note: Delete Item is available only if the page element is an item that was created at the Site level using the Create Item page, page 5-20.</p>

If you wish to exit from the focused Page Hierarchy Personalization page, select the "**Return to Personalize Page**" link at the bottom of the page to return to the Page Layout Personalization page.

Choose Personalization Context Page

The Choose Personalization Context page is launched when you select the Choose Context button from any of the personalization launch pages.

Figure 5-2: Choose Personalization Context page, with the Scope set to Page, and the personalization level set to "Site".

1. In the Choose Personalization Context page, use the **Scope** poplist to specify whether you want to personalize the current page, or personalize a shared region, if any, that is extended by the current page. If there are no shared regions on the page, a single non-updateable radio button for the page itself is displayed for the scope.

When you select the Page as the scope, the personalization you create applies only to the current instance of the page (saved as a per instance personalization). If there are shared regions on the page and you select a shared region as the scope, the personalization you create on that shared region is also saved as a per instance personalization of that shared region but will appear in all occurrences of that shared region in any page that extends it. Refer to Per Instance versus Global Personalization, page 5-6 for further explanation.

Note: If you wish to personalize the Standard Footer that appears in OA Framework-based pages, you can use the Standard Footer Personalization Shortcut, page 4-2 that appears in the Application Catalog page under the Personalization tab of the Functional Administrator responsibility.

2. Specify the personalization level at which you want to make your changes and specify a value for that level. The Site level does not require any value as the personalization applies to all users of the current site. To create a Site level personalization, check the Include checkbox. To create personalizations at any other level, enter a value for that specific level. Responsibility or Function levels, for example, require a specific responsibility or function name, respectively.

You may specify more than one personalization level when you define a personalization. Simply specify values for each of the levels that you want this personalization to apply.

You can specify a personalization value in one of the following ways:

- Select a value from a specific level LOV (that is, from Function, Industry, Location, Organization, or Responsibility).
- Choose the **Set to My level** button. The **Set to My level** button resets the personalization value to the default value for that personalization level. The default value, in turn, is derived from your current apps context. For example, if you are logged in as the System Administrator responsibility, then choosing **Set to My Responsibility** would set the Responsibility level to `System Administrator`. For the Localization level, the default value is determined by the profile option ICX: Territory (ICX_TERRITORY). For the Organization level, the default value is determined by the profile MO: Operating Unit, which can be set at the Site or Responsibility level.

For more information about the ICX: Territory profile option, refer to the **Internationalization** section in the OA Framework Profile Options appendix of the *Oracle Application Framework Developer's Guide*.

3. Once you finish setting your personalization context, choose **Apply**. Choosing **Cancel** returns you to the previous page.

Per Instance versus Global Personalizations

Prior to OA Framework 11.5.10, when you personalized a region that extends a shared region (that is, it contains a region that is defined elsewhere and the current region simply references or extends that base region), the personalization was saved to the base region. As a result, if another page also extended that same base region, the personalization you saved would be seen on that page as well.

In 11.5.10 and on, the personalization you save on a region that extends a shared region, if you set the scope of your personalization context to the Page, is now saved on the current instance of that page, as a per instance personalization, for the context specified. As a result, if another page extends the same shared region, you will not see your personalization in that other page.

If you want to save the personalization of a shared region so that it applies to all pages or regions that extend that shared region, set the scope of your personalization context to the Shared region. Note however, that the personalization you save on that Shared region is also technically saved as a per instance personalization, even though the personalization applies to all pages or regions that extend this Shared region.

The reason is because a shared region can itself extend another shared region. Suppose you have Page A that extends Shared region B and Shared region B extends Shared region C. Say you set your personalization context scope to Shared region B and create a personalization that alters Shared region C. The personalization that you save, although it appears to be saved as a global personalization because it now applies to all pages that extend Shared region B, is actually saved as a per instance personalization of Shared region B. The reason is because the change that was made in Shared region C was only saved to the instance where it was extended by Shared region B. In other words, the personalization you saved on Shared region B was not saved to the base of Shared region C, but only to that instance of Shared region C as it appeared in Shared region B.

Personalize Page

Use the Personalize page to personalize the properties of a selected page element.

When you select the Personalize (pen) icon for a page element listed in the "complete" version, page 4-12 or "focused" version, page 5-2 of the Page Hierarchy Personalization page, the Personalize page appears. The Personalize page varies in appearance depending on the page element you are personalizing, for example, a non-table region versus a table region.

Figure 5-3: Personalize page for the "Priority" Message Styled Text item in the Notifications table of the Workflow Notifications page.

ORACLE [Diagnostics](#) [Home](#) [Logout](#) [Preferences](#) [Personalize Page](#)

Personalize Message Styled Text : Priority Cancel Apply

Personalization Context

Scope **Page: Notifications**
 Document Name **/oracle/apps/fnd/wf/worklist/webui/AdvancWorklistPG**
 Site **Include**

Personalization Properties

Clear Personalization Go Choose Levels Displayed

	Original Definition	Site	Result / Source
Access Key	Default	Inherit 5	Default / Original Definition
CSS Class	Default	Inherit 5	Default / Original Definition
Destination Function	Default	Inherit 5	Default / Original Definition
Destination URI	Default	Inherit 5	Default / Original Definition
Long Tip Region	Default	Inherit 5	Default / Original Definition
Prompt	Priority	Inherit 5	Priority / Original Definition
Rendered	false	Inherit 5	false / Original Definition
Required	no	Inherit 5	no / Original Definition
Search Allowed	false	Inherit 5	false / Original Definition
Selective Search Criteria	false	Inherit 5	false / Original Definition
Tip Message Name	Default	Inherit 5	Default / Original Definition
Tip Type	none	Inherit 5	none / Original Definition
Total Value	false	Inherit 5	false / Original Definition
User Personalization	true	Inherit 5	true / Original Definition

Clear Personalization Go Choose Levels Displayed

TIP The table rows show the personalizable properties for the object. Table columns show the personalization levels for each property. Any property set to inherit will assume its value from the next highest level.

Cancel Apply

[Diagnostics](#) | [Home](#) | [Logout](#) | [Preferences](#) | [Personalize Page](#)

Copyright 2000-2004 Oracle Corporation. All rights reserved. [Privacy Statement](#)
[About this Page](#)

Figure 5-4: Personalize page for a classic table region.

ORACLE Diagnostics Home Logout Preferences Personalize Page

Personalize Table : Notification Worklist Table Cancel Apply

Personalization Context

Scope **Page: Notifications**
 Document Name **/oracle/apps/fnd/wf/worklist/webui/AdvancWorklistPG**
 Site **Include**

Personalization Properties

Original Definition	Site	Result / Source
CSS Class Default	Inherit	5 Default / Original Definition
Controller Class oracle.apps.fnd.wf.worklist.webui.NtfWorklistCO	Inherit	5 oracle.apps.fnd.wf.worklist.webui.NtfWorklistCO / Original Definition
Message Appl Short Name FND	Inherit	5 FND / Original Definition
Message Name Default	Inherit	5 Default / Original Definition
Records Displayed 25	Inherit	5 25 / Original Definition
Rendered true	Inherit	5 true / Original Definition
Text Default	Inherit	5 Default / Original Definition
User Personalization false	Inherit	5 false / Original Definition
Width Default	Inherit	5 Default / Original Definition
Query		

Clear Personalization Choose Levels Displayed

TIP The table rows show the personalizable properties for the object. Table columns show the personalization levels for each property. Any property set to inherit will assume its value from the next highest level.

Cancel Apply

Diagnostics | Home | Logout | Preferences | Personalize Page

Copyright 2000-2004 Oracle Corporation. All rights reserved. Privacy Statement
[About this Page](#)

The personalization context is displayed at the top of the page. All the properties that you can personalize for the current page element are listed in the Personalization Properties table, page 5-8. This list of properties is dynamically determined by the page element you select. For example, the properties listed when you personalize a classic table region are different from the properties listed when you personalize a messageStyledText item. In addition, a special Query row, page 5-12 appears in the Personalization Properties table when you personalize a table region.

For information on the properties that you can personalize for a particular page element, refer to the Personalization Considerations appendix, page A-2. You may also refer to the *OA Component Reference* for a complete list and description of personalizable properties relevant for each region or item style.

Personalization Properties Table

Following is a description of the Personalization Properties table and how you can use the table to personalize your page element.

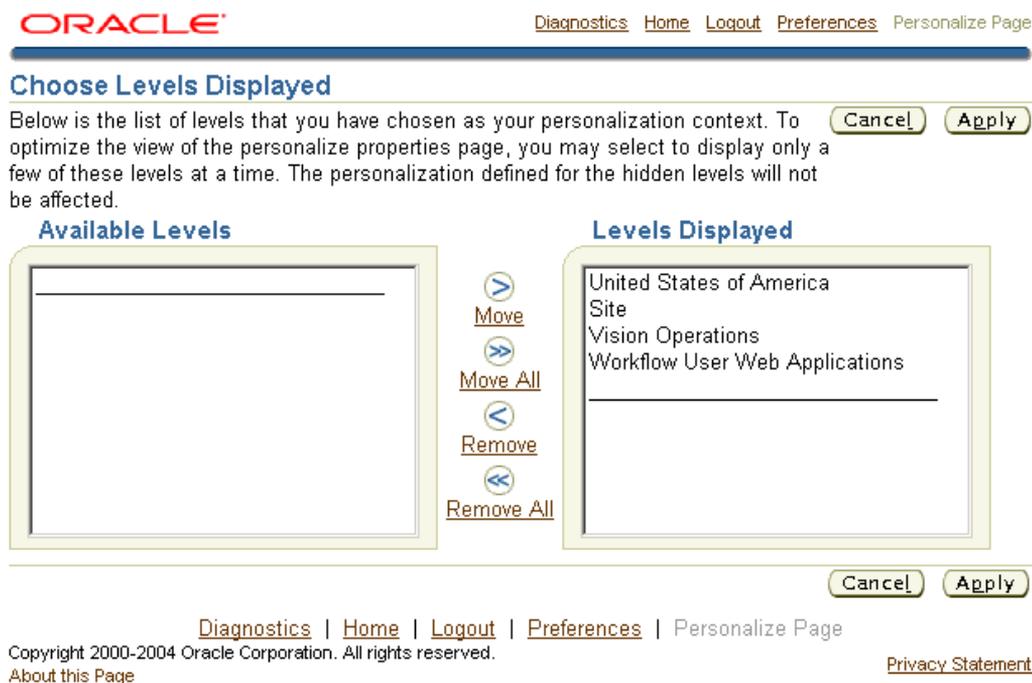
Global Table Actions

The Personalization Properties table provides the following controls for global table action, which you can select at any time:

- **Clear Personalization** - clears for a selected personalization level, all personalizations for all properties of the current page element. The poplist displays a list of the personalization levels for the current personalization context. Select a personalization level from the poplist and choose Go to perform this table action.

- **Choose Levels Displayed** - navigates to the Choose Levels Displayed page, where you can choose to hide or show specific personalization level columns in the Personalization Properties table using the shuttle region provided. Only the personalization levels for the current personalization context are displayed in this page. Use the arrows to move the desired personalization level between the Available Levels and Levels Displayed lists. Choose **Apply** to save your changes and return to the previous page. Note that personalization levels that are not displayed still affect the final cumulative results of your personalizations.

Figure 5-5: Choose Levels Displayed page



Original Definition column

The first column in the Personalization Properties table lists the name of each property that you can personalize for the current page element. The next column, **Original Definition**, lists the read-only base value of that property, as defined in the AK or current MDS repository, page 10-1.

The list of properties varies according to the page element you are personalizing. Refer to the *OA Component Reference* for a description of the personalizable properties for each region and item style.

Note: If you are personalizing an item that has a CSS Class property, you can use the CSS Styles Lookup page, page 2-31 to first preview your item with different CSS styles before setting the CSS Class name for the item. See the section on Style Sheets, page 2-23 for additional information.

Step 1: Update the personalization level columns

Following the Original Definition column, are columns for each of the personalization levels, shown in order of lowest to highest precedence, as defined by your personalization

context. These columns are collectively referred to as the **personalization level columns**. For example, if you select the Site, Organization, and Responsibility levels as your personalization context in the Choose Personalization Context page, page 5-4, the table displays Site, Organization: <Organization Value>, and Responsibility: <Responsibility Value> personalization level columns, so that you can personalize the page element properties at each of these personalization levels.

The fields in these personalization level columns are updateable, so you can enter a personalized value for each property at each personalization level.

The fields in the personalization level columns are used as follows:

- If a field is not personalized, it displays a special value of `Inherit`, by default. This special value indicates that at that personalization level, the property inherits its value from the personalization level above it (the preceding column).
- You can update a value displayed in a field and if partial page rendering is enabled, the field refreshes to display the new result value. If partial page rendering is disabled, you must select the Go button, page 4-1 that renders to refresh the field with this new value.

Note: The personalized value at that personalization level is not actually applied to the page element itself until you choose the **Apply** button on the Personalize page.

- If a field contains a personalized value, a **Clear Personalization** icon is enabled next to the updateable field. Selecting this icon sets the value back to `Inherit`, so that the property inherits its value from the level above it.
- Note that a blank value is not equivalent to a value of `Inherit`.

Personalizing SPEL-supporting Properties

Release 11.5.10.2CU supports SPEL (Simplest Possible Expression Language) personalizations on the following properties:

- **Read Only**
- **Rendered**
- **Required**

When you select the poplist icon for any of these properties in the personalization level columns, the poplist generally displays the choices `Inherit`, `true`, `false`, and `SPEL`, but with the following caveats:

- If the base value of **Rendered** is `SPEL`, the poplist choices for **Rendered** are `false` or `SPEL`.
- If the base value of **Required** is `SPEL`, the poplist choices for **Required** are `true` or `SPEL`.
- If the base value of **Read Only** is `SPEL`, the poplist choices for **Read Only** are `true` or `SPEL`.

These caveats maintain the integrity of the business logic and security criteria of the page.

For more information about SPEL personalizations, refer to the **Implementing Partial Page Rendering Events > Changing UI Properties** section in the Partial Page Rendering (Dynamic User Interface) chapter of the *Oracle Application Framework Developer's Guide*.

When you select `SPEL` from the poplist, an additional text box renders for you to enter a SPEL expression (string), such as `#{oa.<viewObject>.<viewAttr>}` or `#{oa.function.<functionName>}`.

If the value for any of the above properties is already a SPEL expression, the property displays the value `SPEL` in its poplist and a text box shows the SPEL expression.

If you select any value other than `SPEL` in the poplist, the text box is not rendered.

When you personalize the values for any of these SPEL-supporting properties, you may see warnings.

- If the base value of **Rendered** is `true` or `SPEL`, and you personalize the value to `false` or `SPEL` with a different SPEL expression, the following warning appears:
"Setting *<property name>* to *<property value>* may hide this item and its children. The application may be inoperable if this item or any of its children is required."
- If you personalize **Required** to a SPEL expression other than its original SPEL expression, the following warning appears:
"Setting *<property name>* to *<property value>* may affect the components that depend on the original SPEL expression or value of this property."
- If you personalize **Read Only** to `true` or `SPEL`, the following warning appears:
"Setting *<property name>* to *<property value>* may make this item read-only. A read-only item is not validated on the browser and is not included in the form submitted to the server."

Setting Default Values

You may set a default value for a page element in the Personalization Properties table only if the page element is of one of the following region item styles:

- `MessageTextInput` (for Datatypes - `VarChar`, `Date`, `Number`)
- `MessageLOVInput`
- `MessageCheckBox`
- `MessageChoice`
- `MessageRadioButton`
- `MessageRadioGroup`

The above item styles all have a property called `Initial Value` that you may personalize. The value you specify for this property becomes the default value that an end-user sees in the page element when the page initially displays. When you set the `Initial Value` property, be sure the value matches the data type and field-level restrictions of that item page element, otherwise you get an exception. For example, you cannot specify a value of `"True"` on a date field or set the value to a length of 20 characters if the field length is 10 characters.

The default value is effective for both non-view object-mapped and view object-mapped page elements. If a page element is not mapped to a view object, the default value is applied immediately and reflected on the page. If a page element is mapped to a view object, the default value is set on the row and will appear when a user creates a new row. The default value also overrides any default value set in the page element's underlying business object.

In cases where the page element exists beneath a container page element in the Page Hierarchy, the following applies:

- If the item page element is contained beneath a **table** or **Table-In-Table** page element, the Initial Value property you set for the item page element becomes effective immediately.
- If the item page element is beneath an **advanced table** container page element, the Initial Value property you set for the item page element will not be effective until a future release.
- The Initial Value property is not supported for an item page element contained beneath an **HGrid** page element.

Note: If a developer sets a default value on an item page element in their controller code, the developer-set default will override the default value you personalize in the Initial Value property.

Step 2: Verify the Value Displayed in the Result/Source column

The last column in the Personalization Properties table is **Result/Source**. Although you cannot update a value in this column directly, a value in this column is refreshed whenever you update a value for a property at a specific **personalization level**. This column displays the resulting value of a property if all the personalized values for the current context of personalization levels are applied, as well as the source of that resulting value, which can either be from the personalization level that takes the highest precedence or the Original Definition, if the property has not been updated.

Verify that the value shown in this column is what you want to accomplish when you apply all the personalizations in the current context to the property. Otherwise, update the property value again for the appropriate personalization level, until you get the desired result.

Step 3: Personalize the Data Displayed in a Table Region using the Query Row

If the page element you personalize is a table region, a special row called **Query** appears at the end of the Personalization Properties table. The Query row displays a Query icon in each personalization level column. When you select the Query icon, you navigate to the Create Query page, page 5-13. You can specify sort information and search criteria on this page to sort and filter the data that displays in the rendered table region when your personalizations are applied. After you create a query and return to the Personalization page, the Query icon in the Query row for that personalization level, changes to a **Personalize** icon (a pen) to indicate that you can view or author personalize that existing query.

Step 4: Apply Personalizations

Once you finish making changes to the Personalize page, you can choose one of the following buttons:

- **Apply** - to save your personalizations and return to the Page Layout Personalization page.
- **Apply and Personalize Another** - to save your personalizations and return to the "focused" Page Hierarchy Personalization page.
- **Cancel** - to cancel any changes and return to the previous page.

Note: If you navigate to the Personalize page by launching a flow from the "complete" Page Hierarchy Personalization page, page 4-12, the **Apply and Personalize Another** button does not render. The **Apply** button also has slightly different behavior in this case; it saves your personalizations and returns to the Page Hierarchy Personalization page.

Create Query Page

You can navigate to the Create Query page by selecting the Query row icon from the Personalization Properties table in the Personalize page, page 5-6 when you personalize a table region. This page allows you to specify sort information and search criteria for a table region to filter the data that displays in that table.

Figure 5-6: Create Query page for a table region.

ORACLE [Diagnostics](#) [Home](#) [Logout](#) [Preferences](#) [Personalize Page](#)

Create Query

[Cancel](#) [Apply](#)

Data Filter

Show table data when all conditions are met. [Clear Filter](#)

Show table data when any condition is met.

Subject

From

Message Attribute

Sent

Add Another [Add](#)

Sorting

[Clear Sorting](#)

	Original Definition (Column Name/Sort Direction)	Column Name	Sort Direction
First Sort	Sent/descending	<input type="text"/>	<input type="text"/>
Second Sort	Subject/ascending	<input type="text"/>	<input type="text"/>
Third Sort	Default/Default	<input type="text"/>	<input type="text"/>

[Cancel](#) [Apply](#)

[Diagnostics](#) | [Home](#) | [Logout](#) | [Preferences](#) | [Personalize Page](#)

Copyright 2000-2004 Oracle Corporation. All rights reserved. [Privacy Statement](#)

[About this Page](#)

1. In the Data Filter region, indicate how you want the filter to match your search conditions by selecting one of the following radio buttons: **Show table data when all conditions are met** or **Show table data when any condition is met**.

2. first four columns of the table are listed for you to specify search criteria. Using the poplist following the column name, choose a search condition and enter a value to search for in that column.
3. You can also select a column from the **Add Another** poplist and choose **Add** to add more search criteria to your filter.

Note: Only columns whose Search Allowed property is set to True are displayed in the Add Another poplist. If this property is initially set to false and you want to be able to search on this column, you must first create an admin-level personalization of the table region that updates the Search Allowed property for the column (item) to True.

If you leave the search criteria blank for a column, the filter does not search on that column.

4. In the Sorting region, you can specify up to three levels of sorting for your data. Select a column from the **Column Name** poplist for each level of sorting you wish to perform.

Note: Only columns whose Sort Allowed property is set to Yes are displayed in the Column Name poplist. If this property is initially set to no and you want to be able to sort on this column, you must first create an admin-level personalization of the table region that updates the Sort Allowed property for the column (item) to yes.

5. For each sort column, you must specify whether to sort in ascending or descending direction.
6. When you are done creating the data filter and specifying your sort settings, choose **Apply** to apply your changes and return to the Personalize page. The Query row of the Personalization Properties table should now display a Personalize icon (pen). The change in icon indicates that a data filter now exists at that personalization level and you can select the icon to view or update the existing data filter again.

Note: When you personalize a table region at various personalization levels and for each of those levels, specify search criteria for different columns in the table region, the search criteria are combined with a logical AND for the cumulative personalizations. The same is true if you personalize a table region at various personalizations levels, with search criteria for the same columns in the table region. The search criteria defined in all levels are AND-ed together.

Page Hierarchy Personalization Page: Expanded Search Region

When you choose the expand icon (+) for Search in the Page Hierarchy Personalize page, the Search region expands as shown in Figure 5-7. You can use this region to specify criteria that searches for a specific subset of page elements.

Note: You can perform a search only on pages, regions and their personalizations that reside in the database repository. Search is not able to locate elements on pages that reside on the local file system.

1. Specify in the **Style** field whether to search for elements of a specific item style. If Style is set to null, then elements of all styles are searched. Note that the Style field displays only the styles used on the page elements of the current page.
2. Specify in the **Title/Prompt/Text** field whether to search for elements that contain a specific title, prompt, or text. You may specify a wild card character (%) in the search string.
3. Check **Include Personalized Items Only** if you want the search to return only matching results that have been personalized.

Figure 5-7: Search region for the Page Hierarchy Personalize page.

▼ Search

Style

Title/Prompt/Text

Include Personalized Items Only

4. Select **Go** to perform the search.

The results are returned in a flat table, where the names of the matching page elements are sorted by style. This flat table has the same columns as the prior Page Hierarchy HGrid, so that you can launch the same personalization tasks from this presentation. However, it also includes a column called **View in Hierarchy**, from which you can select the Hierarchy icon for a page element to focus on the page element as displayed within the page hierarchy.

Figure 5-8: Search Results in the Page Hierarchy Personalization page, showing all personalized page objects.

ORACLE Diagnostics Home Logout Preferences Personalize Page

Personalize Region: Customizable and searchable worklist Choose Context Manage Levels

Personalization Context

Scope **Region: Customizable and searchable worklist**
 Document Name **/oracle/apps/fnd/wf/worklist/webui/FullWorklistPG**
 Site **Include**

▼ Search

Style
 Title/Prompt/Text
 Include Personalized Items Only

Name	Parent	Shown	User Personalizable	Personalize	Reorder	Create Item	Update Item	Delete Item	Seeded User Views	View in Hierarchy
Form Value: FYI	Table		Yes							
Form Value: Message Attribute	Table		Yes							
Form Value: Response Required	Table		Yes							
Message Styled Text: Closed	Table		Yes							
Message Styled Text: Due	Table	Yes	Yes							
Message Styled Text: From Me	Table		Yes							
Message Styled Text: From	Table	Yes	Yes							
Message Styled Text: Information Requested From	Table		Yes							
Message Styled Text: Notification ID	Table		Yes							
Message Styled Text: Priority	Table		Yes							
Message Styled Text: Sent	Table	Yes	Yes							
Message Styled Text: Status	Table	Yes	Yes							
Message Styled Text: Subject	Table	Yes	Yes							
Message Styled Text: To	Table		Yes							
Message Styled Text: Type	Table	Yes	Yes							
Multiple Selection: Select notifications	Table	Yes								
Selection Button: Close	Multiple Selection									
Selection Button: Open	Multiple Selection	Yes	Yes							
Selection Button: Reassign	Multiple Selection	Yes	Yes							
Table: Customizable and searchable worklist		Yes	Yes							

[Return to Personalization Structure](#)

[Return to Application](#) Choose Context Manage Levels

Diagnostics | Home | Logout | Preferences | Personalize Page

Copyright 2000-2004 Oracle Corporation. All rights reserved. [Privacy Statement](#)
[About this Page](#)

5. Select the **Return to Personalization Structure** link below the Search Results table if you want to return to the original Page Hierarchy Personalization page regardless of the focus you had before you selected Go to search for specific page elements.

