Oracle® Identity Manager

Administrative and User Console Customization Guide

Release 9.1.0

E10362-02

June 2008

Oracle Identity Manager Administrative and User Console Customization Guide, Release 9.1.0

E10362-02

Primary Authors: Debapriya Datta, Shiladitya Guha, Lyju Vadassery

Contributing Author: Don Gosselin

Contents

Preface

This guide describes the basic procedures for customizing the appearance and behavior of the Oracle Identity Manager Administrative and User Console. This console is the client portion that is accessible through a Web browser.

Audience

This guide is intended for those responsible for ensuring that the Administrative and User Console adheres to the design for Web applications in your organization. This guide can also be used by personnel responsible for customizing the labels, descriptive text, and workflow of the Administrative and User Console.

It is assumed that users of this guide are familiar with the Oracle Identity Manager system and documentation (specifically, *Oracle Identity Manager Administrative and User Console Guide*).

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.

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, 7 days a week. For TTY support, call 800.446.2398. Outside the United States, call +1.407.458.2479.

Related Documents

For more information, see the following documents in the Oracle Identity Manager documentation set:

- *Oracle Identity Manager Release Notes*
- *Oracle Identity Manager Installation and Configuration Guide for JBoss Application Server*
- *Oracle Identity Manager Installation and Configuration Guide for BEA WebLogic Server*
- *Oracle Identity Manager Installation and Configuration Guide for IBM WebSphere Application Server*
- *Oracle Identity Manager Best Practices Guide*
- *Oracle Identity Manager Globalization Guide*
- *Oracle Identity Manager Design Console Guide*
- *Oracle Identity Manager Administrative and User Console Guide*
- *Oracle Identity Manager Tools Reference*
- *Oracle Identity Manager Audit Report Developer's Guide*
- *Oracle Identity Manager Integration Guide for Crystal Reports*
- *Oracle Identity Manager API Usage Guide*
- *Oracle Identity Manager Concepts*
- *Oracle Identity Manager Reference*

Documentation Updates

Oracle is committed to delivering the best and most recent information available. For information about updates to the Oracle Identity Manager documentation set, visit Oracle Technology Network at

`http://www.oracle.com/technology/documentation/index.html`

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen (or text that you enter), and names of files, directories, attributes, and parameters.

1

Overview

The *Oracle Identity Manager Administrative and User Console Guide* explains how to customize the following components of your Oracle Identity Manager Administrative and User Console:

- General page layout
- Descriptive text, labels, and error messages
- Colors, fonts, and alignment
- Logos
- Self-registration, user initiated profile editing, and related approvals
- Field configuration on pages
- Menu selections that are available to users

To configure the various aspects of the Administrative and User Console, you must edit certain files. These files and how to edit them are described in the various sections of this Guide. For example, to learn how to customize the general page layout, see Chapter 2, "Customizing General Page Layout" .

> **See Also:** *Oracle Identity Manager Globalization Guide* for information about how to globalize Oracle Identity Manager Administrative and User Console

Accessing Administrative and User Console Customization Files

During installation, Oracle Identity Manager is deployed to your application server as an Enterprise ARchive (EAR). This archive file contains some of the files for customizing your Administrative and User Console, for example Xellerate.css, xlWebAdmin.properties, and xlDefaultAdmin.properties. The name of the EAR file varies depending on your application server, as follows:

- JBoss Application Server or Oracle Application Server—XellerateFull.ear
- IBM WebSphere Application Server—Xellerate.ear
- BEA WebLogic Server—WLXellerateFull.ear

To access the files for customizing your client, you unpack a Web Archive (WAR) file, make the required edits, repack the WAR file, and run a script that regenerates the EAR file and deploys it to your application server.

To customize the Administrative and User Console, ensure that your application server is running and then perform the following tasks:

1. Uncompress the entire `xlWebApp.war` file from the *OIM_HOME*`/xellerate/webapp` directory into any directory.
2. Modify the required files to be customized.

   Note: Oracle recommends that you back up any files before modifying them.

3. Create a new archive file called `xlWebApp.war` by using the `jar` utility.
4. Copy and overwrite the `xlWebApp.war` file in *OIM_HOME*`/xellerate/webapp`.
5. Run the `patch_`*appserver* script from the *OIM_HOME*`/xellerate/setup` directory, where *appserver* is the name of your application server. For example, if you are using BEA WebLogic Server, then this file is called `patch_weblogic`.
6. Restart your application server.