Update Layout Page

The Update Layout page lets you alter the layout of a boxed region. You may lay out the content horizontally or vertically. Alternatively, you may split the boxed region into multiple peer layout regions. If you split the boxed region, you must select the Add Content icon, page 4-10 from the Page Layout Personalization page to add content to the new peer region.

This page is launched from the Update Layout icon on the Page Layout Personalization page. You can only alter the layout of a page if the page you are personalizing is a configurable page, page 4-4.

Figure 5-9: Update Layout page

ORACLE [Diagnostics](#) [Home](#) [Logout](#) [Preferences](#) [Personalize Page](#)

Update Layout

[Cancel](#) [Apply](#)

You can change the direction in which content is placed. Alternatively, you can split the current region in rows/columns further.

Place contents vertically
 Place contents horizontally
 Split Region

No of Rows
No of Columns

[Cancel](#) [Apply](#)

[Diagnostics](#) | [Home](#) | [Logout](#) | [Preferences](#) | [Personalize Page](#)
Copyright 2000-2004 Oracle Corporation. All rights reserved.
[About this Page](#) [Privacy Statement](#)

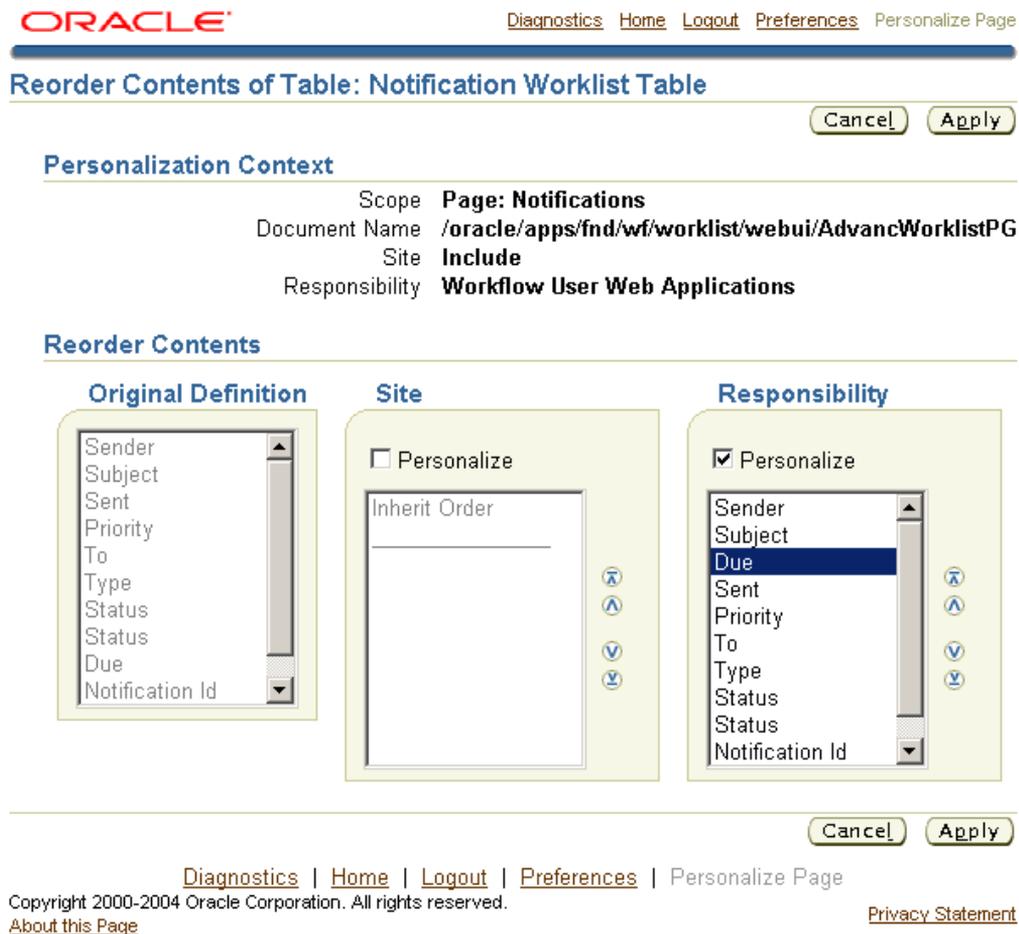
1. Choose to layout the content vertically or horizontally. Alternatively, you may choose to split the current region into multiple peer regions.
2. If you select Split Region, specify the number of row and columns to split the region.
3. Choose **Apply** when you are done. You may also choose to **Cancel** your changes and return to the previous page.

Note: If you split the boxed region, you must select the Add Content icon, page 4-10 from the Page Layout Personalization page to add content to the new peer region.

Reorder Contents Page

When you choose the Reorder Contents icon from the Page Layout Personalization page, page 4-5, the complete Page Hierarchy Personalization page, page 4-12, or the focused Page Hierarchy Personalization page, page 5-2, you navigate to the Reorder Contents page. You can reorder the contents of a page layout only if the page is configurable, page 4-4, otherwise you can reorder the contents of a region.

Figure 5-10: Reorder Contents page



The Reorder Contents page displays a reorder region for each personalization level set in the current personalization context and also displays a read-only region that shows the order of the content in the base definition.

1. Use a reorder region to reorder the contents of a region by personalization level. The usage of a reorder region is as follows:
 - If the contents of a region have not been reordered at a given personalization level, the reorder region displays **Inherit Order**, by default. This indicates that at that personalization level, the boxed region inherits its order from the personalization level above it (lower levels take precedence, page 3-5).
 - If you want to reorder the contents for a given personalization level, check the **Personalize** checkbox for that personalization level. If partial page rendering is enabled, the region refreshes to display the content that can be reordered. If partial page rendering is disabled, you must select the Go button, page 4-1 to refresh the region so it displays the content to reorder.

Note: The personalized order for any given personalization level is not actually applied until you choose **Apply** on the page.

- The **Personalize** checkbox may also be checked for a personalization level if the region has previously been reordered (personalized).
 - Use the arrows on the side of the reorder region to reorder its contents.
2. Choose **Apply** to apply the new order for each personalization-level that was personalized and return to your previous page.

Note: For configurable pages, you can reorder content only within the content's assigned layout region. You cannot reorder content across different layout regions.

Add Content Page

When you choose the Add Content icon from the Page Layout Personalization page, page 4-5 or the complete Page Hierarchy Personalization page, page 4-12, you navigate to the Add Content page. You can add content to a page only if the page is configurable, page 4-4 and if the region you are adding content to is a boxed layout region that contains only content regions and not other layout regions.

Figure 5-11: Add Content page

ORACLE [Diagnostics](#) [Home](#) [Logout](#) [Preferences](#) [Help](#) [Personalize Page](#)

Add Content
Choose from the following predefined contents to add to the selected region.

Search

Title
Description

[Select All](#) | [Select None](#)

Select Title	Description	View Name
<input type="checkbox"/> Related Info		

[Diagnostics](#) | [Home](#) | [Logout](#) | [Preferences](#) | [Help](#) | [Personalize Page](#)

Copyright (c) 2005, Oracle. All rights reserved. [Privacy Statement](#)
[About this Page](#)

The Add Content page displays a catalog of predefined content regions that you can add to the selected region. The Content catalog lists the title of the predefined content and its description.

Note: This page initially displays only the predefined content regions that were delivered with your Oracle application. If you have defined additional independent content regions that you want to include on this list, select the **Create Content** button to access the Create Content page, page 5-20.

If a content contains a user-personalizable region and one or more personalized views of that region exists, the Content catalog will list each defined view by its view name and the content with which it is associated. Once you add a content (or a personalized view of a content) to a boxed layout region, the next time you display the Add Content page, the content or other personalized views of the content will no longer be available

for selection from the Content catalog. You must remove the content from the current layout region, using the Remove Boxed Region icon, page 4-8 , before you can choose a different personalized view of that content to add.

1. Enter a title or description in the Search region and choose **Go** to search for the content you wish to add.
2. Check **Select** for the content you wish to add.
3. Choose **Apply** to save your personalization and return to your previous page.

The new content is added as the last content region of the selected boxed layout region.

Create Content Page

When you click the Create Content button on the Add Content page, page 5-19, you navigate to the Create Content page.

Create Content page

ORACLE [Diagnostics](#) [Home](#) [Logout](#) [Preferences](#) [Personalize Page](#)

Create Content

Please enter the full name of your component; for example, /com/mine/ext/MyTableRN.

Level **Site**

* ID

* Title

* Full Name

Description

TIP Please verify that your component can be displayed by itself without depending on any resources outside the component.

[Diagnostics](#) | [Home](#) | [Logout](#) | [Preferences](#) | [Personalize Page](#)

Copyright (c) 2005, Oracle. All rights reserved. [About this Page](#) [Privacy Statement](#)

Use the Create Content page to select a custom content region that you have already defined in JDeveloper, and want to include on the list of predefined content regions on the Add Content page. Be sure to enter the region's ID and title, and the full name (for example, /com/mine/ext/MyTableRN). It is a good idea to add a description, which will provide useful selection information. When you select **Apply**, the Add Content page, page 5-19 will appear, with this region added to the list of available content regions.

Important: The region you specify must be displayable without the use of any other resources.

Create Item Page

When you choose the Create Item icon from the Page Layout Personalization page, page 4-5, the complete Page Hierarchy Personalization page, page 4-12, or the focused Page Hierarchy Personalization page, page 5-2, you navigate to the Create Item page. Use the Create Item page to declaratively add a new item to a region using the OA Personalization Framework.

Important: You can add a new item to a region using OA Personalization Framework only if:

- The Add Indexed Children property in Oracle 9i JDeveloper OA Extension has been set to True for that region.
- You are personalizing a region at the **Site, Function, or Localization** level.

Any new item that you create while personalizing a region at the Function or Localization level may only be further personalized at the same Function or Localization level at which it was created. New items that you create while personalizing a region at the Site level may later be personalized at any level. For example, you may personalize at the Responsibility level, an item that you create at the Site level.

Important: You may add a new item to a table or HGrid; however, all items in the table or HGrid must be set to the same BC4J view usage name.

Figure 5-12: Create Item page listing the properties you can specify for a Tip item.

ORACLE [Diagnostics](#) [Home](#) [Logout](#) [Preferences](#) [Personalize Page](#)

Create Item

* Indicates required field

Level

Item Style

Property	Value
* ID	<input type="text"/>
Admin Personalization	<input type="text" value="true"/> <input type="button" value="v"/>
Attribute Set	<input type="text"/>
CSS Class	<input type="text"/>
Comments	<input type="text"/>
Extends	<input type="text"/>
Rendered	<input type="text" value="true"/> <input type="button" value="v"/>
Scope	<input type="text" value="."/>
Text	<input type="text"/>
Tip Message Name	<input type="text"/>

[Diagnostics](#) | [Home](#) | [Logout](#) | [Preferences](#) | [Personalize Page](#)

Copyright 2000-2004 Oracle Corporation. All rights reserved. [Privacy Statement](#)
[About this Page](#)

1. Use the **Level** poplist to specify whether to create the new item at the Site or Function level. Use the **Item Style** poplist to select the item style to create. Currently, you can create an item of any of the following styles:

Item Style	Description	Usage Notes
Attachment Image	<p>An Attachment column in a table region that renders the following icons:</p> <ul style="list-style-type: none"> • Attachment icon - navigates to the Attachment page, where users may view the attachments for a current record. • Add icon - navigates to the Add Attachment page, where users can add an attachment to a record <p>Note: This item style is only available for creation under a table or advanced table region.</p>	<ol style="list-style-type: none"> 1. You must enter the name of the View Instance for the entity you are attachment-enabling. 2. After creating Attachment Table item, select the Create Item icon again for categoryMappings in the Page Hierarchy Personalization page to create a Category Map, page 5-24. 3. Select the Create Item icon for entityMappings under the Attachment Table item in the Page Hierarchy Personalization page to create an Entity Map, page 5-25. 4. Finally, select the Create Item icon for primaryKeys under the Attachment Table item in the Page Hierarchy Personalization page to create a Primary Key, page 5-31 if none has yet been defined for the entity map.

Refer to the *OA Component Reference* for information about other properties to set.

Item Style	Description	Usage Notes
Attachment Link	A View link that navigates to the Attachments page where users can view existing attachments for the entity record. This item style also renders an Add button that users can select to add new attachments to the current entity record.	<ol style="list-style-type: none"> 1. You must enter the name of the View Instance for the entity you are attachment-enabling. 2. After creating Attachment Link item, select the Create Item icon again for categoryMappings in the Page Hierarchy Personalization page to create a Category Map, page 5-24. 3. Select the Create Item icon for entityMappings under the Attachment Link item in the Page Hierarchy Personalization page to create an Entity Map, page 5-25. 4. Finally, select the Create Item icon for primaryKeys under the Attachment Link item in the Page Hierarchy Personalization page to create a Primary Key, page 5-31 if none has yet been defined for the entity map.

Refer to the *OA Component Reference* for information about other properties to set.

Item Style	Description	Usage Notes
Attachment Table	An Attachments table that users can use to view existing attachments for the entity record.	<ol style="list-style-type: none"> 1. You must enter the name of the View Instance for the entity you are attachment-enabling. 2. After creating Attachment Table item, select the Create Item icon again for categoryMappings in the Page Hierarchy Personalization page to create a Category Map, page 5-24. 3. Select the Create Item icon for entityMappings under the Attachment Table item in the Page Hierarchy Personalization page to create an Entity Map, page 5-25. 4. Finally, select the Create Item icon for primaryKeys under the Attachment Table item in the Page Hierarchy Personalization page to create a Primary Key, page 5-31 if none has yet been defined for the entity map. <p>Refer to the <i>OA Component Reference</i> for information about other properties to set.</p>
Button	A button that represents an action that you can perform on a region. OA Framework generates buttons as images that represent the underlying function.	<p>You must enter a value for Prompt with this item style.</p> <p>If you want your Button to navigate to another page, you can specify a Destination URI value that navigates to that page.</p> <p>Refer to the <i>OA Component Reference</i> for information about other properties to set.</p>
Category Map	<p>A custom category to associate with an Attachment item.</p> <p>Note: This item style appears in the Item Style poplist only when you choose the Create Item icon for categoryMappings under an Attachment entity hierarchy in the Page Hierarchy Personalization page.</p>	<p>You must enter a value for the Category property. This value should be identical to the predefined internal name of the category in the FND_DOCUMENT_CATEGORIES table.</p> <p>You may specify whether to secure the categoryMap with Application Security rules. The default value for the Secure property is False.</p> <p>Refer to the <i>OA Component Reference</i> for information about other properties to set.</p>

Item Style	Description	Usage Notes
Column	<p>A new column in an advanced table or a column group.</p> <p>Note: This item style appears in the Item Style poplist only when you choose the Create Item icon for an <i>advancedTable</i> or a <i>columnGroup</i> in the Page Hierarchy Personalization page.</p>	<p>Every new column is accompanied by a Create Item icon, as well as a columnHeader child item with its own Create Item icon, on the HGrid of the Page Hierarchy Personalization page, page 4-12.</p> <p>The only style of item you can insert under the columnHeader is a sortableHeader.</p> <p>Caution: If you create a sortableHeader item under the columnHeader, you must also create at least one more child item for the column (by clicking the column's Create Item icon); otherwise a run-time exception will occur when you choose Return to Application. The Prompt property of the sortableHeader will provide the column's "header," and the immediate children of the column will constitute its "body."</p>
Entity Map	<p>An entity to associate with an Attachment item.</p> <p>Note: This item style appears in the Item Style poplist only when you choose the Create Item icon for entityMappings under an Attachment entity hierarchy in the Page Hierarchy Personalization page.</p>	<p>You must enter a value for the Entity property:</p> <ul style="list-style-type: none"> • If you defined an entity in the AK Entity form to share between the Oracle Forms and OA Framework interfaces, specify the Entity ID you defined in the AK Entity form. • If you are enabling attachments only to be viewed from the OA Framework interface, enter a unique arbitrary value for Entity and be sure to create a primaryKeyMap for the entity map. <p>Refer to the <i>OA Component Reference</i> for information about other properties to set.</p>

Item Style	Description	Usage Notes
Export Button	An Export button that when selected, exports the data displayed in a region.	<p>Create this item at the desired position within the region whose data you want to export.</p> <p>You must set the View Instance name to the same view object associated with the region for which you want to enable export.</p>
Flex	A key or descriptive flexfield. A key flexfield represents an object defined by the values a user enters in each of its segments. A descriptive flexfield is customizable "expansion space" on a page.	<p>You must specify values for the following properties of this item style:</p> <ul style="list-style-type: none"> • Appl Short Name • Name - set to the name of the flexfield as it was registered • View Instance • Type - key or descriptive <p>If you want only specific segments of the flexfield to render, specify a value for the Segment List property. If this property is left blank, then all segments are rendered. Segments within a certain context are listed, separated by " ", a single pipe, while data from different context is separated by " ", a double pipe.</p> <p>Refer to the <i>OA Component Reference</i> for information about other properties to set.</p>

Item Style	Description	Usage Notes
Flexible Layout	<p>A flexible layout region.</p> <p>Important: This item type can only be created at the Site level and is intended to be used for inserting another region into a page by adding one flexible content that extends that region. It does not allow the construction of complex flexible layout regions. To construct a fully configurable page, use OA Extension, as described in the <i>Oracle Application Framework Developer's Guide</i>.</p> <p>Note: If a page contains flexible layout and flexible content created from the personalization Create Item page, you get the Page Hierarchy Personalization page UI, page 4-12 when you personalize the page. If a page's base definition contains flexible layouts and flexible contents, you get the Page Layout Personalization page UI, page 4-5.</p>	<ol style="list-style-type: none"> Specify whether the Rendered property is true or false. Specify an optional Comment. Select the Create Item icon for the newly created flexible layout region in the Page Hierarchy Personalization page to create a flexible content element, page 5-27.
Flexible Content	<p>Flexible content that extends another region and renders within a flexible layout region.</p> <p>Note: This item style appears in the Item Style poplist only when you choose the Create Item icon in the Page Hierarchy Personalization page for a newly created Flexible Layout region defined in the Create Item page.</p>	<p>Specify the full package name of the region that the flexible content extends.</p> <p>Important: Flexible content can only be inserted into a flexible layout at the site level and the current personalization scope must be a page.</p> <p>Flexible contents can be deleted, but not updated.</p> <p>Tip: Creating only one flexible content per flexible layout is recommended. If you create more than one flexible content for a single flexible layout, the flexible contents will be vertically aligned and you will not have the option of changing their order or alignment.</p>

Item Style	Description	Usage Notes
Form Value	A value submitted with a form but that is not displayed on the page.	Specify a View Instance and View Attribute from which this item derives its value. Refer to the <i>OA Component Reference</i> for information about other properties to set.
Formatted Text	Formatted text using generic HTML formatting tags.	You must enter an ID and Text for this item style. Refer to the <i>OA Component Reference</i> for information about other properties to set.
Image	An image can be a plain image that simply appears on the page, or can have a URL associated with it so that the user can click on it to navigate elsewhere. Although icon images are generally used as global buttons in Oracle Applications, an icon image may also be used next to text to visually quantify its content, as in the case of tips, or in line messages.	You must specify a value for Image URI with this item style. If you want to optionally associate a URL with the image, specify a value for the Destination URI property. Refer to the <i>OA Component Reference</i> for information about other properties to set.
Link	A link represents an action that you can perform on a region. A link can be used to navigate to another page or region.	You must enter a value for Text with this item style. For navigation, the Destination URI value must be a valid destination. Refer to the <i>OA Component Reference</i> for information about other properties to set.
Message Check Box	A checkbox, that when checked, specifies a value.	You must enter a value for Prompt with this item style. Refer to the <i>OA Component Reference</i> for information about other properties to set.
Message Choice	A poplist where users can select any value displayed in the poplist.	You must specify values for the following properties of this item style: <ul style="list-style-type: none"> • Prompt • Picklist View Instance • Picklist View Definition • Picklist Display Attribute • Picklist Value Attribute Refer to the <i>OA Component Reference</i> for information about other properties to set.

Item Style	Description	Usage Notes
Message Download	A file download link, which when selected opens a small Browser window that allows the user to either open the file and display its content or save the file to a client machine.	<p>You must specify values for the following properties:</p> <ul style="list-style-type: none"> • View Instance - view object instance of the underlying data source. • View Attribute - view attribute that maps to a column in the underlying data source. • File View Attribute - view attribute that maps to the column that stores the file content. <p>Refer to the <i>OA Component Reference</i> for information about other properties to set.</p>
Message File Upload	<p>An Upload File prompt, input field and Browse button that allows users to upload a file from a client machine to the middle tier.</p> <p>Note: If this item is associated with a view object instance and view attribute that returns a non-null value, that is, a file is already uploaded, OA Framework renders a Upload File prompt, View link and Clear button, so the user can view the already uploaded file, or clear it to upload another file.</p>	<p>You must specify values for the following properties:</p> <ul style="list-style-type: none"> • View Instance - view object instance of the underlying data source. • View Attribute - view attribute in the specified view object instance that maps to the column for storing the file content. <p>Refer to the <i>OA Component Reference</i> for information about other properties to set.</p>

Item Style	Description	Usage Notes
Message Inline Attachment	A list of inline attachment links that users can select to view a specific attachment. Also renders the “More...” link, which allows users to navigate to the Attachments user interface to perform further operations on attachments.	<ol style="list-style-type: none"> 1. You must enter the name of the View Instance for the entity you are attachment-enabling. 2. After creating Message Inline Attachment item, select the Create Item icon again for categoryMappings in the Page Hierarchy Personalization page to create a Category Map, page 5-24. 3. Select the Create Item icon for entityMappings under the Message Inline Attachment item in the Page Hierarchy Personalization page to create an Entity Map, page 5-25. 4. Finally, select the Create Item icon for primaryKeys under the Message Inline Attachment item in the Page Hierarchy Personalization page to create a Primary Key, page 5-31 if none has yet been defined for the entity map.
Message LOV Input	A LOV that allows users to select a value from a Search and Select List of Values window.	<p>Refer to the <i>OA Component Reference</i> for information about other properties to set.</p> <p>Follow the instructions outlined in <i>Creating a Message LOV Input Item</i>, page 5-33.</p> <p>Refer to the <i>OA Component Reference</i> for information about other properties to set.</p>
Message Radio Group	A radio group that allows a user to select one of several distinct values.	<p>You must specify values for the following properties of this item style:</p> <ul style="list-style-type: none"> • Prompt • Picklist View Instance • Picklist View Definition • Picklist Display Attribute • Picklist Value Attribute <p>Refer to the <i>OA Component Reference</i> for information about other properties to set.</p>

Item Style	Description	Usage Notes
Message Styled Text	A text field that can be updated and have a default value.	<p>You can optionally enter a value for the Prompt, or View Instance and View Attribute properties.</p> <p>If you wish to change the style of the text or associate a URL with the text, you can specify values for the CSS Class or Destination URI properties, respectively.</p> <p>Refer to the <i>OA Component Reference</i> for information about other properties to set.</p>
Message Text Input	A text input field.	<p>You must enter a value in the Prompt field for this item style.</p> <p>You may also optionally enter a value for the View Instance and View Attribute properties.</p> <p>Refer to the <i>OA Component Reference</i> for information about other properties to set.</p>
Primary Key	<p>A primary key associated with the Entity Map of an attachment entity.</p> <p>Note: This item style appears in the Item Style poplist only when no other primary key has been defined yet for the entity map and you choose the Create Item icon for primaryKeys under an Attachment entity hierarchy in the Page Hierarchy Personalization page.</p>	<p>You must enter a View Attribute name for the primary key.</p> <p>Refer to the <i>OA Component Reference</i> for information about other properties to set.</p>
Raw Text	Any type of text. For example, the raw text style allows you to enter HTML tags to display text in bold.	<p>You can optionally enter a value for the following properties: <i>Additional Text</i>, or <i>View Instance</i> and <i>View Attribute</i>.</p> <p>Refer to the <i>OA Component Reference</i> for information about other properties to set.</p> <p>Caution: In general, you should avoid using custom HTML as it does not meet the NLS, accessibility, security, and Oracle Browser Look-and-Feel (BLAF) UI guidelines and standards.</p>
Separator	A line to separate portions of a region. By default, the separator is rendered as a blue dotted line.	Refer to the <i>OA Component Reference</i> for information about other properties to set.

Item Style	Description	Usage Notes
Servlet Include	Includes HTML content loaded from a local Servlet or JSP under the parent region.	You must enter a value for Source URI to indicate the Servlet or JSP source. Refer to the <i>OA Component Reference</i> for information about other properties to set.
Spacer	A space that you can include in a region to improve the appearance of its layout.	You must specify values for spacer Width and spacer Height with this item style. Refer to the <i>OA Component Reference</i> for information about other properties to set.
Stack Layout	A stack layout region that renders its children regions and items in a vertical stack without any indentation.	Stack layouts may be created in any region, but only at the Site level. Refer to the <i>OA Component Reference</i> for information about other properties to set.
Static Styled Text	Text that is for display only and not for editing purposes. For example, if you wish to create a URL link, you would define its item style as static text.	You must enter a value for Prompt with this item style. If you wish to change the style of the text, you can specify the CSS Class whose style you want the text to inherit. Refer to the <i>OA Component Reference</i> for information about other properties to set.
Tip	Text that provides a tip about the contents of the region.	You must specify a Tip Message Name for this item style. Refer to the <i>OA Component Reference</i> for information about other properties to set.
URL Include	Includes HTML content loaded from an external source under the parent region.	You must enter a value for Source URI to indicate the source of this HTML content. No post-processing on the HTML is performed, so the included HTML should not include HTML tags like <html> or <body>. Refer to the <i>OA Component Reference</i> for information about other properties to set.

2. Enter a required ID for the new region item. Region item IDs should not exceed 30 characters, and they should be as short as possible. Abbreviations (which would be understood by consultants and customers) are encouraged. Item IDs must be unique within a single page regardless of its placement within the page hierarchy.
3. The properties that you can set for your new item are dynamically determined based on the item style that you select in Step 1. Check the Usage Notes column in the table shown in Step 1, page 5-21 for information about the specific properties you need to set for a specific item style. Note that the properties that you can set when you

create an item in the Create Item page may not be personalizable. Thus, for a specific item style, you may see more properties listed in the Create Item page, than what is shown for the same item style in the Personalize page.

If you must update some non-personalizable property of an item you created in the Create Item page, use the Update Item page, page 5-34.

Tip: If an item style allows you to specify a value for its CSS Class, you can use the CSS Styles Lookupelement, page 2-31 to first preview your item with different CSS styles before setting the CSS Class name for the item. See the section on Style Sheets, page 2-23 for additional information.

Note: If you want to create an item that allows you to specify a Destination URI, you should specify the URI syntax as: `<protocol>://<machine>:<port>/...` For example, `http://www.abcompany.com/home/page1.htm`. If you want to call an Oracle Applications function, specify the function name for the Destination Function property. For example, to call the PL/SQL General Preferences page from an item, you would enter the following value for the Destination Function property: `ICX_USER_PREFERENCES`. Make sure that the page specified as the Destination URI has a link that navigates you back to the original page. See Examples: Creating New Items Using the Personalization UI, page 4-17 for additional information.

4. Choose **Apply** to create the new item and return to the previous page. The new item is created as the last item in the region. You can use the Reorder Content control from the complete Page Hierarchy Personalization page, page 4-12 or the Page Layout Personalization page, page 4-5 to reorder the position of the new item.

Note: For configurable pages, you can only create a new item in a boxed content region or to the top pageLayout region. You cannot create a new item in a boxed layout region that contains only boxed layout regions as children.

Tip: If you make a mistake when you create a new item, such that it causes the base page to display an error, you can sign on to Oracle Applications as the Functional Administrator responsibility, page 4-2 and remove the personalization using the Manage Personalization Levels page, page 5-36.

Creating a Message LOV Input Item

Follow these steps if you wish to create a Message LOV Input item using the Create Item page:

1. Use the **Item Style** poplist to select **Message LOV Input**.
2. Enter a required ID for the new region item. Region item IDs should not exceed 30 characters, and they should be as short as possible. Abbreviations (which would be understood by consultants and customers) are encouraged. Item IDs must be unique within a single page regardless of its placement within the page hierarchy.

3. Enter values for the **External LOV** and **Prompt** properties.
4. Set the **Data Type** property to match the data type of the attribute returned from the LOV.
5. Choose **Apply** to create the new item and return to the Page Hierarchy. You should see a new grouping called **lovMappings** created under this new item. LOV Mappings define the data communication between the base page and the and LOV window.
6. Select the **Create Item** icon for **lovMappings** in the Page Hierarchy. In the Create Item page, create a related LOV Map for the Message LOV Input item. Specify a value for the required ID property and these other required properties:
 - **Criteria Item** - the base page item whose value is to be used as LOV search criteria
 - **LOV Region Item** - the item in the LOV for which the mapping is defined.
 - **Return Item** - the base page item whose value is to be populated by the LOV selection
7. Choose **Apply** to create the new item and return to the previous page. Repeat Step 6 to create additional LOV Maps.
8. When you finish creating your related LOV Maps, select **Return to Application** from the Page Hierarchy to view the new Message LOV Input item on your application page.

Update Item Page

When you choose the Update Item icon from the Page Layout Personalization page, page 4-5 or the complete Page Hierarchy Personalization page, page 4-12, you navigate to the Update Item page. Use the Update Item page to update the properties of an item that you created declaratively in the Create Item page, page 5-20 of OA Personalization Framework.

Important: You can only add a new item to a region if you are personalizing a region at the **Site**, **Function**, or **Localization** level. Any new item that you create while personalizing a region at the Function or Localization level may only be further personalized or updated at the same Function or Localization level at which it was created. New items that you create while personalizing a region at the Site level may later be personalized at any level. For example, you may personalize at the Responsibility level, an item that you create at the Site level.

Figure 5-13: Update Item page for a Tip item that was created.

ORACLE Diagnostics Home Logout Preferences Personalize Page

Update Item

* Indicates required field Cancel Apply

Level **Site**
Item Style **Tip**

Property	Value
* ID	NtfAddTip
Admin Personalization	true
Attribute Set	
CSS Class	
Comments	
Extends	
Rendered	true
Scope	
Text	
Tip Message Name	

Cancel Apply

Diagnostics | Home | Logout | Preferences | Personalize Page

Copyright 2000-2004 Oracle Corporation. All rights reserved. Privacy Statement

[About this Page](#)

The Update Item page displays the current personalization level, which can only be Site.

1. The properties that you can update for your item are dynamically determined by the item's style. Check the Usage Notes column in the table shown in Step 1, page 5-21 of the Create Item page for information about the specific properties that you can update for a specific item style. Note that the properties that you can set in the Update Item page may not be personalizable. Thus, for a specific item style, you may see more properties listed in the Update Item page, than what is shown for the same item style in the Personalize page.

Note: If an item style allows you to specify a value for its CSS Class, you can use the CSS Styles Lookup page, page 2-31 to first preview your item with different CSS styles before setting the CSS Class name for the item. See the section on Style Sheets, page 2-23 for additional information.

Note: If you want to update an item that allows you to specify a Destination URI, you should specify the URI syntax as:

```
<protocol>://<machine>:<port>/...
```

For example:

```
http://www.abcompany.com/home/page1.htm
```

If you want to call an Oracle Applications function, omit the *protocol://machine:port* specification in the URI, and only enter *OA.jsp?* with the following syntax: *OA.jsp?OAFunc=<Apps function you want to call>*. OA Framework automatically

prepends the correct syntax for the OA_HTML directory path so that your function runs properly. For example, to call the PL/SQL General Preferences page from an item, you would enter the following value for the Destination URI property:

```
OA.jsp?OAFunc=ICX_USER_PREFERENCES
```

Make sure that the page specified as the Destination URI has a link that navigates you back to the original page.

2. Choose **Apply** to update the new item and return to the previous page.

Tip: If you make a mistake when you update a new item, such that it causes the base page to display an error, you can sign on to Oracle Applications as the Functional Administrator responsibility, page 4-2 and remove the personalization using the Manage Personalization Levels page, page 5-36.

Manage Personalization Levels Page

The Manage Personalization Levels page is launched when you choose the **Manage Levels** button from the complete Page Hierarchy Personalization page, page 4-12 or the Page Layout Personalization page, page 4-5. You can use the Manage Personalization Levels page to select existing personalizations to activate, inactive or delete. You can also launch from this page, tasks to manage the translation of your personalizations.

Figure 5-14: Manage Personalization Levels page

ORACLE [Diagnostics](#) [Home](#) [Logout](#) [Preferences](#) [Personalize Page](#)

Manage Personalization Levels

Below is a list of personalization levels that have been defined for the page and/or region.

Page: Notifications

Select	Personalization Level	Value	Last Updated	Activate	Translate
	No Personalization Defined				

OA Footer

Select	Personalization Level	Value	Last Updated	Activate	Translate
	No Personalization Defined				

Shared Region: Notifications Worklist Function

Select Object: [Extract Translations](#) [Delete Personalizations](#) [Activate](#) [Deactivate](#) [Upload Translations](#)

[Select All](#) | [Select None](#)

Select	Personalization Level	Value	Last Updated	Activate	Translate
<input type="checkbox"/>	Site	Site	6-July-2004	Y	
<input type="checkbox"/>	Function	Compensation Workbench Home	16-July-2003	Y	
<input type="checkbox"/>	Function	Applications Home Page	10-July-2003	Y	
<input type="checkbox"/>	Responsibility	Workflow User Web Applications	5-July-2004	Y	

[Return to Personalization Structure](#)

[Diagnostics](#) | [Home](#) | [Logout](#) | [Preferences](#) | [Personalize Page](#)

Copyright 2000-2004 Oracle Corporation. All rights reserved. [About this Page](#) [Privacy Statement](#)

The Manage Personalization Levels page displays a table for each scope of the original page. For example, the original page might have three possible scopes: the Page itself, the OA Footer shared region, and the Notifications Worklist Function shared region. Shared regions are regions that can be shared among different pages. If you select a shared region as your scope when you set your personalization context, page 5-4, any personalizations you create get propagated to all occurrences of that shared region for the personalization level specified. If you select the page as your scope, then any personalizations you make are applied to just that local page. Refer to Per Instance versus Global Personalization, page 5-6 for more details.

1. Each table lists all the existing personalizations and identifies the personalization levels at which they were created (function, localization, site, organization, responsibility, "Oracle-seeded user-level" and "admin-seeded user-level"). The table identifies when the personalization was last updated and whether the personalization is active. You may select a personalization and use the **Activate** or **Deactivate** buttons to activate or deactivate the personalization.
2. If you wish to delete one or more personalizations, select the personalization(s) and choose **Delete Personalization**. Deleted personalizations are not recoverable.

Note: "Oracle-seeded" and "admin-seeded" user-level personalizations cannot be deleted from this page. If you wish to delete seeded user-level personalizations, you can do so from the Personal Views page, page 5-41.

3. Select the **Translate** icon for a personalization if you want to translate the personalization to another language. The Translate icon launches the Translate page, page 5-39 where you can translation your personalization inline. The Translate Column appears only if you have a language other than the base language installed on your system.

Note: "Oracle-seeded" and "admin-seeded" user-level personalizations cannot be translated. The **Translate** icon is disabled for these personalizations.

4. If you wish to extract translation files for one or more personalizations, select the personalization(s) and choose **Extract Translation File**. This button navigates to the Extract Translation Files page, page 5-37, where you can specify the languages to extract.
5. Select a personalization and choose **Upload Translations** if you want to upload the translations of a personalization from XLIFF files. This button launches the Upload Translations page, page 5-38.
6. Once you are done making changes to the Manage Personalization Levels page, select **Apply** to save your changes and return to the previous page.

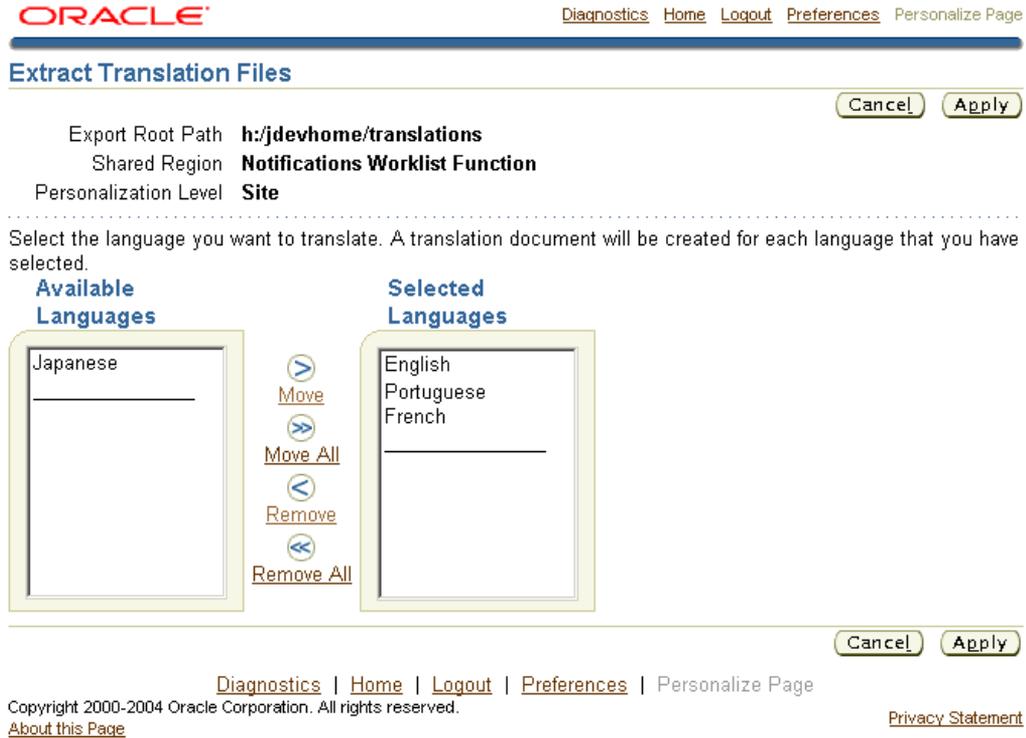
Extract Translation Files Page

The Extract Translation Files page is launched from the **Extract Translation File** button on the Manage Personalization Levels page, page 5-36. Use this page to select the languages for which you want to extract translation files.

Note: You must set the **Fnd Xliff Export Root Path** (FND_XLIFF_EXPORT_ROOT_PATH) profile option before you can extract translation

files. The directory that you specify for this profile option should have permissions set to read, write, create for all users, using `chmod 777 <dir_path>`.

Figure 5-15: Extract Translation Files page



1. Use the arrow buttons between the Available Languages and Selected Languages lists to shuttle selected languages between the two lists.
2. Choose **Apply** to extract the personalization translations for these languages to an oraXLIFF file, page 8-8 and return to the Manage Personalization Levels page. Language-specific subdirectories are created for each selected language in the package structure generated from the setting of the **Fnd Xliff Export Root Path** profile option.

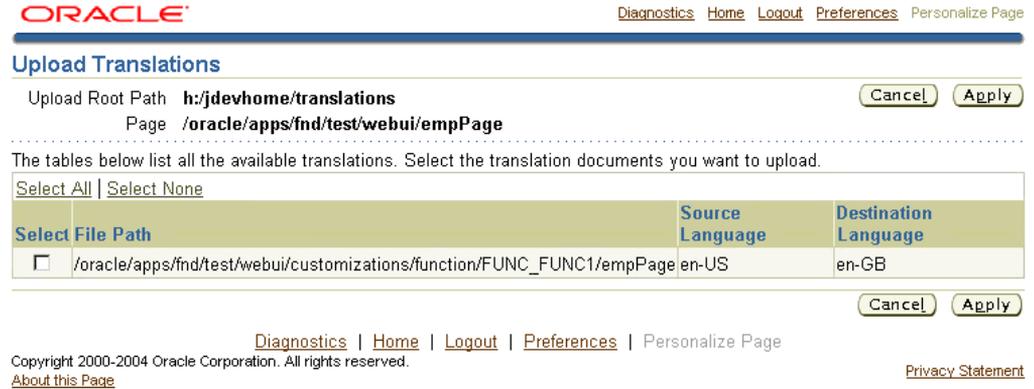
For more information about the **Fnd Xliff Export Root Path** profile option, refer to the **Personalization** section in the OA Framework Profile Options appendix of the *Oracle Application Framework Developer's Guide*.

Upload Translations Page

The Upload Translations page is launched from the **Upload Translations** button on the Manage Personalization Levels page, page 5-36. Use this page to upload available personalization translations from oraXLIFF files located in the language subdirectory of the package structure generated from the setting of the **Xliff Import Root Path** (FND_XLIFF_IMPORT_ROOT_PATH) profile option.

For more information about the **Xliff Import Root Path** profile option, refer to the **Personalization** section in the OA Framework Profile Options appendix of the *Oracle Application Framework Developer's Guide*.

Figure 5-16: Upload Translations page



The table lists the personalization translations that are available for upload. The table also identifies source language of the file and the language of the translation.

1. Select one or more personalization translation(s).
2. Choose **Apply** to upload the translations to the MDS repository and return to the Manage Personalization Levels page. The uploaded translations now also appear in the Translate page, page 5-39.

Translate Page

The Translate page is launched from the **Translate** icon on the Manage Personalization Levels page, page 5-36. Use this page to perform inline translations of your personalizations.

Figure 5-17: Translate page



The Translate page displays a table that lists all the page elements (**Style** column) and their translatable properties (**Attribute** column) that were modified in the selected personalization. It provides a read-only column that lists the personalized value for each property in the base language.

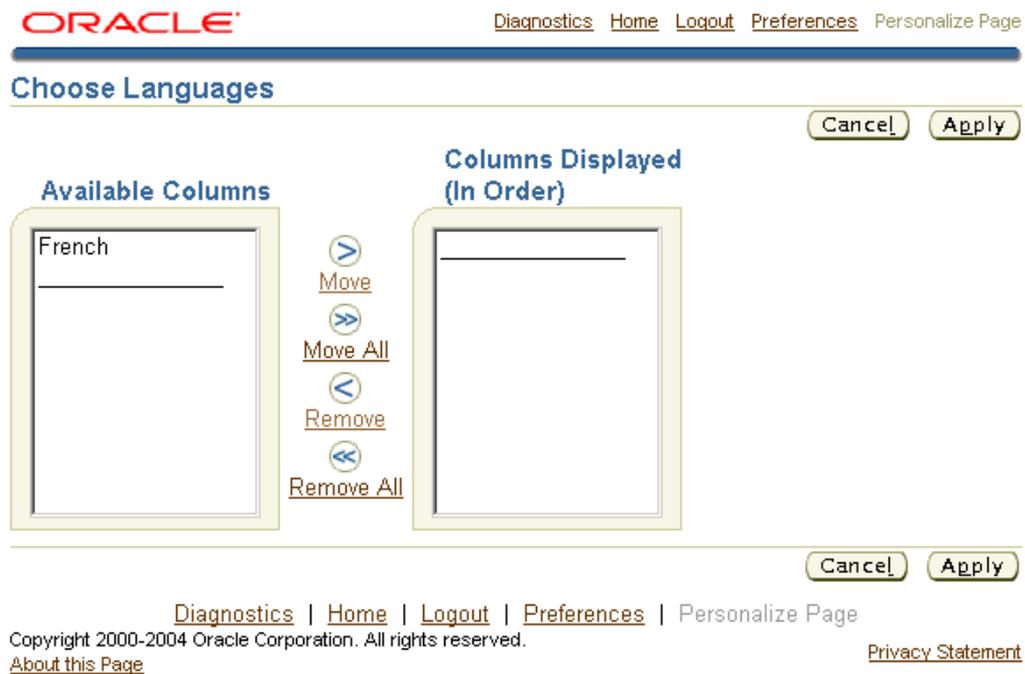
The table also shows columns for up to four languages. Any existing translations that have been previously uploaded, page 5-38 are displayed in these language columns. If a field in a language column is empty, the translation for that field is defaulted from the base language.

1. Select **Choose Languages** to navigate to the Choose Languages page, page 5-40, where you can select the languages (up to four) that display in the Translate page. When you are done choosing translation languages, return to the Translate page.
2. In the column for the language to which you want to translate the personalization, enter a translation value for the translatable property.
3. Repeat Step 2 for the other languages to which you want to translate the personalization.
4. Choose **Apply** to save your translations to the MDS repository and return to the Manage Personalization Levels page.

Choose Languages Page

The Choose Languages page is displayed when you select the **Choose Languages** button from the Translate page, page 5-39.

Figure 5-18: Choose Languages page



The languages that are currently installed are displayed in the Available Columns list.

1. Use the shuttle controls to select up to four languages to display in the Columns Displayed list.
2. Choose **Apply** to display the selected languages in the Translate page.

Personalize Views Page

The Personalize Views page is displayed when you select the **Seeded User Views** icon from the complete Page Hierarchy Personalization page, page 4-12 or the focused Page Hierarchy Personalization page, page 5-2. From this page, you can seed new end user views or duplicate, update, or delete pre-seeded end-user views.

Important: "Oracle-seeded" or "Admin-seeded" end user views can be created only for tables (including HGrids) in a query region.

Note: A single query region can be used by multiple pages. By default, saved views are stored at the page level; other pages that share the same query region will not provide access to the same set of saved views. However, to improve product usability, developers can choose to share views created on a page with all other pages that share the same query region. In this case, when you create or modify a view on that page, your changes are reflected on all of those pages.

For more information, see the Search section under "Implementing Specific UI Features" in the *OA Framework Developer's Guide*.

Figure 5-19: Personalize Views page

ORACLE [Diagnostics](#) [Home](#) [Logout](#) [Preferences](#) [Help](#) Personalize Page

Personalize Views

[Cancel](#) [Apply](#)

Below is a list of all pre-configured and/or personalized views applicable to "Worklist" table on the previous screen.

Select View: [Duplicate](#) | [Create View](#)

Select	View Name	Description	Display View	Default	Update	Delete
<input type="radio"/>	All Notifications		Yes ▾			
<input type="radio"/>	FYI Notifications		Yes ▾			
<input type="radio"/>	Notifications From Me		Yes ▾			
<input type="radio"/>	Open Notifications		Yes ▾			
<input type="radio"/>	To Do Notifications		Yes ▾			

[Cancel](#) [Apply](#)

[Diagnostics](#) | [Home](#) | [Logout](#) | [Preferences](#) | [Help](#) | Personalize Page

Copyright 2000-2004 Oracle Corporation. All rights reserved. [Privacy Statement](#)
[About this Page](#)

If you are an Oracle in-house E-Business Suite developer, only "Oracle-seeded" end user views are displayed in the Personalize Views table. If you are an Administrator, "Oracle-seeded" and "Admin-seeded" end user views are displayed and the only action you can perform on an "Oracle-seeded" view is to duplicate it.

Note: As an Administrator, you cannot update or delete an "Oracle-seeded" end user view. Instead, if you want to personalize the

view, you should duplicate the view and personalize the duplicate. This ensures that if Oracle ever ships changes to an "Oracle-seeded" end user view, it will never override your personalizations, because you will have personalized not the "Oracle-seeded" view, but a duplicate of the "Oracle-seeded" view.

Note: When an end-user chooses to personalize a table in a Query region, the user sees all "Oracle-seeded" and "Admin-seeded" end user views, in addition to the personalized views he or she creates, in the Personalize Views page. Although users can select any "Oracle-seeded" or "Admin-seeded" user view as their default view, they cannot update or delete the seeded view. Instead they can duplicate the seeded view and personalize the duplicate to ensure that their personalizations are preserved when there is an upgrade. See Default Views, page 3-6 for additional information.

The Personalize Views page displays the name of each view, along with a description.

- If you want to make a copy of a view, select the view you wish to copy and choose **Duplicate**. This makes a duplicate of the view and takes you to the Duplicate View page, page 5-42 where you can personalize the duplicate.
- If you want to modify an existing view, select **Update** to navigate to the Update View page, page 5-42 where you can personalize the view. Note that as an Administrator, you cannot update "Oracle-seeded" views.
- If you want to delete an existing view, select **Delete**. Note that as an Administrator, you cannot delete "Oracle-seeded" views.
- To create a new view, select **Create View** to navigate to the Create View page, page 5-42 where you can create a new personalization for the Query table (or HGrid) region.
- Use the **Display View** poplist to control whether the view displays in the View poplist of the Views panel for end users and in the Personalize Views page for end user personalizations.

Create View, Update View, and Duplicate View Pages

The Create View page is identical to the Duplicate View and Update View pages, except that the fields in the latter pages are pre-populated with settings from the selected personalization.

Figure 5-20: Create View page

ORACLE

[Diagnostics](#) | [Home](#) | [Logout](#) | [Preferences](#) | [Help](#) | [Personalize Page](#)

Create View

Below is a list of attributes that can be edited to change the view and/or filter the data that is displayed in your table.