2

Customizing General Page Layout

This chapter describes how to customize the overall layout of the pages in the Oracle Identity Manager Administrative and User Console. It contains these topics:

- Overview of Page Layout Customization
- Files to Modify
- Example of How to Customize Your General Page Layout

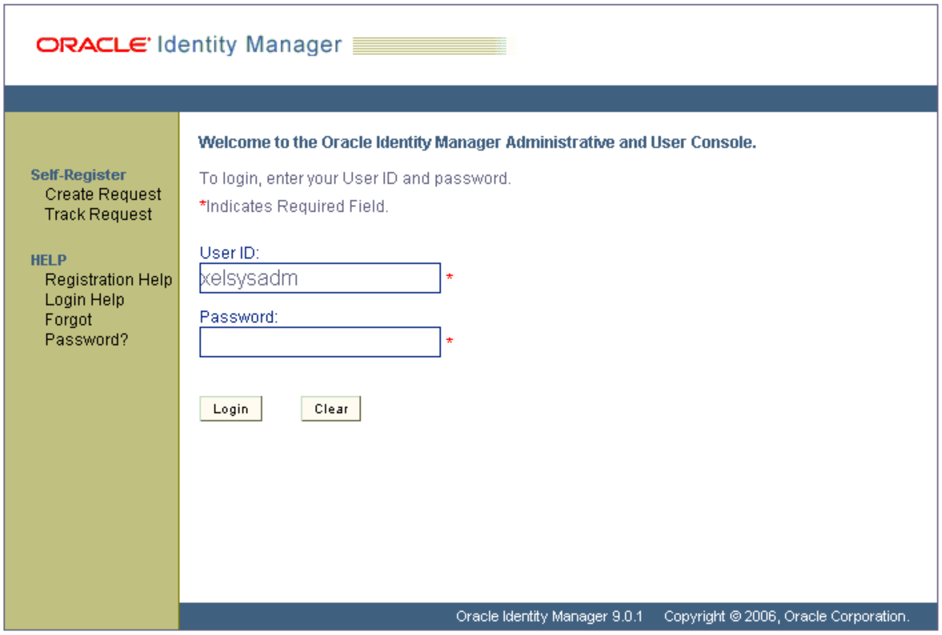
Overview of Page Layout Customization

Page layout customization includes but is not limited to:

- Modifying the width of banners
- Modifying the alignment and orientation of images
- Adding links to the affected regions, for example, a link to your corporate home page from the company logo)

The Oracle Identity Manager Administrative and User Console page is divided into regions. Figure 2–1 shows the Oracle Identity Manager Administrative and User Console login page. The page layout is controlled by using a JSP page.

Figure 2–1 Oracle Identity Manager Administrative and User Console Layout



Note: This chapter assumes you are a developer familiar with JSPs and Struts. In addition, because editing the files mentioned in this chapter controls how the pages are rendered, and potentially introduces cosmetic anomalies, Oracle recommends that you test the effects of your changes before releasing them into a production environment.

Files to Modify

To customize page layout, you edit these files:

```
xlWebApp\tiles\common\tjspHeader.jsp
xlWebApp\tiles\common\tjspFooter.jsp
```

Example of How to Customize Your General Page Layout

To control the layout of your Administrative and User Console GUI elements, edit the appropriate JSP file. For example, if you want your company logo to appear to the right of the header banner and the Oracle Identity Manager product logo to appear on the left, then perform the following steps:

1. Locate the relevant portion of the `tjspHeader.jsp` file that controls the display of the logo image files to be aligned. This includes sections of the file that refer to the following:

   - `global.image.clientlogo`

     This reference by default points to the Oracle Identity Manager logo, or the product logo.

   - `global.image.xelleratelogo`

     This reference points to a placeholder image. If you want to add another image, replace this logo.

2. Swap the references of the two property variables within the file, as shown in the following example. For example, change the settings from this:

   Change the code from:

```
<TR>
<TD valign="center" align="center" width="150px" height="60px"
class="LogoCell">
     <html:img pageKey="global.image.clientlogo"/>
   </TD>

   <TD valign="center" align="right" height="60px" colspan='2' >
     <html:img pageKey="global.image.xelleratelogo"/>
   </TD>

   <TD width='20px'>
     <html:img width='20px' pageKey="global.image.spacer"/>
   </TD>
</TR>
```

   Change the code to:

```
<TR>
  <TD valign="center" align="center" width="150px" height="60px"
class="LogoCell">
    <html:img pageKey="global.image.xelleratelogo"/>
  </TD>

  <TD valign="center" align="right" height="60px" colspan='2' >
    <html:img pageKey="global.image.clientlogo"/>
  </TD>

  <TD width='20px'>
    <html:img width='20px' pageKey="global.image.spacer"/>
  </TD>
</TR>
```

3. Save the changes to the `tjspHeader.jsp` file.
4. Test the display of your Administrative and User Console to ensure that the intended changes have taken effect. Do this by reloading the Administrative and User Console page.

3

Customizing Text

This chapter describes how to customize the descriptive text and labels that appear on the pages of the Administrative and User Console. It also describes how to edit the error messages that appear under the specific conditions that generate them.

You can customize the following text:

- Descriptions of procedures, such as instructional text
- Labels for pages
- Labels for fields within pages
- Labels that appear on buttons
- Labels for links to other pages
- Copyright dates and information

This chapter discusses the following topics:

- Files to Modify
- Customizing Descriptive Text and Labels

Files to Modify

To customize the text in the Administrative and User Console, you edit the `xlWebApp\WEB-INF\classes\xlWebAdmin.properties` file or the locale-specific properties file.