* Indicates required field

General Properties

* View Name

Number of Rows Displayed Set as Default

Securing Function

Description

Column Properties

Update the appropriate column attributes as desired.

Columns Shown and Column Order

Available Columns

To

Information Requested From

Closed

Priority

Notification ID

From Me

>

Move

>>

Move All

<

Remove

<<

Remove All

Columns Displayed

Subject

From

Sent

Type

Status

Due

TIP Columns with totaling capabilities shown can only display as the end column of the table.

Sort Settings

	Column Name	Sort Order
First Sort	Sent	descending
Second Sort	Subject	ascending
Third Sort	From	ascending

Search Query to Filter Data in your Table

Specify parameters and values to filter the data that is displayed in your table.

Advanced Search

Show table data when all conditions are met.
 Show table data when any condition is met.

Subject
 From
 Message Attribute
 Sent

Add Another

[Diagnostics](#) | [Home](#) | [Logout](#) | [Preferences](#) | [Help](#) | [Personalize Page](#)

Copyright 2000-2004 Oracle Corporation. All rights reserved. [Privacy Statement](#)

[About this Page](#)

Important: Although a region may be personalizable, not all region items in a region are personalizable. Each region item in a region has an ADMIN_CUSTOMIZABLE property associated with it. If the property is set to false by the developer of the region, the region item is not personalizable at the Admin level. For example, a developer would very likely set ADMIN_CUSTOMIZABLE to false for the vertical spacer item in a table layout region to prevent disruption of the spacing in a table.

1. Edit the following **General Properties** for your table or HGrid region:
 - **View Name** - the name of the personalized end user view as it appears in the Personalize Views page, page 5-41 or the Personal Table Views page, page 6-3 for end-users.
 - **Number of Rows Displayed** - the number of rows of data you wish to display in the table.
 - **Set View as Default** - check if you want to make this personalized view the default view. As an Administrator, you can only set an "Admin-seeded" end user view as a default.

Note: If an "Oracle-seeded" end user view and an "Admin-seeded" end user view of the same region are both marked as defaults, the "Admin-seeded" end user view takes precedence and is displayed as the default for the end user. However, if an end user also has a personalized view of the same region that he or she marks as the default, then the end user's personalized view takes precedence over both seeded views and is the default for that end user. See Default Views, page 3-6 for additional information.
 - **Securing Function** - the function for which this view is available. Specifying a securing function allows you to restrict the use of this view to only users who have been granted access to that function. Refer to the section on Securing Views, page 5-46 for complete information.
 - **Description** - an optional description for this personalized view.
2. Use the **Columns Shown and Column Order** shuttle to specify the columns to display in the table or HGrid region. Select an item from the **Available Columns** or **Columns Displayed** list and use the arrows between these lists to either move the selected item to or remove the selected item from the Columns Displayed list.

Note: Columns that are required fields table appear with an asterisk (*) and cannot be removed from the Columns/Attributes Displayed list.

Example

Following is an example of an advanced table with two levels of column spans.

Group X				Group Y			
Group A		Group B		Group A		Group B	
Column 1	Column 2	Column 3	Column 4	Column 1	Column 2	Column 3	Column 4

Each of the columns in the top row (labeled "Group X" and "Group Y") spans two columns in the second row (labeled "Group A" and "Group B" in both cases). Each "Group A" column in turn spans "Column 1" and "Column 2" below it, while each "Group B" column spans "Column 3" and "Column 4."

If the region you are personalizing is an Advanced Table that displays column spans, as shown in the example, the columns listed in the Available Columns and Columns Displayed shuttle will have the entire hierarchy of column group names appended to each column name. This ensures that you hide/show the correct column, especially in the case where multiple columns of the same name may exist within different column groups. For the example shown, the columns shown in the Available Columns and Columns Displayed lists would be:

```
Group X: Group A: Column 1
Group X: Group A: Column 2
Group X: Group B: Column 3
Group X: Group B: Column 4
Group Y: Group A: Column 1
Group Y: Group A: Column 2
Group Y: Group C: Column 3
Group Y: Group C: Column 4
```

3. Once you are satisfied with the columns to display, use the arrows to the right of the **Columns Displayed** list to reorder their sequence.
4. Choose **Rename Columns / Totaling**, page 5-49 to rename or show a total for a column.
5. You can specify up to three levels of sorting for your data in the **Sort Settings** region. Select a column from the **Column Name** poplist for each level of sorting you wish to perform.

Note: Only columns whose Sort Allowed property is set to Yes are displayed in the Column Name poplist. If this property is initially set to no and you want to be able to sort on this column, you must first create an admin-level personalization of the table region that updates the Sort Allowed property for the column to yes.

6. For each sort column, specify whether to sort in ascending or descending order.
7. You can filter the data that is displayed in the table based on criteria that you specify in the **Search Query to Filter Data in Table** region:
 1. Indicate how you want the filter to match your search conditions by selecting one of the following radio buttons: **Show table data when all conditions are met** or **Show table data when any condition is met**.

2. The first four columns of the table are listed for you to specify search criteria. Using the poplist following the column name, choose a search condition and enter a value to search for in that column.
3. Select a column from the **Add Another** poplist and choose **Add** to add more search criteria to your filter.

Note: Only columns whose Search Allowed property is set to True are displayed in the Add Another poplist. If this property is initially set to False and you want to be able to search on this column, you must first create an admin-level personalization of the table region that updates the Search Allowed property for the column to True.

Tip: If you leave the search criteria blank for a column, the filter will not search on that column.

8. When you are done personalizing your view of the table or HGrid region, you can choose **Apply** to save the personalized view and return to the Personalize Views page. If you choose **Revert** to revert to default settings, the following occurs depending on the page you are using:
 - Create View page - the page defaults to the pre-seeded display settings, where no query options are set.
 - Duplicate View or Update View page - the page defaults to the display settings and query options of the saved existing view.

Note: How you set the Scope of your personalization context in the Choose Personalization Context page, page 5-4 affects how your personalized view is saved. The view may be saved to the current instance of the page as a per instance personalization if you set the Scope to Page. If there is a shared region on the page and you select the Shared region as the Scope, the personalization you create on that shared region is also saved as a per instance personalization of that shared region but will appear in all occurrences of that shared region in any page that extends it. Refer to Per Instance versus Global Personalization, page 5-6 for further details.

Securing Views

The way in which OA Personalization Framework secures views through the Securing Function property in the Create/Update/Duplicate View pages has changed:

- *Old Behavior prior to OA Framework 11.5.10.2CU* - As an administrator, you are not able to update or delete "Oracle-seeded" end user views. As a result, seeded views are always visible to all end users. For example, you cannot update "Oracle-seeded" views to restrict them to a specific set of end users or delete some of these views for all end users.

In addition, if an "Oracle-seeded" or "Admin-seeded" end user view had a securing function associated with it, only end users who were granted access to that securing function were able to see that view in the Views poplist of the Views panel, page 6-2 of their Search page. Unfortunately, the securing function did not prevent that view from appearing in the Personalize Views page, page 6-3, which any user can navigate to, by selecting the Personalize button in the Views panel. As a result, any

end user was able to select the secured view from the Personalize Views page and duplicate it, even without being granted access to it.

- *New Behavior as of OA Framework 11.5.10.2CU* - The Securing Function, in addition to restricting visibility of a secured view in the Views poplist of the Views panel, now also restricts visibility of a secured view in the Personalize Views page.

In addition, as an administrator, if you want to turn off some "Oracle-seeded" views, you can duplicate the seeded view and set the securing function on the duplicated view to a function to which a user or group of users does not have access.

Finally, if you want to simply control the visibility of a view for an end user in the Views poplist of the Views panel, first duplicate the view and set a securing function on the duplicate, granting the user access to that function. Then in the Personalize View page, set Display View to **Yes** or **No** for that duplicated view.

The following table identifies the accessibility of views to administrators as well as end users, when a securing function is specified for "Oracle -seeded" and "Admin-seeded" end user views.

Created By	Active	Effect on an Administrator's Personalize Views Page	Effect on an End User's Personalize Views Page	Effect on the Views Poplist in the Views Panel
Oracle	Yes	Securing function: Yes Shown: Yes Can be deleted: No Can be modified: No Can be duplicated: Yes	Shown: Only if user has access to the securing function. Can be deleted: No Can be modified: No Can be duplicated: Yes	Shown only if user has access to the securing function.
Oracle	No	Securing function: Yes Shown: Yes Can be deleted: No Can be modified: No Can be duplicated: Yes	Shown: Only if user has access to the securing function. Can be deleted: No Can be modified: No Can be duplicated: Yes	Not shown, since the Active property is set to No.
Administrator	Yes	Securing Function: Yes Shown: Yes Can be deleted: Yes Can be modified: Yes Can be duplicated: Yes	Shown: Only if user has access to the securing function. Can be deleted: No Can be modified: No Can be duplicated: Yes	Shown only if user has access to the securing function.
Administrator	No	Securing Function: Yes Shown: Yes Can be deleted: Yes Can be modified: Yes Can be duplicated: Yes	Shown: Only if user has access to the securing function. Can be deleted: No Can be modified: No Can be duplicated: Yes	Not shown, since the Active property is set to No.
End User	Yes	Not applicable	Shown: Yes Can be deleted: Yes Can be modified: Yes Can be duplicated: Yes	Shown: Yes
End User	No	Not applicable	Shown: Yes Can be deleted: Yes Can be modified: Yes Can be duplicated: Yes	Not shown

Rename Columns / Totaling Page

Use the Rename Columns / Totaling page to rename the table or HGrid columns that are displayed in your personalized view, to display a total for a column.

Figure 5-21: Rename Columns / Totaling page

ORACLE [Diagnostics](#) [Home](#) [Logout](#) [Preferences](#) [Help](#) [Personalize Page](#)

Rename Columns / Totaling

[Cancel](#) [Apply](#)

Rename the columns below that will be displayed in your table. Select the checkbox in the "Show Total" column as applicable, if you would like a total to display.

Original Column Name	New Column Name	Show Total
Subject	<input type="text" value="Subject"/>	<input type="checkbox"/>
From	<input type="text" value="From"/>	<input type="checkbox"/>
Sent	<input type="text" value="Sent"/>	<input type="checkbox"/>
Type	<input type="text" value="Type"/>	<input type="checkbox"/>
Status	<input type="text" value="Status"/>	<input type="checkbox"/>
To	<input type="text" value="To"/>	<input type="checkbox"/>
Information Requested From	<input type="text" value="Information Requested From"/>	<input type="checkbox"/>
Due	<input type="text" value="Due"/>	<input type="checkbox"/>
Closed	<input type="text" value="Closed"/>	<input type="checkbox"/>
Priority	<input type="text" value="Priority"/>	<input type="checkbox"/>
Notification ID	<input type="text" value="Notification ID"/>	<input checked="" type="checkbox"/>
From Me	<input type="text" value="From Me"/>	<input type="checkbox"/>

[Cancel](#) [Apply](#)

[Diagnostics](#) | [Home](#) | [Logout](#) | [Preferences](#) | [Help](#) | [Personalize Page](#)
Copyright 2000-2004 Oracle Corporation. All rights reserved. [Privacy Statement](#)
[About this Page](#)

1. You can change the **Column Names** for the columns you chose to display.
2. You may check **Show Total** to turn on totaling for a specific column in your table, if it is applicable to the underlying data.

Tip: If you choose to display a column with totaling capabilities, you may want to display this column as the last column of the table. You can reorder the columns in the Columns Displayed list located on the previous page.

3. If after you make changes to your columns, you decide you want to revert back to the default values, choose **Reset to Defaults**.
4. Choose **Apply** when you are done to save your changes and return to the previous page.

User-Level Personalization User Interface

This chapter covers the following topics:

- Overview
- Personalizing the Views Panel of a Search Page
- Personalizing a Table Region at the User Level
- Personalize Views Page
- Create View, Update View, and Duplicate View Pages
- Rename Columns / Totaling Page
- Personalizing an LOV Choice List
- Known Issues

Overview

In Oracle applications, the following page elements may be end-user personalized:

- Views panel of a Search page
- LOV Choice List

Personalizing the Views Panel of a Search Page

In OA Framework-based applications, tables are used to display results from a search. The region above the table generally contains the search panel where you specify the search criteria. You may see any one of these possible search panels above a table:

- Simple Search - allows you to specify simple search criteria.
- Simple Search or Advanced Search - buttons allow you to toggle between a simple search panel and an advanced search panel to specify search criteria.

If a table is user-personalizable, it will support user-personalizable searches that you can save as *Views*. The saved views can be selected from the Views panel, which you access using the **Views** button on the main search page.

Note: A single query region can be used by multiple pages. By default, saved views are stored at the page level; other pages that share the same query region will not provide access to the same set of saved views. However, to improve product usability, developers can choose to share views created on a page with all other pages that share the same

query region. In this case, when you create or modify a view on that page, your changes are reflected on all of those pages.

The first time you access a Views-enabled page (and you have no saved searches to run) the following occurs:

- If the page only has a Views panel, the Views panel displays with an empty View poplist. You need to select the **Personalize** button to create one or more saved searches.
- If the page has a Views panel and a Simple Search panel, the Simple Search panel renders. You need to execute a query and select the **Save Search** button to access the personalized query in the Views panel.
- If the page has a Views panel and an Advanced Search panel, the Advanced Search panel renders. You need to execute a query and select the **Save Search** button to access the personalized query in the Views panel.

Often times, if a table is end-user personalizable, seeded personalized views for that table may already exist and be displayed in the View poplist.

Note: When the Views panel is displayed, the table in the Query region displays the personalized view that you previously marked as the default. If you did not specify a default personalized view, then any "admin-seeded user-level" view or "Oracle-seeded user-level" view that is set as a default is displayed, where the Admin-seeded view takes precedence over the Oracle-seeded view.

Tip: When you navigate to the Search or Advanced Search panel, the table in the Query region no longer reflects the user-defined personalizations, but rather, reflects the base view of the table with any admin-level personalizations that apply. If you have the appropriate privileges, you can create Admin-level personalizations, page 4-1 for the table in the Simple and Advanced Search panels.

Personalizing a Table Region at the User Level

You can create a personalized view of a table in a Query page using one of two methods:

Method 1:

1. Navigate to the application page that contains the personalizable table you wish to alter.
2. Use the search panel to query for specific results in the table.
3. Select **Save Search** to save the search criteria that queried for these results.
4. When you select Save Search, the Create View page appears, allowing you to save this search as a new personalized view of the table. You can also edit other attributes and properties of the table.
5. Once you apply your changes, the saved search becomes a personalized view that you can select from a panel that lists your personal views.

Method 2:

1. Navigate to the application page that contains the table you wish to personalize.
2. If the page does not already show the Views panel, select **Views**.

Note: If the region does not have any predefined personalizations, the Views panel does not appear. Instead the Search region appears and you can create a personalization by choosing the **Save Search** button after specifying and performing a search.

3. In the Views panel, you have two options:
 - Apply a specific personal view, if one exists, by selecting a view from the **View** poplist and choosing **Go**.
 - Create a new view or update an existing view by choosing **Personalize** to display the Personalize Views page.

Restrictions

OA Personalization does not retain the view selection for the current Query page when you move to another page. For example, suppose you display a personalized view of the Query page by selecting a view (such as View X) from the Views panel. Then you select the Personalize button to display the Personalize Views page, but choose Cancel without making any changes. When you return to the Query page, the base definition is displayed and not View X. As a developer, if you do not want this behavior and instead, want to retain the view, you need to modify the controller for the page to pass the user-selected view as a parameter and set that view as the current personalization when ever the user returns to the Query page.

Personalize Views Page

The Personalize Views page displays a description of each view.

1. Select a view and choose **Update**, **Duplicate** or **Delete**.
 - In the Update View page, the fields are pre-populated with data from the saved view that you want to update.
 - The Duplicate View page is identical to the Update View page and is also pre-populated with data from the original view that you want to duplicate. The **View Name** field also shows a default value of "Copy of [ViewName]".
 - If you choose to delete a view, a warning message appears to confirm that you really want to delete the message.

Note: If the view is an "admin-seeded user-level" view or "Oracle-seeded user-level" view, you cannot update or delete the view as an end-user. If you wish to modify a seeded view, you must make a duplicate of the seeded view and modify the duplicate. This ensures that when Oracle ships a new release that contains changes to its seeded views, your personalizations are not overwritten.

2. If you wish to create a new personalized view of the table, choose **Create View**.

3. Use the **Display View** poplist to control whether the view displays in the View poplist of the Views panel.

Create View, Update View, and Duplicate View Pages

The Create View page is identical to the Update View and Duplicate View pages, except that the fields in the latter pages are pre-populated with settings from the selected view.

Per Instance Personalizations

Prior to OA Framework 11.5.10, when you saved a view of a query region that was a shared region (that is, the query region was actually defined in some other base region, and the current region shared or extended that base region), the view was actually saved to the base region. Therefore, if another page also shared that same base query region, the view you saved was seen on that page as well. In 11.5.10 and on, the view you save on a query region that extends another base region, is now saved on the current instance of the region you are using and not on the base region. So if another page extends the same base region, you will not see your saved view in that other page.

In 11.5.10 and on, Administrators are able to seed user-level views that apply to either the base region of a shared region, or to a specific instance of a shared region (per-instance view).

As a user, you can see the list of all Admin-seeded per-instance views for a region in the Views poplist of the View panel, as well as all your own personalized views (including those that may have been saved to the base region prior to 11.5.10). Although you cannot modify an Admin-seeded view, you can always create a duplicate of any Admin-seeded per-instance view and mark the duplicate as your default view, page 6-4.

To specify general properties:

1. Enter a user-friendly **View Name** to identify your personalization.
2. Select the number of rows of data you wish to display in the table in your personalized view.
3. Check **Set as Default** if you wish to make this view your default view. There can only be one default view at any time. If you check Set as Default for the current view, than any view that you previously marked as the default becomes unmarked.

Note: If you wish to mark a predefined seeded personalized view as your default, create a duplicate of the seeded view, and check **Set as Default** for the duplicate view.

Note: If an "Oracle-seeded" end user view and an "Admin-seeded" end user view of the same region are both marked as defaults, the "Admin-seeded" end user view takes precedence and is displayed as the default. If, however, you also create your own personalized view of the same region and set it as the default, then your personalized view takes precedence over both levels of seeded views and becomes your default. See Default Views, page 3-6 for additional information.

4. Enter a description for this personalized view.

To specify column properties:

The columns of the table region appear in the Available Columns list. Edit the Column Properties to specify the columns you wish to display and the order in which to display them.

1. Select an item from the **Available Columns** or **Columns Displayed** list and use the buttons between these lists to shuttle the selected item between the two lists.

Note: Columns that are required fields in a page appear with an asterisk (*) and cannot be removed from the Columns Displayed list.

Example

Following is an example of an advanced table with two levels of column spans:

Group X				Group Y			
Group A		Group B		Group A		Group B	
Column 1	Column 2	Column 3	Column 4	Column 1	Column 2	Column 3	Column 4

Each of the columns in the top row (labeled "Group X" and "Group Y") spans two columns in the second row (labeled "Group A" and "Group B" in both cases). Each "Group A" column in turn spans "Column 1" and "Column 2" below it, while each "Group B" column spans "Column 3" and "Column 4."

If the region you are personalizing is an Advanced Table that displays column spans, as shown in the example, the columns listed in the Available Columns and Columns Displayed shuttle will have the entire hierarchy of column group names appended to each column name. This ensures that you hide/show the correct column, especially in the case where multiple columns of the same name may exist within different column groups. For the example, the columns shown in the Available Columns and Columns Displayed lists would be:

```
Group X: Group A: Column 1
Group X: Group A: Column 2
Group X: Group B: Column 3
Group X: Group B: Column 4
Group Y: Group A: Column 1
Group Y: Group A: Column 2
Group Y: Group B: Column 3
Group Y: Group B: Column 4
```

2. Once you are satisfied with the items to display, select an item from the **Columns Displayed** list and use the buttons to the right of the list to reorder the sequence in which the item appears.
3. Choose Rename Columns / Totaling, page 6-6 to alter other settings for your columns.

To specify sort settings:

1. You can specify up to three levels of sorting for your data. Select a column from the **Column Name** poplist for each level of sorting you wish to perform.

2. For each column to sort, you must specify whether to sort in ascending or descending order.

Note: Only columns whose User Personalization property is set to **True** by the developer or administrator, appear in the Column Name poplist.

To filter data in the table:

You can filter the data that is displayed in the table based on criteria that you specify.

1. Indicate how you want the filter to match your search conditions by selecting one of the following radio buttons:
 - **Show table data when all conditions are met.**
 - **Show table data when any condition is met.**
2. The first four columns of the table are listed for you to specify search criteria. Using the poplist following the column name, choose a search condition and enter a value to search for in that column.
3. Select a column from the **Add column** poplist and choose **Add** to add more search criteria to your filter.

Note: Only columns whose Search Allowed property is set to True by the developer or administrator, appear in the Add Another poplist.

4. If you leave the search criteria blank for a column, the filter will not search on that column.

To save your personalized view:

1. When you are done personalizing your view of the table, choose **Apply** or **Apply and View Results**. Choosing **Apply** saves the personalized view and returns you to the Personalize Views page. Choosing **Apply and View Results** saves the view you just created or updated, and returns you to the application page where the view you just created is set as the current personalized view.
2. If you choose **Revert** to revert to default settings, the following occurs depending on the page you are using:
 - Create View page - the page defaults to the pre-seeded display settings and no query options are set.
 - Update View or Duplicate View page - the page defaults to the display settings and query options of the saved existing view you are trying to update or duplicate.

Rename Columns / Totaling Page

1. Use this page to change the labels of the table columns.
2. Check **Show Total** to turn on totaling for a specific column, if it is applicable to the underlying data.

Tip: If you choose to display a column with totaling capabilities, make sure this column is displayed as the last column of the table.

3. Choose **Apply** to accept your changes and return to the Create View or Update View page.

Personalizing an LOV Choice List

An LOV Choice List is a hybrid between a poplist and a List of Values. If the LOV Choice List is implemented with a Personalize button, you can personalize the LOV Choice List by adding, removing, or reordering values in the list.

1. Select the **Personalize** button next to the LOV Choice List. (For a LOV Choice List within a table, the Personalize button is displayed in the table action area. If there are multiple LOV Choice List columns within a table, a separate Personalize button is rendered in the table action area for each of those columns.)
2. A Personalize (LOV Choice List) page appears, containing a shuttle region that displays an Available and a Selected list. To avoid performance problems in case the complete list of values is very large, the Available list is rendered empty when the page displays. Use the data filter above the Available list to search for a subset of values to display.
3. Use the shuttle controls to add or remove values from the LOV Choice List.
4. Use the reorder buttons next to the Selected list to reorder the values in the LOV Choice List.
5. Choose **Apply** to save your personalization.

Known Issues

See a summary of key Personalization issues, page B-1 with suggested workarounds if available.

Portlet-Level Personalization User Interface

This chapter covers the following topics:

- Overview
- Personalizing a Portlet
- Create View and Update View Pages
- Rename Columns / Totaling
- Known Issues

Overview

In Oracle E-Business Suite, a specific region of an application page can be displayed as a portlet in Oracle Portal. Oracle Portal provides you with a common, integrated starting point for accessing all your data. Since Oracle Portal lets you personalize the content and look of your page, you can also personalize the application region that is displayed as a portlet. Any personalizations you make to that portlet region appear only when you display that region from the same portlet.

Note: If a portlet contains a **More...** link that drills down to a personalizable region, then any portlet-level personalization you make to that region also applies to the drill-down region (from the **More...** link).

Personalizing a Portlet

1. Display Oracle Portal in your web browser and login.
2. Select the link to display the portlet region you wish to personalize.
3. Select the **Personalize Region** link within the portlet region.
4. Use the Create View or Update View page that appears to make your personalizations to the portlet region.

Create View and Update View Pages

The Create View page is identical to the Update View page, except that the fields in the latter page are pre-populated with settings from a prior personalization.

To specify general properties:

If your portlet region is not a table, the Create View or Update View page displays only the General Properties of the region for you to personalize.

1. Enter a user-friendly **View Name** to identify your personalizations.
2. Select the number of rows of data you wish to display in your personalized view.
3. Enter a description for this personalized view.

To specify column properties:

If the portlet region is a table, the columns of the table region appear in the Available Columns list. Edit the Column Properties to specify the columns you wish to display and the order in which to display them.

1. Select a column from the **Available Columns** or **Columns Displayed** list and use the buttons between these lists to shuttle the selected item between the two lists.

Note: Columns that are required fields in a page appear with an asterisk (*) and cannot be removed from the Columns Displayed list.

Tip: If the region you are personalizing is an Advanced Table that displays column spans, the columns listed in the Available Columns and Columns Displayed shuttle will have the entire hierarchy of column group names appended to each column name. This ensures that you hide/show the correct column, especially in the case where multiple columns of the same name may exist within different column groups. For example:

```
Group X: Group A: Column 1
Group X: Group A: Column 2
Group X: Group B: Column 3
Group X: Group B: Column 4
Group Y: Group A: Column 1
Group Y: Group A: Column 2
Group Y: Group B: Column 3
Group Y: Group B: Column 4
```

For an illustration of this example, see To specify column properties, page 6-5 in the **Create View, Update View, and Duplicate View Pages** section of the User-Level Personalization User Interface chapter.

2. Once you are satisfied with the columns to display, select an item from the **Columns Displayed** list and use the buttons to the right of the list to reorder the sequence in which the item appears.
3. Choose Rename Columns / Totaling, page 7-3 to alter other settings for your columns.

To specify sort settings:

1. You can specify up to three levels of sorting for your data. Select a column from the **Column Name** poplist for each level of sorting you wish to perform.
2. For each column to sort, you must specify whether to sort in ascending or descending order.

Note: Only columns whose User Personalization property is set to **True** by the developer, appear in the Column Name poplist.

To filter data in the table:

You can filter the data that is displayed in the table based on criteria that you specify.

1. Indicate how you want the filter to match your search conditions by selecting one of the following radio buttons:
 - **Show table data when all conditions are met.**
 - **Show table data when any condition is met.**
2. The first four columns of the table are listed for you to specify search criteria. Using the poplist following the column name, choose a search condition and enter a value to search for in that column.
3. Select a column from the **Add column** poplist and choose **Add** to add more search criteria to your filter.

Note: Only columns whose Search Allowed property is set the True by the developer, appear in the Add Another poplist.

4. If you leave the search criteria blank for a column, the filter will not search on that column.

To save your personalized view:

1. When you are done personalizing your view of the portlet region, choose **Apply**.
2. If you choose **Revert** to revert to default settings, the following occurs depending on the page you are using:
 - Create View page - the page defaults to the preseeded display settings and no query options are set.
 - Update View or Duplicate View page - the page defaults to the display settings and query options of the saved existing view you are trying to update or duplicate.

Rename Columns / Totaling

1. You can change the name of the columns that you chose to display.
2. Check **Show Total** to turn on totaling for a specific column, if it is applicable to the underlying data.

Tip: If you choose to display a column with totaling capabilities, make sure this column is displayed as the last column of the table.

3. Choose **Apply** to accept your changes and return to the Create View or Update View page.

Known Issues

See a summary of key Personalization issues, page B-1 with suggested workarounds if available.

Translating Personalizations

This chapter covers the following topics:

- Overview
- Translating Personalizations Stored in the AK Repository
- Translating Personalizations Stored in MDS
- OA Extension Translation Toolset
- Mapping a Language Code to a Language-Territory Code
- Known Issues
- Related Information

Overview

OA Framework allows you to personalize HTML pages so you can tailor Oracle Applications to better fit your organizational needs. Using the Personalization user interface, you can modify the text on a page by either changing the prompt for an item or by adding new items to a page. You can then translate this text, into any installed languages at your site. Translation of personalized text is available only at the administrative level and not at the end-user level. End user personalizations are maintained in the selected user session language.

To translate your personalizations and manage your translations inline in the OA Personalization Framework UI, use the Manage Personalization Levels page, page 5-36 of the Administrative-Level Personalizations UI, page 4-1.

To perform bulk translations of your personalizations, you can use the set of command line translation tools, following the steps outlined in this document for page definitions stored either in AK, page 8-1 or MDS, page 8-3. Note that for OA Framework 11.5.10 and above, all page definition metadata is stored in the MDS repository.

Translating Personalizations Stored in the AK Repository

If you personalize a page that still resides in the AK Repository, page 10-1, you can translate the personalization using the following steps:

Note: The executables that you use to export and import these translation files are described in the Oracle Applications System Administrator's Guide. These include `akload`, page 9-12 and `FNDLOAD` to extract and upload page meta data and upload translations.

1. Go to the **APPL_TOP** directory and source the file **APPSORA.env** to set all your environment variables. For Windows, refer to the AD Procedures Guide to display an appropriate command prompt window and use the resulting command prompt window to set your environment variables (run %APPL_TOP%\envshell.cmd).
2. Use **akload**, page 9-12 to extract the base region that you personalized. You need to identify the **application short name** and the **region code** of the personalized region. This information is available on the first HTML page of the personalization transaction flow. You need to convert the **region application id** shown in the HTML page to the **application short name** that is required by **akload**. For example:

Example

```
java oracle.apps.ak.akload scott tiger THIN "(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp) (HOST=sfosun) (PORT=1521))
  (CONNECT_DATA=(SID=prod2)))" DOWNLOAD region_cust_US.jlt
GET CUSTOM_REGION NOREGION AK WFNTFWORKLIST LEVEL=SITE """"
```

3. Determine the list of required translations by using the following **SELECT** statement:

```
select language_code, nls_language from fnd_languages
  where installed_flag in ('I', 'B');
```

4. For every language retrieved that needs translations, make a copy of the original base personalization file that **akload** downloaded, for that language. From this point on, the copies of the base personalization files are referred to as the **language jlt files**. For example:

```
cp region_cust_US.jlt region_cust_JA.jlt
```

5. In the **header information** for **each language jlt file**, update the **language.code** for that file. For example:

```
LANGUAGE = "JA"
```

6. In **each language jlt file**, find the location of the personalizations that you would like to translate, by searching for the text **BEGIN CUSTOMIZATION** in the file.
7. There is an entry for every item in the list of personalizations that you have created or updated through the personalization process. Not all items require translation. To find instances of items that require translation in each language jlt file, search for the text **ATTRIBUTE_LABEL_LONG**. These items have the following identifier:

```
BEGIN CUSTOM_REGION_ITEM <attrApplId> <attrCode>
  "ATTRIBUTE_LABEL_LONG"
```

8. For every **CUSTOM_REGION_ITEM** that has the above syntax, change the value of **PROPERTY_VARCHAR2_VALUE_TL** for this attribute to the appropriate translated value.
9. Once you complete the translation of this personalized language jlt file, run the following command to upload this file into the AK personalization translation repository:

```
FNDLOAD <apps_user_name>/<apps_password> 0 Y UPLOAD
  $AK_TOP/patch/115/import/akjlt.lct <trans_jlt_file_name>
- UPLOAD_MODE=NLS WARNINGS=TRUE
```

Translating Personalizations Stored in MDS

If you personalize pages that reside in the MDS Repository, you can perform bulk translations of the personalizations using the following steps, which employ the OA Extension Translation toolset, page 8-8:

1. You must initially create MDS Personalizations in the *base language of your Oracle Applications installation* before you can translate the text portion of the personalizations to other languages. System Administrators who create the personalizations should first set their Language Preference to the base language specified in FND_LANGUAGES.

Note: This requirement is only for OA Framework 11.5.10 and above. In 11.5.9, your personalizations must first be created in English before you can translate the personalizations to another language.

2. Go to the **APPL_TOP** directory and source the file **APPSORA.env** to set all your environment variables. For Windows, refer to the *AD Procedures Guide* to display an appropriate command prompt window and use the resulting command prompt window to set your environment variables (run %APPL_TOP%\envshell.cmd).

Warning: The following MDS utilities require JDK (Java) version 1.3 or higher. If you do not have access to JDK 1.3 please follow the instructions found in MetaLink Note 130091.1 [https://metalink.oracle.com/metalink/plsql/ml2_documents.showDocument?p_database_id=NOT&p_id=130091.1], titled *Upgrading to JDK 1.3 with Oracle Applications 11i*. You should do this before attempting to export/extract and translate personalizations.

3. Determine the location of the personalization document to translate. You can determine the path of the MDS personalization document you wish to export/extract by using the following rules:

- Note the original path to the document you personalized. This is found in the Personalization Context section of the Personalize page, page 5-6 (for example: Document Name: /oracle/apps/fnd/wf/worklist/webui/AdvancWorklistRG).
- Add '/customizations/', the personalization level, and level value that you have chosen in the personalization UI to the path of the document following the 'webui' directory, but before the component name in the document reference path. The personalization levels and level values are defined in the following table:

Level	Level Value
Function	Function Code
Site	0 (zero)
Organization	Organization ID
Responsibility	Responsibility ID
User	User ID

These values combined with the original document reference form the path to the customization document stored in the MDS repository.

Example 1

The Notification Worklist Table has a base document path of: **/oracle/apps/fnd/wf/worklist/webui/AdvancWorklistRG**.

If you created a site level personalization for this document your resulting path to the customization document would be: **/oracle/apps/fnd/wf/worklist/webui/customizations/site/0/AdvancWorklistRG**

Example 2

Suppose you create for the Notification Worklist Table, a function level personalization for the 'High Priority Worklist' custom function. In this case you need to know the function code for the 'High Priority Worklist' function. Let's assume it's OAFHP_WORKLIST. The path to the document would be: **/oracle/apps/fnd/wf/worklist/webui/customizations/function/OAFHP_WORKLIST/AdvancWorklistRG**

You can also use SQL*Plus to review all the personalizations for a given base document. JDR_UTILS is a PL/SQL package that allows you to evaluate the list of personalization documents that are in your MDS repository. Included in this package is a procedure called `jdr_utils.listcustomizations("")`; which allows you to see the personalization document path names that are currently defined in MDS. To run this procedure, launch Sql*Plus, set `serveroutput on`, and execute the `jdr_utils.listcustomizations("")` command. Replace the "" reference with an actual base document reference. For example, to see all the personalization documents for the Notifications Worklist Table, execute the following command:

```
exec jdr_utils.listcustomizations('/oracle/apps/fnd/wf/
worklist/webui/AdvancWorklistRG');
```

If you run the example above, you may notice function personalization document references that you did not create. These are Oracle-seeded function-level personalizations created by Oracle Applications development teams. Personalization definitions are seeded by development teams so that they can share components across products and vary their look and behavior slightly with each use.

For more information about JDR_UTILS, refer to the **Inspecting the MDS Repository Content** section in the Testing and Debugging chapter of the *Oracle Application Framework Developer's Guide*.

4. Determine the list of required translations by using the following SELECT statement:

```
select language_code, nls_language from fnd_languages
where installed_flag in ('I', 'B');
```

For every language retrieved that needs translations, you will need to specify that language for the `-languages` option when you run the XLIFF Extractor in the next step. Specify the languages for the `-languages` option as the language-territory code using the format `xx-YY`, where `xx` is the two character language code, in lower case, and `YY` is the two character territory code in upper case.

Note: If the last two characters of the language-territory code are not in upper case, the XLIFF Importer automatically converts them to upper case and logs a warning.

The following mapping, page 8-14 was used when migrating translations from AK format to OA Extension format, to map the Oracle Applications language code to the xx-YY format.

5. Use the XLIFF Extractor tool, page 8-8 directly against the MDS DB repository to export and extract the base language personalized content into an XLIFF file (.xlf) in one step. (Note that earlier versions of the XLIFF Extractor were only able to extract language content from .xml files and as a result, you had to first export the personalizations into .xml files using the Export tool, then use the XLIFF Extractor tool to extract the base language content from the Xml to an XLIFF file.)

The XLIFF formatted file contains information regarding the translatable personalization properties and is the document that you physically translate into your installed languages. You can send out large volumes of translation content to a 3rd party translator, if necessary, using this mechanism.

Insert into the path of the original document you personalized, the personalization level and level value mapping information determined in Step 3 to derive the *fullPathOfFileName_or_directoryname* for the following command:

```
Java oracle.jrad.tools.trans.extractor.XLIFFExtractor
<fullPathOfFileName_or_directoryname>
  -username <MDS repository username>
  -password <MDS repository password>
  -dbconnection <database connection string in TNSNAMES format>
  -root <root directory, where all the MDS documents (Xml files)
can be found or be saved to>
  -source db
  -languages <List of comma separated target languages
for which the XLIFF documents are to be created.>
  -mmd_dir <directory location of the MMD files to use>
  -xliff_dir <source directory for XLIFF files. This is where you
r XLIFF files
will be created or expected by the XLIFF Importer.>
```

Example

The following example first exports the base language site level personalizations of the Advanced Worklist table document /oracle/apps/fnd/wf/worklist/webui/customizations/site/0/AdvancWorklistRG and saves the content to a Xml document in the root directory \$APPL_TOP/admin/patch. It then extracts the translatable information from the Xml document and saves it to two .xlf files with the target language set for each file as "fr-FR" or "ja-JP". The two .xlf files are saved under two new subdirectories named for the respective languages specified, beneath the directory specified by the -xliff_dir option.

Note: You can place the documents under any root directory you wish using the -root parameter. These examples use \$APPL_TOP/admin/patch for convenience.

```

Java oracle.jrad.tools.trans.extractor.XLIFFExtractor /oracle/apps/
s/
  fnd/wf/worklist/webui/customizations/site/0/AdvancWorklistRG
  -rootdir $APPL_TOP/admin/patch -username APPSNAME
  -password APPSPWD -dbconnection "(DESCRIPTION=(ADDRESS=(PROTOCOL
=tc)
  (HOST=yourhost) (PORT=yourport)) (CONNECT_DATA=(SID=yoursid)))"
  -source db -languages Fr-FR,ja-JP -mmd_dir $OA_HTML/jrad
  -xliiff_dir $APPL_TOP/admin/patch/oracle/apps/fnd/wf/
  worklist/webui/customizations/site/0/

```

The preceding example exports and extracts the translatable information for

```

/oracle/apps/fnd/wf/worklist/webui/
  customizations/site/0/AdvancWorklistRG

```

to

```

$APPL_TOP/admin/patch/oracle/apps/fnd/wf/worklist/webui/
  customizations/site/0/Fr-FR/AdvancWorklistRG.xlf

```

```

$APPL_TOP/admin/patch/oracle/apps/fnd/wf/worklist/webui/
  customizations/site/0/JA-JP/AdvancWorklistRG.xlf

```

Note: The XLIFF files should be UTF-8 encoded so that the translated strings are imported correctly and subsequently used correctly during runtime.

6. From this point on, the copies of the base XLIFF files are referred to as the **language XLIFF files** and you will be updating these language XLIFF files with the appropriate language translations.

An Example Language XLIFF File

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<xliff version="1.0">
  <file datatype="jdr" original="PayTermsLOVRN"
    product-version="$Header: PayTermsLOVRN.xlf 115.1
    2003/02/26 19:58:26 jfrost noship $" source-language=
    "en-US" target-language="Fr-FR" product-name="qp">
    <body>
      <trans-unit id=".oracle.apps.qp.lov.webui.
        PayTermsLOVRN..PTName...prompt" translate="yes"
        maxbytes="4000" maxwidth="26" size-unit="char">
        <source>Payment Terms</source>
        <target>Payment Terms</target>
        <prop-group name="ora_untranslatable">
          <prop prop-type="tagName">messageStyledText
            </prop>
          <prop prop-type="attributeName">prompt</prop>
        </prop-group>
      </trans-unit>
      <trans-unit id=".oracle.apps.qp.lov.webui.
        PayTermsLOVRN..TermId...prompt" translate="yes"
        maxbytes="4000" maxwidth="14" size-unit="char">
        <source>Term Id</source>
        <target>Term Id</target>
        <prop-group name="ora_untranslatable">
          <prop prop-type="tagName">formValue</prop>
          <prop prop-type="attributeName">prompt</prop>
        </prop-group>
      </trans-unit>
    </body>
  </file>
</xliff>
```

7. For each language XLIFF file, there is a `<trans-unit>` element in the XLIFF file for every translatable component extracted from the XML file. The `<source>` element in each `<trans-unit>` contains the translatable string specified in the source language. The `<target>` element represents the same string translated to the target language. Update the string in the `<target>` element for each `<trans-unit>` element you want to translate.
8. Once you complete the translation of each language XLIFF file, use the XLIFF Importer, page 8-13 to import the translations into the MDS repository.

Example

The following example imports the translated XLIFF file

```
$APPL_TOP/admin/patch/oracle/apps/fnd/wf/worklist/webui/
  customizations/site/0/Fr-FR/AdvancWorklistRG.xlf
```

to the MDS repository:

```
Java oracle.jrad.tools.trans.imp.XLIFFImporter
  -username APPSNAME
  -password APPSPWD
  -dbconnection "(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)
    (HOST=yourserver) (PORT=yourport)) (CONNECT_DATA=
    (SID=yoursid)))"
  $APPL_TOP/admin/patch/oracle/apps/fnd/wf/worklist/webui/
  customizations/site/0/Fr-FR/AdvancWorklistRG.xlf
```

OA Extension Translation Toolset

The OA Extension Translation toolset deals with translatable information contained in OA Extension pages using XLIFF, a widely used XML format for transferring and manipulating translatable resources.

Note: The toolset actually uses oraXLIFF, an Oracle dialect of XLIFF defined and used by WPTG (Oracle's Worldwide Product Translation Group).

The Translation toolset consists of the following:

- XLIFF Extractor, page 8-8 - extracts translatable information from MDS applications (held in the filesystem or in a MDS repository) or from MDS personalization (XML) files into well formed oraXLIFF files.
- XLIFF Importer, page 8-13 - imports translated information in an oraXLIFF file into the MDS repository.

XLIFF Extractor

The XLIFF Extractor extracts the translatable information from a given OA Extension document, into an oraXLIFF file. The XLIFF Extractor can:

- Extract the base language information from an OA Extension document or from all documents in a directory present in a file system.
- Extract the base language information from an OA Extension personalization XML document on the file system.
- Extract the base language information from an OA Extension document or personalization in a MDS repository.
- Extract translated language information from an OA Extension document, only from a MDS repository.

The XLIFF Extractor for Oracle Applications behaves according to the following rules:

Oracle Applications-Specific Rules

- Do not extract strings containing at least one underscore and no space.
- The trans-unit ID should start with ".oracle.apps".
- Support the DBDRV command.

General Rules

- Do not extract numeric strings (string does not contain any alpha characters).
- Do not extract null strings.

Important: If the value of `xml:lang` is 'end-US', apply only Oracle Applications-specific rules 2 and 3 and general rule 2.

To use the XLIFF Extractor, ensure that your classpath, path and environment are set up similar to the environment required for applying an AD patch, and call `Java oracle.jrad.tools.trans.extractor.XLIFFExtractor <full_path_of_file_or_directory_name>` with the appropriate parameters.

If you have Oracle 9i JDeveloper OA Extension, you may alternatively use the `xliffextract.bat` script for Windows or the `xliffextract` shell script for Linux that is

packaged with the JDeveloper IDE, located in the `jdevbin\jdev\bin` directory of the JDeveloper install area. The batch file and shell script each set up the classpath, path and environment for you. Just typing `xliffextract` for Windows or `./xliffextract` for Linux at the command prompt without any parameters will give help about its usage. Parameters enclosed in square brackets [] are optional.

Example

```
Java oracle.jrad.tools.trans.extractor.XLIFFExtractor
  <fullPathOfFileName_or_directoryname>
[-username <username>]
[-password <password>]
[-dbconnection <database_connection>]
[-xliff_extension <xliff_extension>]
[-root <MDS_root_dir>]
-mmd_dir <dir_of_MMD_files>
[-DBDRV <APPS_db_drivers>]
[-xliff_dir <XLIFF_output_dir>]
[-rootPackage <rootpkg>]
[-includeSubpackages]
[-product_name <product_name>]
[-category <category>]
[-source (db | file)]
[-languages <target_language(s)>]
```

Note: In addition to the parameter passing style shown above, you may also pass parameters using the style, `parameter=<value>`.

Replace the arguments as indicated in the explanation of the parameters below:

- `<fullPathOfFileName_or_directoryname>` - (Required) Replace with the full path or fully qualified package name of an **OA Extension XML document** to extract translatable information from a single document, or replace with the full path of a directory to extract information from all the documents in the directory. This argument is case-sensitive. For example:
 - To specify a file name on the filesystem:


```
$APPL_TOP/admin/patch/oracle/apps/fnd/wf/worklist/webui/
  customizations/site/0/AdvancWorklistRG.xml
```
 - To specify a package name in the MDS repository:


```
/oracle/apps/fnd/wf/worklist/webui/
  customizations/site/0/AdvancWorklistRG
```
 - To specify a directory name:


```
$APPL_TOP/admin/patch/oracle/apps/fnd/wf/worklist/webui/
  customizations/site/0/
```
- **-username** - (Optional if extracting from the MDS repository) Specify a username for the database to extract from.
- **-password** - (Optional if extracting from the MDS repository) Specify a password for the database to extract from.
- **-dbconnection** - (Optional if extracting from the MDS repository) Specify a database connection for the database to extract from, in TNSNAMES format.
- **-xliff_extension** - (Optional) Specify a file extension for the generated XLIFF files. The default is `xliff`.

- **-root** - (Optional) Specify a MDS root directory from where the xml files can be found or are to be saved to. The default is the current directory. Note that if you extract your documents from an MDS repository, the new documents will overwrite any existing ones in the specified MDS Root directory.
- **-mmd_dir** - (Required) Specify the directory location of the OA Extension MMD files to use. The MMD files define which properties are translatable and enumerate the properties that are part of the translation context (the useful context information to carry out the translation of the translatable property).
- **-DBDRV** - (Optional) Specify the directive for the APPS database drivers. This is added as a comment in the XLIFF file before the root element. The default is " ".
- **-xliff_dir** - (Optional) Specify the source directory for XLIFF files. This is where the XLIFF files are created or expected by the XLIFF Importer. The default is the current directory.
- **-rootPackage** - (Optional) Specify the virtual package after rootdir and the next subdirectory.
- **-includeSubpackages** - (Optional) This parameter flag is only valid when your input for extraction is a directory. If specified, this flag recursively extracts from all xml files in the directory and the subdirectories under it. It is therefore important you only have MDS xml files in the directory hierarchy when using this argument. If this argument is not specified, the XLIFF Extractor just processes the files in the current director.
- **-product_name** - (Optional) Specify the product name for generated XLIFF files.
- **-category** - (Optional) Specify the product category for generated XLIFF files.
- **-source** - (Optional) Specify whether the source for XLIFF extraction is a database (`db`) or `file`. If you specify `db`, XLIFF Extractor will extract translation content for xml from the MDS repository. The default is `file`.
- **-languages** - (Optional) Specify a comma separated list of target languages for which the XLIFF documents are to be created. Specify the languages as the language-territory code using the format `xx-YY`, where `xx` is the two character language code, in lower case, and `YY` is the two character territory code in upper case. The 'target-language' attribute in an XLIFF file is set based on the language specified in the `-languages` parameter. XLIFF Extractor creates multiple copies of the same `.xlf` file with different target languages and stores each in a subdirectory with the target language as its name. XLIFF Extractor also merges the translated attributes from the repository into the generated XLIFF file if database connection parameters are provided. The default is " ".

Using XLIFF Extractor with the `-languages` parameter

If both the XML file from the file system and the XML file from repository are different (XML file from file system contains extra translatable strings), you can use XLIFF Extractor with the `-languages` option. This merges the translated attributes from the repository so that the generated XLIFF file contains both translated and untranslated strings.

XLIFF Extractor does the following when the `-languages` option is specified:

- If there is a language translation for an attribute in the repository, for say `Fr-FR`, but the XML file from the file system no longer has that attribute (meaning that it is a stale trans-unit), the XLIFF Extractor ignores that string.

- If the XML file from the file system contains a translated attribute, but the translation for that attribute is not available in the repository, the XLIFF Extractor extracts the translation for the attribute so that the generated XLIFF file contains the untranslated target string.
- If there is a translation for an attribute in the repository, and the XML from the file system also contains that attribute, the XLIFF Extractor extracts and merges the translation for the attribute so that the generated XLIFF file contains the translated target string.

Attribute "AAA" in XML in file system	Translation for attribute "AAA" in repository	XLIFF file
Not present	Present	Attribute string ignored.
Present	Not present	Contains untranslated "<target>AAA<target>"
Present	Present, translation string is BBB	Contains "<target>BBB<target>"

Using the Export tool with -translation option

If both the XML file from the file system and the XML file from the repository are the same, you can use Export, page 9-4 -translations to export and extract the XLIFF file. The resulting XLIFF file contains only the translated strings. You can use this method to transfer the already translated contents from your repository to the file system, so you can then move the translations from one repository to another. Note that in the degenerate case where translations for all the translatable attributes of a XML file exist in a repository, the XLIFF files created by both "Export -translations" and "xliffExtract -languages" are similar.

Notes:

- If a "package" attribute is specified in the XML file's top level component, that attribute will determine the package to which the document belongs. This takes precedence over the values specified by the -rootPackage and -rootDir options even though rootDir is still mandatory.
- If for some reason (such as duplicate ID's in the input XML files) duplicate trans-unit IDs get created, the XLIFF Extractor displays the possible duplicate trans-unit IDs as errors and exits without saving the XLIFF file.
- XLIFF Extractor does not generate trans-units if the string does not contain any alphabetical character even when other "translation rules" are satisfied. For example, no trans-unit would be generated for "123 456", but a trans-unit would be generated for "123 456 test."
- **XLIFF Extractor requires XDK903/XDK904 to be used. However, XLIFF can be extracted from personalized documents with XDK817 [the version in IAS 2.0.2.9, which the runtime version uses.]**

XLIFF Extractor Examples

To extract the base language content from the FND_MESSAGES_PAGE.xml file in the file system into a new document located in `f:\temp\oracle\apps\fnd\pages\US\FND_MESSAGES_PAGE.xlf` (for Windows) or `/private/oaext/oracle/apps/fnd/pages/US/FND_MESSAGES_PAGE.xlf` (for Linux):

- **For Windows**

```
Java oracle.jrad.tools.trans.extractor.XLIFFExtractor
  f:\temp\oracle\apps\fnd\pages\FND_MESSAGES_PAGE.xml
  mmd_dir=f:\deliver\jdev
  root=f:\temp\oracle DBDRV="my dbdrv command"
  xliff_dir=f:\temp\oracle\apps\fnd\pages\US
  rootPackage=/oracle
```

- **For Linux**

```
Java oracle.jrad.tools.trans.extractor.XLIFFExtractor
  /private/oaext/oracle/apps/fnd/pages/FND_MESSAGES_PAGE.xml
  mmd_dir=/usr/local/jdeveloper/jdev/lib/ext/jrad/config/mmd
  root=/private/oaext DBDRV="my dbdrv command"
  xliff_dir=/private/oaext/oracle/apps/fnd/pages/US
  rootPackage=/oracle
```

To extract the base language content from the **personalized** FND_MESSAGES_PAGE.xml file in the file system into a new document located in f:\temp\oracle\apps\fnd\pages\US\FND_MESSAGES_PAGE.xlf (for Windows) or /private/oaext/oracle/apps/fnd/pages/US/FND_MESSAGES_PAGE.xlf (for Linux):

- **For Windows**

```
Java oracle.jrad.tools.trans.extractor.XLIFFExtractor
  f:\temp\oracle\apps\fnd\customizations\site\0\pages\FND_MESSAGES
  _PAGE.xml
  mmd_dir=f:\deliver\jdev\lib\ext\jrad\config\mmd
  root=f:\temp DBDRV="my dbdrv command" rootPackage=/oracle
  xliff_dir=f:\temp\oracle\apps\fnd\customizations\site\0\pages\US
```

- **For Linux**

```
ava oracle.jrad.tools.trans.extractor.XLIFFExtractor
  /private/oaext/oracle/apps/fnd/customizations/site/0/pages/FND_M
  ESSAGES_PAGE.xml
  mmd_dir=/usr/local/jdeveloper/jdev/lib/ext/jrad/config/mmd
  root=f:\temp DBDRV="my dbdrv command" rootPackage=/oracle
  xliff_dir=/private/oaext/oracle/apps/fnd/customizations/site/0/p
  ages/US
```

To extract translation content for XML from a repository:

- **For Windows**

```
Java oracle.jrad.tools.trans.extractor.XLIFFExtractor \XLIFFExtra
ctTest1
  -root d:\temp -mmd_dir d:\jdev495\jdev\lib\ext\jrad\config\mmd
  -xliff_dir d:\temp\xliff -source db -username jdr -password jdr
  -dbconnection "(description = (address_list = (address =
  (community = tcp.world) (protocol = tcp) (host = mach1.company.com
  )
  (port = 1521))) (connect_data = (sid = db100)))"
```

- **For Linux**

```

Java oracle.jrad.tools.trans.extractor.XLIFFExtractor /XLIFFExtra
ctTest1
  -root /private/oaext -mmd_dir /usr/local/jdeveloper/jdev/lib/ext
/jrad/config/mmd
  -xliff_dir /private/oaext/xliff -source db -username jdr -passwo
rd jdr
  -dbconnection "(description = (address_list = (address =
(community = tcp.world)(protocol = tcp)(host = mach1.company.com
)
(port = 1521))) (connect_data = (Sid = db100)))"

```

When you extract translation content for XML with different target languages, XLIFF Extractor creates multiple copies of the XLIFF file, one for each language. Each copy is saved to the same directory structure, under a subdirectory of the same name as the target language. The following Windows example creates the files c:\temp\Fr-FR\page1.xlf with target-language="Fr-FR" and c:\temp\JA-JP\page1.xlf with target-language="JA-JP".

- **For Windows**

```

Java oracle.jrad.tools.trans.extractor.XLIFFExtractor
c:\temp\oracle\apps\fn\page1.xml -rootdir c:\temp
-xliff_dir c:\temp -languages Fr-FR,JA-JP

```

- **For Linux**

```

Java oracle.jrad.tools.trans.extractor.XLIFFExtractor
/private/oaext/oracle/apps/ak/pages/page1.xml
-rootdir /private/oaext -xliff_dir /private/oaext/output
-languages Fr-FR,JA-JP

```

XLIFF Importer

The XLIFF Importer takes a translated XLIFF file and imports it into the MDS Repository. This effectively deploys the translated OA Extension document. The Importer tool performs the following validations on the XLIFF files it imports:

- Verifies that the XLIFF file has been translated to another language, where the target-language attribute in the top <File> element is different than the source-language attribute.
- Verifies that the target-language attribute is different from the original OA Extension document's base language (to prevent users from bloating the MDS repository).

To use the XLIFF Importer, ensure your classpath, path and environment are set up similar to what is required for applying an AD patch and call Java oracle.jrad.tools.trans.imp.XLIFFImporter <full_path_of_file> with the appropriate parameters.

If you have Oracle 9i JDeveloper OA Extension, you may alternatively use the Windows **xliffimport.bat** file or the **xliffimport** Linux shell script that is packaged with the JDeveloper IDE, located in the jdevbin\jdev\bin directory of the JDeveloper install area. The batch file and shell script each set up the classpath, path and environment for you. Just typing xliffimport without any parameters will give help about its usage.

```

Java oracle.jrad.tools.trans.imp.XLIFFImporter
<full_path_of_file>
-username <username>
-password <password>
-dbconnection <database>

```

The arguments should be replaced as described for the corresponding parameters below:

- *<full_path_of_file>* - (Required) Replace with the full path to an **XLIFF filename** to import. This argument is case-sensitive. For example:

```
$APPL_TOP/admin/patch/oracle/apps/fnd/wf/worklist/webui/  
customizations/site/0/FR/AdvancWorklistRG.xlf
```

Note: While you should keep your XLIFF file on your file system in a hierarchy that mirrors your MDS repository, (for example, the oracle/apps/fnd/pages/FND_MESSAGES_PAGE.xlf file corresponds to the /oracle/apps/fnd/pages/FND_MESSAGES_PAGE OA Extension page), the Importer tool can accept any valid oraXLIFF file from any file system location.

- **-username** - (Required) Specify the username for the database to import to.
- **-password** - (Required) Specify the password for the database to import to.
- **-dbconnection** - (Required) Specify the database connection for the database to import to, in tnsnames format.

Mapping a Language Code to a Language-Territory Code

The following list identifies the language code to language-territory code mapping used when migrating translations from AK format to OA Extension format:

US = en-US
AR = ar-AE
AS = as-IN
BN = bn-IN
PTB = pt-BR
BG = bg-BG
FRC = fr-CA
CA = ca-ES
HR = hr-HR
CS = cs-CZ
DK = da-DK
NL = nl-NL
EG = ar-EG
GB = en-GB
ET = et-EE
SF = fi-FI
F = Fr-FR
D = de-DE
EL = el-GR
GU = gu-IN
IW = he-IL
HI = hi-IN
HU = hu-HU
IS = is-IS
IN = in-ID
I = it-IT
JA = JA-JP
KN = kn-IN
KO = ko-KR
ESA = es-US
LV = lv-LV
LT = lt-LT
MS = ms-MY
ML = ml-IN
MR = mr-IN
ESM = es-MX
N = no-NO
OR = or-IN
PL = pl-PL
PT = pt-PT
PA = pa-IN
RO = ro-RO
RU = ru-RU
ZHS = zh-CN
SK = sk-SK
SL = sl-SI
E = es-ES
S = sv-SE
TA = ta-IN
TE = te-IN
TH = th-TH
ZHT = zh-TW
TR = tr-TR
UK = uk-UA
VN = vi-VN

Known Issues

See a summary of key Personalization Translation issues, page B-2 with suggested workarounds if available.

Related Information

- BLAF UI Guideline(s)
- Javadoc File(s)
- Lessons(s)
- Sample Code

Deploying Personalizations

This chapter covers the following topics:

- Overview
- Deploying Personalizations Using the Functional Administrator Responsibility
- Deploying Personalizations Using the Import/Export Command Line Tools
- Deploying Personalizations in the AK Repository
- Known Issues
- Related Information

Overview

Both admin-level and user-level personalizations may be extracted from one database and loaded into another. This allows you the freedom to create and test personalizations in a test database before deploying the personalizations to a production instance. Starting in OA Framework 11.5.10.2CU, you can use the new Import/Export UI in the Functional Administrator responsibility to deploy your personalizations.

Deploying Personalizations Using the Functional Administrator Responsibility

For personalized pages that have been created in or migrated to Oracle 9i JDeveloper OA Extension, the meta data can either be in the form of XML files on the file system or stored in the MDS (Meta Data Services) repository. By exporting meta data into XML files on the file system, you can easily move those files to another system or simply login to a different environment and import those XML files to a new database instance. The Functional Administrator responsibility provides a simple UI that lets you both export meta data to XML files, and import XML files into a MDS repository.

1. Set the profile option **FND: Personalization Document Root Path** (FND_PERZ_DOC_ROOT_PATH) to a root directory in your file system where your files are to be exported to or imported from.

For more information about the **FND: Personalization Document Root Path** profile option, refer to the **Personalization** section in the OA Framework Profile Options appendix of the *Oracle Application Framework Developer's Guide*.

2. Log in to Oracle Applications under the **Functional Administrator** responsibility.
3. Select the Personalizations tab, then select the Import/Export sub tab.

- The **Personalization Repository** page renders an HGrid that represents the *personalized* contents of the MDS repository. You may expand nodes in the HGrid to search for and select specific personalized regions or pages to export to an XML file. Note that base documents are not shown in this repository. You may also select nodes (packages) from this HGrid to export to an XML file. Note that if you select a node to export, then all packages and documents contained within that selected node are exported.

The focus of the HGrid is initially set to the `/oracle/apps/` node. The root path when the HGrid first renders, is always `/oracle`.

You may expand the HGrid at the initial focus, but note that there are a large number of applications to browse through at this level. Instead, you may want to plan ahead and use the Search region above the HGrid to identify specific applications or personalized pages, regions or packages. The Search region allows you to search by Application Name or Document Path and well as by a range of Last Updated dates. The search results are shown in the HGrid.

Figure 9-1: Personalization Repository page of the Functional Administrator responsibility

The screenshot shows the Oracle Applications Administration interface for the Personalization Repository. The breadcrumb trail includes: ORACLE Applications Administration > Home > Logout > Preferences > Personalize Page > Diagnostics > Security > Core Services > Personalization > Application Catalog > Import/Export > Personalization Repository. The main content area is titled "Personalization Repository" and includes a search section with the following fields:

- Application: [Text Input]
- Document Path: `/oracle/apps/fnd/customizations/responsibility`
- Last Updated: From [Date Picker] To [Date Picker]
- [Go] button

Below the search fields, there is a tip: "TIP Expand nodes to hierarchically browse all administrator level personalizations in the metadata repository. Base documents will not appear in this list. If an action is performed by selecting a package, this action will apply to all packages and documents contained within that package." Action buttons include "Export to File System" and "Delete".

The search results table is as follows:

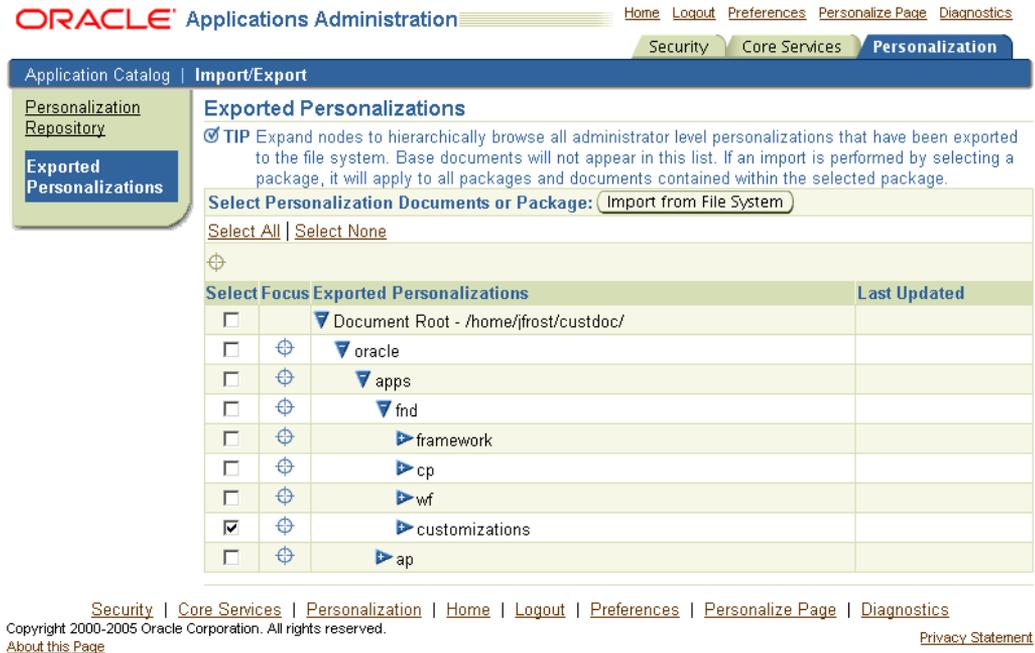
Select Path	Last Updated
<input type="checkbox"/> <code>/oracle/apps/fnd/customizations/responsibility/23349/wf/worklist/webui/AdvancWorklistRG</code>	27-October-2004
<input type="checkbox"/> <code>/oracle/apps/fnd/customizations/responsibility/23350/wf/worklist/webui/AdvancWorklistRG</code>	27-October-2004

At the bottom of the page, there are navigation links: Security | Core Services | Personalization | Home | Logout | Preferences | Personalize Page | Diagnostics. Copyright information: Copyright 2000-2005 Oracle Corporation. All rights reserved. Privacy Statement. About this Page.

- Select the pages, regions or packages you wish to export, then select Export to File System to export the selection to the directory specified by the profile option **FND: Personalization Document Root Path** (FND_PERZ_DOC_ROOT_PATH). You may also select Delete to delete your selection from the repository.
- Once you export your meta data to XML files on the file system, you should login to the other Oracle Applications environment that you want to import these files to, and set the profile option **FND: Personalization Document Root Path** (FND_PERZ_DOC_ROOT_PATH) to the file system location of the exported XML files.

- To import the XML files from your file system into another MDS repository, login to the other Oracle Applications environment as a **Functional Administrator**. Select the Personalizations tab, then select the Import/Export sub tab. Select Exported Personalizations from the side navigation menu to display the **Exported Personalizations** page.

Figure 9-2: Exported Personalizations page of the Functional Administrator responsibility



- The HGrid in the Exported Personalizations page displays all the documents present in the directory specified by the profile option **FND: Personalization Document Root Path** (FND_PERZ_DOC_ROOT_PATH). Select all the documents you wish to import and choose Import from File System.

You may expand nodes in the HGrid to search for and select specific personalized regions or pages to import from the file system. You may also select nodes (packages) from this HGrid to import all packages and documents contained within that selected node.

Deploying Personalizations Using the Import/Export Command Line Tools

Although using the Import/Export UI under the Functional Administrator responsibility to export and import your personalizations is preferred, you may also use the Import/Export command line tools to deploy your personalizations. For personalized pages that have been created in or migrated to Oracle 9i JDeveloper OA Extension, you can use the Export tool to export a personalized region from a repository to an XML file and use the Import tool to import an XML file into a repository.

The Import/Export tools requires JDK version 1.1.8, but can also run with JDK version 1.3. Before running the tools, your classpath, path and environment should be set up similar to the environment required for applying an AD patch.

The meta data for personalized regions are stored in personalization files in the MDS repository, under the following directory structures:

For OA Framework 11.5.57 and prior

```
<prod_top>
|-mds
  |-customizations
  | |-<layer type>
  | | |-<layer value>
  | | | |-<component>
  | | | | |-<webui
  | | | | | + file.xml
  | | | | |-<component>
  | | | | | |-<webui
  | | | | | + file.xml
```

For OA Framework 11.5.10 EAP1 and later

```
<prod_top>
|-mds
  |-<component>
  | |-<webui
  | | | + file.xml
  | | | |-<customizations
  | | | | |-<layer_type>
  | | | | | |-<layer_value>
  | | | | | + file.xml
```

The *<layer_type>* is the set of personalizations belonging to a given personalization level: Function, Verticalization, Localization, Site, Organization, Responsibility or User. The *<layer_value>* is the level value associated with the *<layer_type>*. The levels and corresponding level values are defined in the following table:

Layer Type (Level)	Level Value
Function	<i>Function Name</i>
Verticalization	<i>Industry ID</i>
Localization	<i>Location Code</i>
Site	0 (zero)
Organization	<i>Organization ID</i>
Responsibility	<i>Responsibility ID</i>
User	<i>User ID</i>

Export Tool

The Export tool allows you to export a package or xml file (along with translation information) from the MDS repository of a database instance to a .xml file (or .xlf file for translations). The command line Export tool is necessary if you wish to perform bulk translations of personalization documents, page 8-3.

To use the Export tool, ensure that your classpath, path and environment are set up similar to the environment required for applying an AD patch, and call `java oracle.jrad.tools.xml.exporter.XMLExporter <Package_or_Document_Name>` with the appropriate parameters.

If you have Oracle 9i JDeveloper OA Extension, you may alternatively use the **export.bat** file or the **export** shell script that is packaged with the JDeveloper IDE, located in the `jdevbin\jdev\bin` directory of the JDeveloper install area. The batch file and shell script each set up the classpath, path and environment for you. Just typing `export` without any parameters will give help about its usage.

Usage of the Export tool is as follows:

```
java oracle.jrad.tools.xml.exporter.XMLExporter
<Package_or_Document_Name> -rootdir <output_dir> -username <username>
-password <password> -dbconnection <database> [-mmdir <MMD_dir>]
[-includeSubpackages]
[-displayOnly]
[-jdk13]
[-validate]
[-translations]
[-language <language>]
[-dbdrvFile <dbdrv_file>]
```

Note: There are two styles of exporting based on the version of JDK you are using. One style requires a minimum of JDK 1.1.8. The other style requires JDK 1.3 and provides additional convenience options, described below, that are not available with JDK 1.1.8. If you specify `-jdk13` as an argument in the command line, the tool uses the JDK 1.3 style of importing.

The arguments should be replaced as follows:

- `<Package_or_Document_Name>` - (Required) Replace with an **OA Extension package name or file name**. You can export all the OA Extension xml files in a package (even if they were imported separately) or export a specific OA Extension file. This argument points to the relevant package or file using the OA Extension syntax, which is case-sensitive when interpreting the OA Extension ID.
- `<output_dir>` - (Required) Output directory where the exported xml file structure is to be stored. You may set this to any directory, however, we recommend that you export your packages or XML files to `$APPL_TOP/personalizations`.

If you run the export tool for the package `/oracle/apps/ak/dem/webui/customizations/site/0/REQORDERSTATUSPAGE` and specify `-rootdir $APPL_TOP/personalizations`, the xml file is saved as `$APPL_TOP/personalizations/oracle/apps/ak/dem/webui/customizations/site/0/REQORDERSTATUSPAGE.xml`.

Note: The command line Export tool does not consider the value of the **FND: Personalization Document Root Path** (FND_PERZ_DOC_ROOT_PATH) profile option to determine its output directory.

- `<username>` - (Required) Username for the database to export from.
- `<password>` - (Required) Password for the database to export from.

- `<database>` - (Required) Database connection for the database to export from, in tnsnames format.
- `<MMD_dir>` - (Required) This argument is available only when you run the JDK version 1.3 style of the Export tool. Use this argument to specify the directory location of the OA Extension MMD files. (OAElementList.xml, JRADElementList.xml, UIXElementList.xml).
- **-includeSubpackages** - (Optional) This argument is available only when you run JDK version 1.3 and applies only when you specify a package name to export. If you include this argument, then all the documents in the subdirectories of the package directory specified are exported. For example, consider the following directory structure:

```

oracle
|-apps
  |-icx
    |-regions
      + region1.xml
  |-ak
    |-regions
      + region2.xml

```

If you run the export tool for the package oracle\apps with the argument **-includeSubpackages**, both region1 and region2 are exported.

- **-displayOnly** - (Optional) This argument is available only when you run the JDK version 1.3 style of the Export tool. Include this argument to just display the list of documents to Export. The documents themselves are not actually exported from the repository.
- **-jdk13** - (Optional) Include this argument to run the JDK 1.3 style of the Export tool. The JDK 1.3 style of exporting supports the **-withRefs**, **-includeSubpackages**, **-mmdir**, **-displayOnly** and **-validate** options, whereas the JDK 1.1.8 style does not.
- **-validate** - (Required) This argument is available only when you run the JDK version 1.3 style of the Export tool. Include this argument to validate the OA Extension files before exporting from the repository. The Export tool displays warning messages for any validation issues in the files, but the files are still exported from the repository even if there are validation warning messages.
- **-translations** - (Optional) If this argument is specified, the Export tool exports from the repository, the translations, if any, for the specified XML documents and does not export the XML documents themselves. The translations (XLIFF documents) are exported to the appropriate language subdirectory as .xlf files, under the output directory specified by **-rootdir**. If the **-translations** argument is not specified, the Export tool exports only the specified MDS XML files from the repository.
- `<language>` - (Optional) Language for which translations should be exported. The **-language** argument is valid only when the **-translations** argument is specified. If the **-language** argument is not specified but the **-translations** argument is, then translations for all languages are exported.
- `<dbdrv_file>` - (Optional) File that contains the DBDRV command to insert into the exported XML document. A template file containing a DBDRV command is available at `..\jdev\lib\ext\jrad\config\TemplateAppsJRADCCustomizationFile.xml`.

Warning: The **-dbdrvFile** option should be used only by Oracle's in-house E-Business Suite developers.

Example Export Tool Usage

The following example exports the XML for the document /oracle/apps/fnd/dem/hello/webui/HelloWorldPG from the repository to the file system directory \$APPL_TOP/personalizations, and inserts a DBDRV command into the exported XML document, using the JDK 1.1.8 style of the export tool (typical Apps ARU/ DBDRV use case):

```
java oracle.jrad.tools.xml.exporter.XMLExporter
/oracle/apps/fnd/dem/hello/webui/HelloWorldPG
-rootdir $APPL_TOP/personalizations
-username user1
-password testing
-dbconnection "(description = (address_list = (address =
  (community = tcp.world)(protocol = tcp)
  (host =machine1.oracle.com)(port = 1521)))
  (connect_data = (sid = mach1)))"
-dbdrvFile d:\jdev\lib\ext\jrad\config\
  TemplateAppsJRADCustomizationFile.xml
```

The following example exports the French translation for the document /oracle/apps/fnd/dem/hello/webui/HelloWorldPG from the repository to \$APPL_TOP/personalizations/fr_FR/oracle/apps/fnd/dem/hello/webui/HelloWorldPG.xlf, using the JDK 1.3 style of the export tool:

```
java oracle.jrad.tools.xml.exporter.XMLExporter
/oracle/apps/fnd/dem/hello/webui/HelloWorldPG
-rootdir $APPL_TOP/personalizations
-username user1
-password testing
-dbconnection "(description = (address_list = (address =
  (community = tcp.world)(protocol = tcp)
  (host =machine1.oracle.com)(port = 1521)))
  (connect_data = (sid = mach1)))"
-mmddir d:\jdeveloper\jdev\myhtml\oa_html\jrad
-jdk13
-translations
-language fr-FR
```

Exporting Personalizations

If for any reason you cannot use the Personalization Repository page, page 9-2 of the Functional Administrator responsibility to export personalizations, you may use the command line Export tool.

Step 1 - Determine the Path

To export a personalization, you must first determine the path to the document you personalized. You can determine the path of the MDS personalization document you wish to export by using the following rules:

- Note the original path to the document you personalized. This is found in the Personalize page, page 5-6 of the Personalization UI. (For example: Document Name: /oracle/apps/fnd/wf/worklist/webui/AdvancWorklistRG)
- Add '/customizations/', the personalization level, and level value that you have chosen in the personalization UI to the path of the document following the webui directory, but before the component name in the document reference path. The personalization levels and level values are defined in the following table:

Level	Level Value
Function	<i>Function Name</i>
Verticalization	<i>Industry ID</i>
Localization	<i>Location Code</i>
Site	0 (zero)
Organization	<i>Organization ID</i>
Responsibility	<i>Responsibility ID</i>
User	<i>User ID</i>

These values combined with the original document reference form the path to the customization document stored in the MDS repository.

Example 1

The Notification Worklist Table has a base document path of:

```
/oracle/apps/fnd/wf/worklist/webui/AdvancWorklistRG.
```

If you created a site level personalization for this document your resulting path to the customization document would be:

```
/oracle/apps/fnd/wf/worklist/webui/customizations/site/0/AdvancWorklistRG
```

Example 2

Suppose you create a function level personalization for the 'High Priority Worklist' custom function. In this case you need to know the function code for the 'High Priority Worklist' function. Let's assume it's OAFHP_WORKLIST. The path to the document would be:

```
/oracle/apps/fnd/wf/worklist/webui/customizations/function/OAFHP_WORKLIST/AdvancWorklistRG
```

You can also use SQL*Plus to review all the personalizations for a given base document. JDR_UTILS is a PL/SQL package that allows you to evaluate the list of personalization documents that are in your MDS repository. Included in this package is a procedure called `jdr_utils.listcustomizations("");` which allows you to see the personalization document path names that are currently defined in MDS. To run this procedure, launch SQL*Plus, set `serveroutput on`, and execute the `jdr_utils.listcustomizations("");` command. Replace the `"` reference with an actual base document reference. For example, to see all the personalization documents for the Notifications Worklist Table, execute the following command:

```
exec jdr_utils.listcustomizations('/oracle/apps/fnd/wf/worklist/webui/AdvancWorklistRG');
```

If you run the example above, you may notice function personalization document references that you did not create. These are Oracle-seeded function-level personalizations created by Oracle Applications development teams. Personalization definitions are seeded by development teams so that they can share components across products and vary their look and behavior slightly with each use.

For more information about JDR_UTILS, refer to the **Inspecting the MDS Repository Content** section in the Testing and Debugging chapter of the *Oracle Application Framework Developer's Guide*.

Step 2 - Export

Use the Export tool to export the base language personalized page from the MDS repository to an XML file on the file system. Insert into the path of the original document you personalized, the personalization level and level value mapping information determined in the previous step to derive the personalization document name.

Example of Exporting Personalizations

Export the site level personalizations made to the Advanced Worklist table document `/oracle/apps/fnd/wf/worklist/webui/AdvancWorklistRG` to `$APPL_TOP/personalizations/oracle/apps/fnd/wf/worklist/webui/customizations/site/0/AdvancWorklistRG.xml`.

Note: You can place the documents under any root directory you wish using the `-rootdir` parameter. The following example uses `$APPL_TOP/personalizations` as a recommendation.

```
java oracle.jrad.tools.xml.exporter.XMLExporter
/oracle/apps/fnd/wf/worklist/webui/customizations/site/0/
AdvancWorklistRG
-rootdir $APPL_TOP/personalizations
-username APPSNAME
-password APPSPWD
-dbconnection "(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
(HOST=yourhost)(PORT=yourport))(CONNECT_DATA=
(SID=yoursid)))"
```

Import Tool

If for whatever reason, you cannot use the Exported Personalizations page, page 9-3 of the Functional Administrator responsibility to import personalizations, you may use the command line Import tool. The Import tool allows you to import an xml file or package directory into the MDS repository of a database instance.

To use the Import tool, ensure your classpath, path and environment is set up similar to the environment required for applying an AD patch and call `java oracle.jrad.tools.xml.importer.XMLImporter <Package_or_Document_Name>` with the appropriate parameters.

If you have Oracle 9i JDeveloper OA Extension, you may alternatively use the **import.bat** file or the **import** shell script that is packaged with the JDeveloper IDE, located in the `jdevbin\jdev\bin` directory of the JDeveloper install area. The batch file and shell script each set up the classpath, path and environment for you. Just typing `import` without any parameters will give help about its usage.

Usage of the Import tool is as follows:

```
java oracle.jrad.tools.xml.importer.XMLImporter
<full_path_of_file_or_directory_to_import> -username <username>
-password <password> -dbconnection <database> [-userId <userID>]
-rootdir <root_dir> [-rootPackage <root_pkg_dir>]
[-validate] [-includeSubpackages] [-jdk13]
[-mmddir <MMD_dir>]
[-displayOnly]
```

Note: There are two styles of importing based on the version of JDK you are using. One style requires a minimum of JDK 1.1.8, which is

how the ARU/DBDRV commands use the import tool. The other style requires JDK 1.3 and provides additional convenience options, described below, that are not available with JDK 1.1.8. If you specify **-jdk13** as an argument in the command line, the tool uses the JDK 1.3 style of importing.

The arguments should be replaced as indicated following:

- `<full_path_of_file_or_directory_to_import>` - (Required) Replace with the full path to an **OA Extension package name** to import all the xml files in a package directory, or replace with the **full path of a file name** to import a specific xml file. This argument is case-sensitive.

Note: Although you can import XML files of your customizations from any location, we recommend that you copy the file or package directory you want to import to \$APPL_TOP/personalizations and import from this staging area. For example, to import the file `<JDEV_USER_HOME>/myprojects/mycompany/oracle/apps/fnd/dem/hello/webui/HelloWorldPG.xml`, you would first copy it to the staging area \$APPL_TOP/personalizations, so that the full path of the file you now specify is \$APPL_TOP/personalizations/mycompany/oracle/apps/fnd/dem/hello/webui/HelloWorldPG.xml. To import all the files in the package, you specify \$APPL_TOP/personalizations/mycompany/oracle/apps/fnd/dem/hello/webui/.

Note: If you wish to import a custom XML page, we recommend that you stage the file in \$APPL_TOP/<CompanyIdentifier>/<Custom ProductShortName>/<product-version>/mds/. Refer to "Deploying Customer Extensions" in the *OA Framework Developer's Guide* for additional information.

Note: JDK 1.1.8 only supports importing one file at a time.

- `<username>` - (Required) Username for the database to import to.
- `<password>` - (Required) Password for the database to import to.
- `<database>` - (Required) Database connection for the database to import to, in tnsnames format.
- `<userID>` - (Optional) User ID used to set the created_by or last_updated_by columns in the repository tables
- `<root_dir>` - (Required) Root directory from where the xml files are loaded.

This should be the directory where the OA Extension package structure resides. If you follow our staging area recommendation for personalizations, this is: \$APPL_TOP/personalizations.

Note: The command line Import tool does not consider the value of the **FND: Personalization Document Root Path** (FND_PERZ_DOC_ROOT_PATH) profile option to determine the root directory from where the xml files are loaded.

- `<root_pkg_dir>` - (Optional) Top level directory under rootdir to which the OA Extension package belongs. For example, if under rootdir, you have a "fnd" directory, and the xml files belong to the "/oracle/apps/fnd" package, you would set the rootPackage argument as `-rootPackage /oracle/apps`. Note that this parameter has to start with "/".

Note: In the JDK version 1.1.8 style of the Import tool, if a "package" attribute is specified in an xml file's top level component, that "package" attribute takes precedence over the rootPackage and rootDir arguments and determines the package the document is imported into.

- **-validate** - (Required) This argument is available only when you run the JDK version 1.3 style of the Import tool. Include this argument to validate the OA Extension files before importing into the repository. The Import tool displays warning messages for any validation issues in the files, but the files are still imported into the repository even if there are validation warning messages.
- **-includeSubpackages** - (Optional) This argument is available only when you run the JDK version 1.3 style of the Import tool and only if you are importing a package directory. Include this argument to import all the xml files located in the subdirectories of the package directory specified. It is important you only have MDS xml files in the directory hierarchy when using this argument. For example, consider the following directory structure:

```
oracle
|-apps
  |-icx
  | |-regions
  |   + region1.xml
  |-ak
  | |-regions
  |   + region2.xml
```

If you run the import tool for the package `oracle\apps` with the argument `-includeSubpackages`, both region1 and region2 are imported.

- **-jdk13** - (Optional) Include this argument to run the JDK 1.3 style of the Import tool. The JDK 1.3 style of importing supports the `-loadRefs`, `-includeSubpackages`, `-mmddir`, `-displayOnly` and `-validate` options, whereas the JDK 1.1.8 style does not.
- `<MMD_dir>` - (Required) This argument is available only when you run the JDK version 1.3 style of the Import tool. Use this argument to specify the directory location of the OA Extension MMD files. (OAELEMENTLIST.xml, JRADELEMENTLIST.xml, UIXELEMENTLIST.xml).
- **-displayOnly** - (Optional) This argument is available only when you run the JDK version 1.3 style of the Import tool. Include this argument to just display the list of documents to import. The documents themselves are not actually imported into the repository.

Example Usage

The following example imports the XML file `<JDEV_USER_HOME>/myprojects/mycompany/oracle/apps/fnd/dem/hello/webui/HelloWorldPG.xml` (after copying it to `$APPL_TOP/personalizations`), using the JDK 1.1.8 style of the import tool (typical Apps ARU/ DBDRV use case):

```

java oracle.jrad.tools.xml.importer.XMLImporter
  $APPL_TOP/personalizations/mycompany/oracle/apps/fnd/dem/hello/
  webui/HelloWorldPG.xml
  -username user1
  -password testing
  -rootdir $APPL_TOP/personalizations
  -rootPackage /mycompany/oracle/apps/
  -dbconnection "(description = (address_list = (address =
    (community = tcp.world) (protocol = tcp)
    (host =machine2.us.oracle.com) (port = 1521)))
    (connect_data = (sid = mach2)))"

```

The following example imports all of the XML files for the package specified by the directory `<JDEV_USER_HOME>/myprojects/mycompany/oracle/apps/fnd/dem/hello/webui/`, using the JDK 1.3 style of the import tool. Import all files in subpackages as well and validate the files to import.

```

java oracle.jrad.tools.xml.importer.XMLImporter
  $APPL_TOP/personalizations/mycompany/oracle/apps/fnd/dem/hello/
  webui/
  -jdk13
  -mmdir "d:\deliver\jdev\mywork\config\jrad"
  -includeSubpackages
  -username user1
  -password testing
  -rootdir $APPL_TOP/personalizations
  -rootPackage /mycompany/oracle/apps/
  -validate
  -dbconnection "(description = (address_list = (address =
    (community = tcp.world) (protocol = tcp)
    (host =machine2.oracle.com) (port = 1521)))
    (connect_data = (sid = mach2)))"

```

The following example imports the site level personalized document, `$APPL_TOP/personalizations/oracle/apps/fnd/wf/worklist/webui/customizations/site/0/AdvancWorklistRG.xml`, into the MDS repository:

Example

```

java oracle.jrad.tools.trans.imp.XLIFFImporter
  $APPL_TOP/personalizations/oracle/apps/fnd/wf/worklist/webui/cus
  tomizations/site/0/AdvancWorklistRG.xml
  -username APPSNAME
  -password APPSPWD
  -dbconnection "(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)
    (HOST=yourserver) (PORT=yourport)) (CONNECT_DATA=
    (SID=yoursid)))"

```

Deploying Personalizations in the AK Repository

If the personalized pages still reside in the AK repository and have not migrated to OA Extension, the executable you use to perform this function is `akload`, a Java program that lets you download components defined in the AK Runtime Dictionary into an XML-like file format called a `.jlt` file. These `.jlt` files can then be uploaded to other databases, again using the `akload` Java program.

Requirements for `akload` in Release 11i are as follows: JDK (version 1.1.8 or above) and Oracle JDBC (version 8.0.6.0.0 or above). If using `Parameters.class` version 115.12 or above, no Oracle client software needs to be installed on the PC. Otherwise, if

running from a PC, Oracle8 Client must be installed on the PC. Your classpath, path and environment should be set up similar to the environment required for applying the latest AD patch.

AKLOAD

The syntax for using akload to download or upload personalization data from AK Runtime Dictionary in a Oracle database is:

```
Java oracle.apps.ak.akload  
<P1><P2><P3><P4><P5><P6><P7><P8><P9><P10><P11>...<Pn-1><Pn> <P8a>  
<P8ai>
```

The parameters *P1* through *P8* should be replaced as follows:

Note: Replace *italicized* text with a value as indicated.

- **P1** - (Required) *Username*
- **P2** - (Required) *Password*
- **P3** - (Required) *JDBC Driver Type* = THIN
- **P4** - (Required) *Database Address*
- **P5** - (Required) *Loader Option* =
 - DOWNLOAD - Downloads AK data from an Oracle database into a flat file.
 - UPLOAD - Uploads AK data from a flat file back into an Oracle database.
- **P6** - (Required) *File Name* = Name of flat file to be downloaded or uploaded. For example: C:\data\x.jlt (PC) or /home/myhome/x.jlt (UNIX)
- **P7** - (Required) *Get/Update Option* =
 - UPDATE - Valid only when P5 = UPLOAD. Instructs AKLOAD to update any existing records in the database, with values from the flat file.
 - NCUPDATE - Valid only when P5 = UPLOAD. Instructs AKLOAD to only update existing records which have not been updated by customers in the database, with values from the flat file.
 - NOUPDATE - Valid only when P5 = UPLOAD. Instructs AKLOAD to not update any existing records in the database. Thus, the flat file values will be discarded whenever AKLOAD encounters an existing record in the database.
 - GET - Valid only when P5 = DOWNLOAD. Downloads a specific business object from an Oracle database into a flat file based on the values specified by parameters P6 through P8.
- **P8** - One of the following:
 - *Business Object* (Required) = CUSTOM_REGION. Business object to be downloaded from an Oracle database into a flat file. Valid only when P5 = DOWNLOAD.

Note that when you download personalizations, akload returns all of the related:

- Regions
- Region Items
- Region_LOV_Relations

- Category_Usages
- Attributes
- Personalization
- Custom_Region
- Custom_Region_Item
- Criteria
- *NLS_LANG* (Optional;) Specify the **value of \$NLS_LANG** if uploading a .jlt file which is in a language that is different from the OS language. For example: AMERICAN_AMERICA.WE8ISO8859P1. Valid only when P5 = UPLOAD.
- **P9** - (Optional) *Download With or Without Attributes Option* =
 - WATTRIBUTE - Valid only when P5 = DOWNLOAD. Downloads the attributes with the specified business object. This is the default behavior if you don't set this parameter.
 - NOATTRIBUTE - Valid only when P5 = DOWNLOAD. Attributes are not downloaded with the specified business object.
- **P10 and P11** - (Required) *Application Short Name and Business Object's Key*. Valid only when P5 = DOWNLOAD and P7 = GET.

Important: P10 and P11 must be specified together as a pair and may be repeated as many times as the operating system allows. You must have at least one *Application Short Name and Business Object's key* value pair. If you do not want to specify a *Business Object's key*, enter "" as a place holder.

...

- **Pn-1 and Pn** - (Optional) *Application Short Name and Business Object's Key*. Valid only when P5 = DOWNLOAD and P7 = GET.
- **P8a** - (Optional - Valid only when P8 = CUSTOM_REGION) *Administrator Personalization Level* =
 - LEVEL=SITE
 - LEVEL=ORGANIZATION
 - LEVEL=RESPONSIBILITY
 - LEVEL=FUNCTION
- **P8ai** - (Required only if P8a is specified) *Primary Key Value of Personalization Level Entity* =
 - "" - if P8a = LEVEL=SITE
 - **Value of ORGANIZATION_CODE** - if P8a = LEVEL=ORGANIZATION
 - **Value of RESPONSIBILITY_KEY** - if P8a = LEVEL=RESPONSIBILITY
 - **Value of FUNCTION_CODE** - if P8a = LEVEL=FUNCTION

Note: Use "" around your input string whenever it contains an embedded space.

Examples

Example 1

If you wish to download personalization data for region REQORDERSTATUS in AK to a file called test.jlt, your syntax might look as follows:

```
Java oracle.apps.ak.akload scott tiger THIN "(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=sfosun) (PORT=1521))(CONNECT_DATA=(SID=prod2)))" DOWNLOAD test.jlt GET CUSTOM_REGION AK REQORDERSTATUS
```

Example 2

If you wish to upload data from the flat file test.jlt, and have akload update any existing records with values from the flat file, your akload syntax might look as follows:

```
Java oracle.apps.ak.akload scott tiger THIN "(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=jfksun) (PORT=1521))(CONNECT_DATA=(SID=prod3)))" UPLOAD test.jlt UPDATE
```

Example 3

If you wish to download personalization data at the Organization level for region REQORDERSTATUS in AK to a file called test.jlt, your syntax might look as follows:

```
Java oracle.apps.ak.akload scott tiger THIN "(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=sfosun) (PORT=1521))(CONNECT_DATA=(SID=prod2)))" DOWNLOAD test.jlt GET CUSTOM_REGION AK REQORDERSTATUS LEVEL=ORGANIZATION ""
```

Example 4

If you wish to download personalization data at the Responsibility level with Responsibility key = REQUISITION_DEMO for region REQORDERSTATUS in AK to a file called test.jlt, your syntax might look as follows:

```
Java oracle.apps.ak.akload scott tiger THIN "(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=sfosun) (PORT=1521))(CONNECT_DATA=(SID=prod2)))" DOWNLOAD test.jlt GET CUSTOM_REGION AK REQORDERSTATUS LEVEL=RESPONSIBILITY REQUISITION_DEMO
```

Known Issues

See a summary of key Personalization issues, page B-1 with suggested workarounds if available.

Related Information

- BLAF UI Guideline(s)
- Javadoc Files
- Lesson(s)
- Sample Code

Migrating AK Personalizations

This chapter covers the following topics:

- Overview
- Oracle-Seeded User-Level Personalizations
- Migrating Customer Personalizations
- Known Issues
- Related Information

Overview

The technology of OA Framework consists of several components. In OA Framework 11.5.56 and earlier, one of those components is AK, an interface and repository that captures and stores the meta data that defines the HTML pages of Oracle Applications. Specifically, the AK repository consists of the following tables: AK_REGIONS, AK_REGION_ITEMS, AK_ATTRIBUTES and AK_REGION_LOV_RELATIONS. When you make personalizations to these pages, the personalization meta data is stored in separate shadow tables called AK_CUSTOMIZATIONS, AK_CUSTOM_REGIONS, and AK_CUSTOM_REGION_ITEMS.

For OA Framework 11.5.57 and above, all the meta data stored previously in AK is rehosted in Oracle 9i JDeveloper OA Extension. In OA Extension, the application meta data is stored either in XML files, in a format defined by MDS (Meta Data Services) Schemas, or in the MDS repository tables.

When you apply the patch for OA Framework 11.5.57, the ARU (Automated Release Update) migrates the OA Framework-based pages from the AK repository to the MDS repository. The patch also migrates and loads any "Oracle-seeded user-level" personalizations from AK to the MDS repository. If you or your users have created personalizations in the past and your AK repository and MDS repository are in the same database instance, the patch also migrates your "customer personalizations" for you. You need only run the Personalization Migration tool to migrate customer personalizations if your AK and MDS repositories are in different database instances. To summarize:

- OA Framework 11.5.57 patch automatically migrates from AK to MDS:
 - OA Framework base pages.
 - "Oracle-seeded user-level" personalizations.

- Customer personalizations, if AK and MDS are both in the same database instance.
- OA Framework 11.5.57 patch does *not* migrate customer personalizations if the AK and MDS repositories are in separate database instances. You need to run the Personalization Migration tool manually to migrate your customer personalizations in this case.

The Personalization Migration tool requires JDK version 1.1.8, although it can also run with JDK version 1.3. Before running the Personalization Migration tool, your classpath, path and environment should be set up similar to the environment required for applying the patch.

Oracle-Seeded User-Level Personalizations

After you apply the OA Framework 11.5.57 patch, you should check your log files to ensure that the migration of the "Oracle-seeded user-level" personalizations are successful. If you encounter any problems, rerun the patch. If you continue to encounter problems after running the patch, contact Oracle Support services for assistance.

Migrating Customer Personalizations

If you have created what are known as customer personalizations for your site or installation, and your AK and MDS repositories are in the same database instance, the OA Framework 11.5.57 patch automatically migrates your customer personalizations for you. If your AK repository is in a different database instance than your MDS repository, you must run the Personalization Migration Tool yourself to migrate these personalizations from the AK to the MDS repository.

To use CustMigrationTool to migrate customer personalizations from the AK repository in one instance to the MDS repository in another instance, ensure your classpath, path and environment is set up similar to the environment required for applying an AD patch and call `java -mx256m oracle.jrad.migration.cust.CustMigrationTool <ApplicationShortName> <RegionCode>` with the appropriate parameters.

If you have Oracle 9i JDeveloper OA Extension, you may alternatively use the **migratecust.bat** file that is packaged with the JDeveloper IDE, located in the `jdevbin\jdev\bin` directory of the JDeveloper install area. The batch file sets up the classpath, path and environment for you. Just typing `migratecust` without any parameters will give help about its usage.

The syntax for running CustMigrationTool is as follows:

Example

```
java -mx256m oracle.jrad.migration.cust.CustMigrationTool
<ApplicationShortName> <RegionCode>
-customer_cust
-username <username>
-password <password>
-dbconnection <database>
-migrationdir <migration_dir>
[-jrad_username <mds_username>]
[-jrad_password <mds_password>]
[-jrad_dbconnection <mds_database>]
[-jrad_userid <mds_userid>]
[-rootPackage <root_pkg>]
```

Replace *<ApplicationShortName>* with the **application short name of the personalized pages** you wish to migrate. Replace *<RegionCode>* with the **region code of the page you wish to migrate**. Specifying a percent sign (%) for the region code allows you to migrate personalizations for all the pages of the application short name specified, in one run. Be sure to specify the parameter *-customer_cust* when you migrate customer personalizations. The parameters enclosed in square brackets [] are optional, based on the assumptions described in the parameter value descriptions for P1 to P9 as follows.

- *<username>* - (Required) Username for the AK database.
- *<password>* - (Required) Password for the AK database.
- *<database>* - (Required) Database connection details for the **AK** repository database in TNS format.

If you are migrating customer personalizations from a test instance to a production instance, this parameter should be used to specify the information for the test instance. Example: " (DESCRIPTION= (ADDRESS_LIST= (ADDRESS = (PROTOCOL = TCP) (HOST = appdb.company.com) (PORT = 1521))) (CONNECT_DATA = (SID = testdb))) "

- *<migration_dir>* - (Required) Directory where the migrated personalization meta data is output as xml files.

This directory should be the same as the output directory of the migration of base HTML pages from AK to MDS. The default value is ".".

- *<mds_username>* - Username for the **MDS** database.
- *<mds_password>* - Password for the **MDS** database.
- *<mds_database>* - Database connection details for the **MDS** database in TNS format.

The syntax is similar to that described for P4, the DBconnection parameter. Note that if parameters P5, P6, and P7 are not specified, then the values for P1, P2, and P3 are used for the MDS repository and the Personalization Migration Tool assumes that the MDS meta data is stored in the same database instance as the AK meta data.

- *<mds_userid>* - The user ID used by the tool to set the *created_by* and *last_updated_by* columns in the MDS repository tables.
- *<root_pkg>* - Root package.

Set this to *"/oracle/apps"* when the directory specified for the *migrationdir* parameter contains */oracle/apps* in the directory structure.

Example

If you wish to migrate your customer personalizations of Oracle HRMS pages from the AK repository in your test database instance to the MDS repository in your production database instance, your syntax would look as follows:

```
java -mx256m oracle.jrad.mmigration.cust.CustMigrationTool
  PER %
-customer_cust
-username apps
-password apps
-dbconnection "(DESCRIPTION=(ADDRESS_LIST =(ADDRESS=
  (PROTOCOL=tcp)(HOST=sfosun) (PORT=1521)))
  (CONNECT_DATA=(SID=test2)))"
-migrationdir D:\dev\jsmith\jdev\myprojects
-jrad_username apps
-jrad_password apps
-jrad_dbconnection "(DESCRIPTION=(ADDRESS_LIST =(ADDRESS=
  (PROTOCOL=tcp)(HOST=laxsun) (PORT=1521)))
  (CONNECT_DATA=(SID=prod2)))"
-jrad_userid jsmith
-root_package "/oracle/apps"
```

Known Issues

See a summary of key Personalization Migration issues, page B-1 with suggested workarounds if available.

Related Information

- [BLAF UI Guideline\(s\)](#)
- [Javadoc File\(s\)](#)
- [Lessons\(s\)](#)
- [Sample Code](#)

Personalization Considerations

This appendix covers the following topics:

- Overview
- Attachments
- Auto-Repeating Layout
- Bound Values
- Branding
- Buttons (Global)
- Charts and Graphs
- Concurrent Processing
- Configurable Pages
- Custom HTML
- Date Picker
- Declarative Page Flow
- Dialog Pages
- Dynamic User Interface
- Flexfields
- Forms / OA Framework Page Integration
- formValue
- HGrid
- Hide/Show
- Images in Your Pages
- Locator Element: Breadcrumbs
- Mobile Applications
- Page Contents Bottom Line
- Page Footer
- Printable Page
- Processing Page

- Rich Text Editor
- Save Model (Warn About Changes)
- Shuttle
- Standard Web Widgets
- SubTab Navigation
- Switchers (Application, Context and Table Content)
- Tables - Advanced
- Tables - Classic
- Tree
- Workflow Worklist

Overview

This appendix lists the personalization considerations, if any, for the various types of elements (regions or items) on a page.

Warning: This document does not list all personalizable properties for a page region or item. Instead, it lists key considerations that we hope will facilitate your personalization efforts. Please refer to the *OA Component Reference*, available on Metalink, for a complete list of properties that may be personalized for a region or item.

Attachments

Please note the following restrictions on personalizing attachments-enabled pages:

- Personalization of the actual Attachments table region that renders on a page is not yet supported, with the exception that administrators can reorder the columns within the table.
- Administrators can personalize the children (columns) of the Attachments table, with the exception of the **File Name**, **Update** and **Delete** columns.
- Administrators can also personalize all aspects of the Search region on an Attachments page, with the exception of its Go submit button.
- If inline attachment links are enabled (with a **messageInlineAttachment** item style), an administrator can also personalize the Links Displayed property of this item.
- An administrator can create a new attachment link item for a region in the Create Item page, page 5-20 of the Personalization UI by selecting the Create Item icon in the Page Hierarchy Personalization page, page 4-12.
- Administrators can update, create or delete non-Oracle seeded entity, primary key and category maps for an attachment item using the Personalization UI. In the Page Hierarchy Personalization page, page 4-12, locate the attachment item you want to personalize in the page hierarchy, then expand the entityMappings to update, create or delete entity maps and primary keys. You can also expand the categoryMappings entity in the hierarchy to update, create or delete categoryMaps. Note that to create a new custom categoryMap, the new custom attachment category must first be

defined in the FND_DOCUMENT_CATEGORIES table. When you create a new category, you will also be able to personalize the Secure property, to secure the category with your defined Application Security rules.

Auto-Repeating Layout

Auto-repeating layout regions may be personalized at any administrative level, but not at the end-user level.

Bound Values

Web bean attributes implemented as bound values are not personalizable.

Branding

- The Text property of the **inContextBranding** component is system administrator personalizable.
- The **User Function Name** used for the basic (non-contextual) regular brand is personalizable; the corresponding image is not.

Buttons (Global)

As of 11.5.10, global buttons cannot be personalized.

Charts and Graphs

As an administrator, you can use the Personalize page, page 5-6 in the OA Personalization Framework, page 4-2 to personalize the following graphTable properties:

- Aggregate Function
- Allow Combination Graph
- Data Axis Label
- Data Axis scale
- Display Data Markers
- Display Secondary Y-Axis
- Graph Type
- Rendered
- Secondary Data Axis Label
- Size
- Title
- X-Axis Label

As an administrator, you can use the Personalize page, page 5-6 in the OA Personalization Framework, page 4-2 to personalize the following Graph data properties:

- Prompt

- Secondary Axis

As an administrator, you can use the Personalize page, page 5-6 in the OA Personalization Framework, page 4-2 to personalize the following Gantt chart properties:

- Show Bubble Text
- Show Dependency Lines
- Axis Major Scale
- Show Current Date
- Auto Scale
- Axis Minor Scale
- Render Description Columns

You can also personalize properties that are inherent to HGrids, page A-5.

Concurrent Processing

Concurrent Processing pages are personalizable at the Admin level.

Configurable Pages

- You should not personalize a configurable page by accessing it from the Functional Administrator Responsibility, page 4-2. Your configurable page might contain controller code that is dependent on certain parameters that are available only when the page itself is launched from its intended flow, and the page might fail with unexpected errors.

You can tell that your page is a configurable page if you attempt to access it from the Functional Administrator home page and the Page Layout Personalization page, page 4-5 appears. You should instead run the actual page in Oracle E-Business Suite and initiate the Personalization UI from the global Personalize Page link, page 4-4, on the rendered page. You can then personalize it using the Page Layout Personalization page.

Note: This restriction applies only to configurable pages whose `flexibleLayout` and `flexibleContent` regions are defined in their base metadata. This restriction does not apply to pages that start out as non-configurable, but are later personalized by administrators who add new `flexibleLayout` and `flexibleContent` regions using the Create Item page, page 5-20 in the Personalization UI. The Functional Administrator Responsibility always displays the Page Hierarchy Personalization page when it launches personalizations for these pages that have been made configurable via prior personalizations.

- If the controller of your configurable page depends on URL parameters and you select the **Personalize Page** link to personalize the configurable page, the Page Layout Personalization page will retain those URL parameters. As a result, when you select the **Return to Application** link to return to the configurable page, the URL parameters will be made available again to the configurable page's controller.

Custom HTML

Custom HTML is not personalizable.

Date Picker

- The Date Picker displayed in a secondary window cannot be personalized.
- The inline Date Picker cannot be personalized.

Declarative Page Flow

The Destination Function can be personalized, but the event cannot be (since a fireAction event cannot be configured in the Personalization module).

Dialog Pages

Dialog pages cannot be personalized.

Dynamic User Interface

You can not personalize a table content Switcher, but you can personalize the items nested in the table content Switcher, if that item is shown when you personalize the region.

Flexfields

- If you want to allow personalizations on a descriptive flexfield based on a specific localization, and you do not want to show the descriptive flexfield context poplist, then you need to explicitly call `setContextListRendered(false)` on the descriptive flex bean.
- The key flexfield Search page does not support saving search criteria for flexfield segments in personalized views.

Forms / OA Framework Page Integration

Web beans that launch Oracle Applications forms from OA Framework pages are not personalizable.

formValue

When the Rendered property is set to **True** for a formValue item, the Rendered property for the formValue item will not be personalizable. That is, you cannot hide a formValue item on a page as it will cause problems in the page.

HGrid

You can personalize a HGrid to a limited extent by hiding/showing columns and renaming column labels in the HGrid.

Hide/Show

As of 11.5.10, Hide/Show web beans cannot be personalized.

Images in Your Pages

Images can be personalized.

Locator Element: Breadcrumbs

Breadcrumbs are not directly personalizable. If the **pageLayout** Title property value is personalized, the breadcrumb link text will reflect this change (assuming the breadcrumb link text is not set programmatically).

Mobile Applications

In the case of a PDA (personal digital assistant) the following buttons are not rendered:

- The Save search button in both the Simple and Advanced search regions.
- The Personalize button in the Views page.

Therefore, the User View Definition page is not supported on a PDA. Use the desktop interface to define the views that you want to access on the PDA.

Page Contents Bottom Line

You can change the color of the "ski" image using the Customizing Look-and-Feel UI. See the Customizing Look-and-Feel (CLAF) topic, page 2-4 for more information.

Page Footer

- If you wish to personalize the URL that points to the Privacy Statement for a page that displays a *standard* Copyright and Privacy (that is, its Auto Footer property is set to **true**), set the Scope to **OA Footer**, in the Choose Personalization Context page, page 5-4 of the Personalization UI.
- If you wish to personalize the URL that points to the Privacy Statement for a page that displays a *custom* Copyright and Privacy (that is, its Auto Footer property is set to **false**), set the Scope to **Page** in the Choose Personalization Context page, page 5-4 of the Personalization UI. In the following Page Hierarchy Personalization page, page 4-12, identify and personalize the **Privacy** page element.

Printable Page

Printable pages are not personalizable.

Processing Page

A processing page is not personalizable.

Rich Text Editor

The Rich Text Editor properties, `Read Only` and `Rendered`, may be personalized.

Save Model (Warn About Changes)

The Warn About Changes property is not personalizable.

Shuttle

The properties of a shuttle region may be personalized at the Admin level but not at the end-user level.

Standard Web Widgets

- If the **Required** property of a **messageTextInput** (text input) item is set to `yes`, its **Read Only** property does not appear on the Personalize page.
- If the **Required** property of a **messageChoice** (poplist) item is set to `yes`, its **Read Only** property does not appear on the Personalize page.
- There is no personalization support for a dynamic poplist because it can only be implemented programmatically.

SubTab Navigation

Limited properties of SubTabs, such as `Rendered`, may be personalized at the Admin level but not at the end-user level.

Switchers (Application, Context and Table Content)

- You can not personalize Application Switchers.
- You can not personalize Context Switchers.
- You can not personalize a table content Switcher, but you can personalize the items nested in the table content Switcher, if that item is shown when you personalize the region.

Tables - Advanced

For the columns in an advanced table to be user personalizable, the User Personalization property must be set to `true` for the following components in the Advanced Table: the `advancedTable` region itself, its column containers, its `sortableHeaders`, and its column leaf items.

- **Column Rendering:**

As an administrator, you can use the Admin-level Personalization UI to hide or show a column in an advanced table (if it is not defined under a query region) by setting the column's **Rendered** property to `false` or `true`, respectively. This automatically hides or shows the column and its item. If the advanced table is defined under a query region, however, you must set both the column's and the item's **Rendered** properties to `false` or `true`.

- **Column Reordering:**

Currently, end-users cannot reorder the columns in an advanced table, when creating a personalized view, as the arrow buttons in the shuttle, used to reorder the columns, do not render.

For Admin-level personalizations, columns and column groups can only be reordered within their region boundary. For example, the column or column group child of a given column group region may be reordered within that given column group region. However, the column or column group child from that given column group may not be moved to another column group region, as that would cross over the region boundaries.
- **Row Headers:**

Row headers are not personalizable.
- **Column Headers:**
 - The prompt or label of a column or column group may be personalized. Modify the Prompt property of the Sortable Header component that resides under the columnHeader for that column or column group.
 - Column headers cannot be reordered at the user level but can be at the admin level.
- **Column Span:**
 - When an end-user creates a personalized view of the advanced table region, the Available Columns/Columns Displayed shuttle in the Create/Update/Duplicate page, page 6-4 appends the complete hierarchy of columnGroup names, if any are defined as the parent of the column, to the actual column name listed. This ensures that a user hides/shows the correct column, especially in the case where multiple columns of the same name may exist within different columnGroups.
 - When creating a personalized view of an advanced table, if you hide all the columns under a columnGroup, the columnGroup itself is automatically hidden.
- **Row Span:**

An advanced table with row span is not personalizable.
- **Navigation:**

Not applicable.
- **Selection and Control Bar:**
 - The label of the Select column cannot be changed in accordance with the BLAF guidelines [<http://otn.oracle.com/tech/blaf/specs/tables.html#rowselectioncontrols>] on the Oracle Technology Network (OTN).
 - Reordering of the Select column is not supported, as the Select column should always be the first column, in accordance with the BLAF guidelines [<http://otn.oracle.com/tech/blaf/specs/tables.html#rowselectioncontrols>] on OTN.
 - Selectors can be hidden or shown.
 - The Control bar button label may be updated.
- **Table Actions:**

A tableActions is a named child of the advanced table, and hence the actual table action components are regions or items. These children of table actions can be individually personalized at an admin level, but not at the user level.

- **Sorting:**

In an end-user personalizable page, if you set sorting on an advanced table column that contains a nested region using the Create View page, page 5-42, but find that the change is not taking effect on the table, it is likely that the Sort By View Attribute property on that column was not set by the developer. If this is the case, the personalization administrator would have to set the Sort By View Attribute property on that column first, using the Admin Personalization UI, page 5-6.

- **Adding Rows:**

The label or prompt of the Add Another Row button, and the number of rows to add, can be personalized at the Admin-level only.

- **Totalling:**

- You can not personalize the Total footer row (which displays the Recalculate button) of an advanced table using OA Personalization Framework. However, if you declaratively implement an advanced table region that contains a total named child, the advanced table can be personalized at the admin level, page 5-6 or user level, page 6-1 to calculate a total or disable totalling for any of its numeric data columns.

- You can not enable totalling for the first column in a table. If you attempt to personalize the first column of an advanced table by setting the Total Value property to **True** in the Personalize page, page 5-6, it is simply ignored. To total the contents of that first column, you must use the OA Personalization Framework, page 3-1 to reorder the column within the advanced table so that it is no longer the first column in the table, then set the Total Value property to True for that column.

- **Detail Disclosure:**

An advanced table with Detail Disclosure can be personalized at the admin level, page 4-1 but not at the user level, page 6-1.

- **Advanced Table-in-Advanced Table:**

An Advanced Table-in-Advanced Table can be personalized at the admin level, page 4-1 but not at the user level, page 6-1.

- **Formatting a Table:**

- The full table formatting properties can be personalized at the admin level, page 4-1 but not at the user level, page 6-1.

- The column formatting properties can be personalized at the admin level, page 4-1 but not at the user level, page 6-1.

- The row formatting properties can be personalized at the admin level, page 4-1 but not at the user level, page 6-1.

- The column header formatting properties can be personalized at the admin level, page 4-1 but not at the user level, page 6-1.

Tables - Classic

- **Reordering Columns:**

If you reorder the columns in a table, such that a checkbox or radio button item results in the first column, the checkbox or radio button in that first column will assume the behavior of a Selection checkbox or radio button, respectively.

For more information about Selection checkboxes, refer to the **Classic Tables > Table Features and Components > Selection and Control Bar** section in the Implementing Specific UI Features chapter of the *Oracle Application Framework Developer's Guide*.

- **Row Headers:**

Row headers are not personalizable.

- **Column Headers:**

The column header of a column that contains a nested region (such as a `FlowLayout` region) cannot be personalized at the Admin or User level. The column itself, however, can still be hidden or reordered at any level.

- **Navigation:**

Not applicable.

- **Selection and Control Bar:**

- The label of the Select column cannot be changed in accordance with the BLAF guidelines [<http://otn.oracle.com/tech/blaf/specs/tables.html#rowselectioncontrols>] on the Oracle Technology Network (OTN).
- Reordering of the Select column is not supported, as the Select column should always be the first column, in accordance with the BLAF guidelines [<http://otn.oracle.com/tech/blaf/specs/tables.html#rowselectioncontrols>] on OTN.
- If you reorder the columns in a table, such that a checkbox or radio button item results in the first column, the checkbox or radio button in that first column will assume the behavior of a Selection checkbox or radio button, respectively.
- Selectors can be hidden or shown.
- The Control bar button label may be updated.

- **Table Actions:**

A `tableActions` is a named child of the table, and hence the actual table action components are regions or items. These children of table actions can be individually personalized at an admin level, but not at the user-level.

- **Sorting:**

In an end-user personalizable page, if you set sorting on a nested region column in a table using the Create View page, page 5-42, but find that the change is not taking effect on the table, it is likely that the Sort By View Attribute property on that column was not set by the developer. If this is the case, the personalization administrator would have to set the Sort By View Attribute property on that column first, using the Personalize Page, page 5-6.

- **Adding Rows:**

There is currently no support for personalizing the Add Another Row button in the table footer.

- **Totalling:**

Users can calculate a total for a table column using OA Personalization Framework. See Personalize Page, page 5-6 for additional information on how to personalize a table to calculate a total for a column with numeric content.

- **Detail Disclosure:**

There are no restrictions for Admin-level personalizations of regions or items under a detail disclosure, however, there is no support for user-level personalizations of the same.

- **Table-in-Table:**

There are no restrictions for Admin-level personalizations of regions or items in the inner table, however, there is no support for user-level personalizations of the same.

- **Formatting a Table:**

There are no restrictions for personalizing full table formatting, column formatting, row formatting, or column header/row header formatting.

Tree

Tree components are not personalizable.

Workflow Worklist

As the Oracle Workflow Worklist is a shared region, personalizing it at the site, location or organization level affects all pages/applications that contain the region. If you want to limit the scope of your changes you should create function or responsibility-level personalizations as appropriate. Starting in release 11.5.10, it is also possible to create a page-specific personalization for shared regions. See Per Instance versus Global Personalizations, page 5-6 for additional information.

OA Framework Personalization and Customization Known Issues

This appendix covers the following topics:

- Overview
- Personalization/Extensibility
- Translation
- Customizing Look-and-Feel (CLAF)

Overview

This appendix lists known bugs and enhancements for OA Personalization Framework and the Customizing Look-and-Feel feature.

Warning: This document does not list all known bugs and enhancements. Instead, it lists selected, key issues that we hope will facilitate your personalization and customization efforts.

Personalization/Extensibility

- All seeded function and user-level personalizations are protected against upgrades. However, HR has also provided some seeded Localization-level personalizations that have been modified by customers and these personalizations are not protected against upgrades. After an upgrade, customers must redo any personalizations they have made to these HR seeded Localization-level personalizations.
- Bug 2870101 (CANNOT DISPLAY COLUMNS HIDDEN BY DEVELOPER MODE FUNCTION CUSTOMIZATION) - In OA Framework 11.5.57, if you have a seeded Function-level personalization containing an item that is set to not be displayed, the item does not display even if you personalize the region at a lower personalization level, like Site, and specify to display the item. In prior releases of OA Framework, the item would display, if you personalized the region at a lower level. The current problem arises due to the way saves are handled in the 11.5.57 architecture.
- Bug 2919819 (ITEMS WITH RESERVED WORD IDS NOT MIGRATED) - In Oracle 9i JDeveloper OA Extension, you cannot have components that have a "." (dot) in its name or ID, as it has a special meaning in OA Extension. If a component contains a "." in its ID, the customer Personalization Migration Tool will replace the character

with another valid character (making sure, in doing so, it doesn't create a duplicate ID) and document the new ID change in the tool's log file. If the new ID is an invalid xml nmtoken, the tool continues its current behavior and logs it as an error and does not migrate the component.

Translation

Bug 2678638 - If the input document path to the XLIFF Extractor contains a relative path, such as '.' or '..', in the portion of the pathname after the root directory, the XLIFF Extractor will print an error message and exit with exit code = 1.

Customizing Look-and-Feel (CLAF)

- Bug 4371146 - If you reuse an existing Look-and-Feel (LAF) name to create a new LAF, overwriting the existing LAF with that name, then you must stop and restart Oracle HTTP Server after creating the new LAF for the changes to take effect.
- Bug 4311580 - Simplify the CLAF UI to remove Include Property as a property type for a style. This should be handled transparently.
- Bug 4300086 and Bug 4300083 - Fix the adding of styles and properties to a stylesheet.
- Bug 4281940 - Use UIX IconBean instead of ImageBean to render different types of icons.
- ER 4226036 - Support registering uit template renderers in the CLAF UI.
- ER 4177696 - Create a set of sample skins.
- ER 4093799 - Develop utility APIs to return all Valid CSS properties.
- ER 4111170 - Show color picker for color based components. This is dependent on the color picker ER 2576865. Color picker is ready for programmatic support. This is an open issue whether color enhancement will be included in the base release of CLAF.

OA Framework Personalization Caveats

This appendix covers the following topics:

- Overview
- Personalization Caveats - for Customers

Overview

This appendix lists the personalization caveats you should be aware of when modifying and patching pages.

Personalization Caveats - for Customers

- All IDs must be unique. If an ID is already being used in the base definition of a page, do not use it again for new items or regions that you create. To reinforce this rule, you should add an intelligible prefix to your IDs to keep them in a separate namespace from any Oracle-seeded IDs.
- A personalization that is inadvertently hidden due to an ID change of the base definition, may reappear if you create a new item with the same ID as that referenced by the personalization.
- Certain types of personalizations may not appear "correctly" because of the nature of specific web beans in the page (particularly "default renderers").

Note: Default single column regions will not allow a button to be added at the top of the region even if the entities in the region are reordered to put the button at the top.

Personalization FAQ

This appendix covers the following topics:

- Overview
- Customizing Look-and-Feel (CLAF)

Overview

This FAQ (frequently asked questions) document describes common problems and solutions related to personalizing OA Framework-based applications.

Customizing Look-and-Feel (CLAF)

Questions

1. **When I create a new Look-and-Feel extending Simple Look-and-Feel, there is no space between the global links. How do I fix this?**
(See answer, page D-1)
2. **In customizing fonts using the CLAF UI, when I change the DefaultFont style to use a different font size, the settings don't take effect. How do I fix this?**
(See answer, page D-2)
3. **How do I package a new LAF configuration for delivery to a customer?**
(See answer, page D-2)

Answers

1. **When I create a new Look-and-Feel extending Simple Look-and-Feel, there is no space between the global links. How do I fix this?**
You need to redefine the style GlobalButtonText in your new custom look and feel with the following property-value pair:
 - Property: **padding**
 - Value: 5px (or whatever number of pixels you want to define as the space between the global links)

Verify that your custom style sheet file has the following tag generated:

```
<style name="GlobalButtonText">
<property name="padding">5px</property>
</style>
```

You can use the CLAF UI to add this style, or you can manually insert the above set of tags in your custom style sheet.

(Back to question, page D-1)

2. **In customizing fonts using the CLAF UI, when I change the DefaultFont style to use a different font size, the settings don't take effect. How do I fix this?**

When you extend the Simple Desktop Look-and-Feel to create your custom Look-and-Feel, you also inherit browser- and locale-specific style sheets, page 2-29. The DefaultFont style is defined for the following combinations:

- browser=ie
- browser=ie, locale=windows

Hence when you use the CLAF UI to define the DefaultFont, it gets written for the no-browser, no-locale case, and would be overridden by the definitions in any matching browser/locale style sheets.

To fix this, you, in turn, need to override the most matching browser/locale combination. Currently, you have to do this manually by editing your custom style sheet as follows:

```
<styleSheet locale="windows" browser="ie">
<style name="DefaultFont">
<property name="font-size">7pt</property>
</style>
</styleSheet>
```

(Back to question, page D-1)

3. **How do I package a new LAF configuration for delivery to a customer?**

You should create a .zip archive of all of the relevant files and ship it to the customer.

Make sure that you preserve the folder structure of the LAF configuration when creating and extracting the archive.

The archive must contain the following files:

- All custom LAF .xml files and metadata files generated in \$HTML_TOP/cabo/lafs.

Note: For every custom LAF you create, two .xml files are generated:

- *lookandfeel_id.xml*
- *lookandfeel_id-metadata.xml*

- All style definition (.xss) files generated in \$HTML_TOP/cabo/styles.

Note: For every custom LAF you create, one .xss file is generated, named *lookandfeel_id.xss*

- All .uit files that you placed in the folder \$HTML_TOP/cabo/templates/*lookandfeel_id*.

Note: To customize the template renderer of each component, you must create a *componentName.uit* file and place it in `$HTML_TOP/cabo/templates/lookandfeel_id`.

- All custom image (.gif) files that you placed in the folder `$HTML_TOP/cabo/images/lookandfeel_id` when you created the custom LAF by the same ID.

Note: To customize icons for a custom LAF, you must place its custom images in the folder `$HTML_TOP/cabo/images/lookandfeel_id`.

(Back to question, page D-1)

Index

A

Adding New Items
 Overview, 4-17

B

Branding, 2-1

C

Customization Primer, 1-1
 Overview, 1-1

D

Deploying personalizations, 9-1
Deploying personalizations in OA Extension, 9-1,
9-3

E

Export tool for personalizations, 9-4

I

Icons, 2-21
 controlling, 2-1
Import tool for personalizations, 9-9
Items
 create new, 4-17

L

Logo size
 See Branding
Lookup codes, 2-22

M

Menus, 2-22
Messages, 2-22
Migrating personalizations from AK, 10-1

P

Personalization levels, 3-2
Personalizations, 2-1, 3-1
 deploying, 9-1
 translating, 8-1
Personalizing pages, 3-1
 Overview, 3-1

R

Region Item
 create new with OA Personalization
 Framework, 4-17

S

Security
 and menus, 2-22