Customizing Descriptive Text and Labels

The text displayed in the console is stored in the `xlWebAdmin.properties` file. For any label, procedural instruction, or error message that you want to edit, perform the following steps:

1. Access the JSP page in which the text is displayed.
2. Reference the property associated with the text.
3. Lookup the property name in the `xlWebAdmin.properties` file.

Properties that control the text on more than one page, or that are associated with multiple product functionalities, are listed in one of the `GLOBAL` sections of the file, such as `GLOBAL` messages. These properties are also divided by type-specific labels, such as messages, buttons, and labels, to organize the groups of the properties.

The properties that control the text that is displayed in more specific contexts of the product functionality, such as create user and self-registration, are listed in sections labeled in a function-specific manner, such as `MANAGE USER` labels.

The width of the labels in the Workflow Visualizer are calculated by converting pixel lengths to character lengths. An incorrect conversion can result in truncated text or extra white space surrounding the label. The `xlRichClient.properties` file, or locale-specific file such as xlRichClient_jp.properties for Japanese, contains a property named `global.workflowRenderer.labelWidthFactor`. This property is used in the conversion of label text from pixel length to character length. To modify the width of labels in the Workflow Visualizer, modify the integer value that is assigned to this property. A higher integer will increase label widths, and a lower integer will decrease them.

> **See Also:** The "Accessing Administrative and User Console Customization Files" section on page 1-1 for details on how to access the `xlWebAdmin.properties` file.

4

Customizing Colors, Fonts and Alignment

This chapter describes how to customize the look and feel of the Administrative and User Console. You do this by editing the cascading style sheets.

Files to Modify

To customize colors, fonts, and alignments, you edit the `xlWebApp\css\Xellerate.css` file or the locale-specific cascading style sheets.

Customizing the Appearance of Administrative and User Console

The `Xellerate.css` style sheet defines the color, font, point size, and alignment of the Administrative and User Console pages. After you determine how you want to edit the appearance of the Administrative and User Console, perform the following steps:

1. View the source for the page.
2. Determine the style sheet class associated with the element on the page that you want to change.
3. Lookup the style sheet class name within the `Xellerate.css` file.

The `Xellerate.css` file contains context labels. Use these context labels to locate the class to edit when customizing a particular aspect of the Administrative and User Console appearance. In addition, the classes within this file are organized according to the region of the screen, such as header, body, and footer, and the HTML elements they affect, such as links, tables, and check boxes.

> **See Also:** The "Accessing Administrative and User Console Customization Files" section on page 1-1 for details on how to access the `Xellerate.css` style sheet.

5

Customizing Logos

This chapter describes how to add your own corporate logo or that of your customers or partners to the Administrative and User Console. You can do this by copying a logo file into the EAR file, and editing the relevant values in the `xlDefaultAdmin.properties` file. This lets you brand the appearance of the solution you are providing to your customers, or apply the branding that your partners may require if they want their own logos displayed.

Figure 5–1 shows the default logo that is displayed on the Administrative and User Console login page.

Figure 5–1 Default Logo on Administrative and User Console Login Page

> **See Also:** Chapter 2, "Customizing General Page Layout" for instructions on how to align logos within the banner region

Files to Modify

The `xlWebApp\WEB-INF\classes\xlDefaultAdmin.properties` file
The `xlWebApp\images` directory

Replacing the Product Logo

To replace the default Oracle Identity Manager logo that is displayed to the left in the uppermost banner:

1. Place the GIF image file that contains your logo in the `\xlWebApp\images` directory.
2. Open the `xlDefaultAdmin.properties` file.
3. In the `xlDefaultAdmin.properties` file, locate the GLOBAL images section. In this section, set the `global.image.clientlogo` property to the full path and name of the image file that contains the desired logo. For example, if the logo is `/images/mycorporate_logo.gif`, then the default value of the `global.image.clientlogo` property is `/images/client_logo.gif`.
4. Redeploy the Enterprise ARchive (EAR) file. The name of the EAR file varies depending on your application server, as follows:
   - JBoss Application Server or Oracle Application Server—XellerateFull.ear

- IBM WebSphere Application Server—Xellerate.ear
- BEA WebLogic Server—WLXellerateFull.ear

See Also: The "Accessing Administrative and User Console Customization Files" section on page 1-1 for details on how to access the `\xlWebApp\images` directory and the `xlDefaultAdmin.properties` file and how to redeploy the EAR file

Inserting the Company Logo

To replace the placeholder image that is displayed to the right in the uppermost banner:

1. Place the image file that contains your logo in the `\xlWebApp\images` directory.
2. Open the `xlDefaultAdmin.properties` file.
3. Locate the GLOBAL images section and set the `global.image.xelleratelogo` property to the full path and name of the image file that contains the desired logo. For example, if the logo is `/images/myproductbrand_logo.gif`, then the default value of this property is `/images/xellerate-trans-grey.gif`.
4. Redeploy the `Xellerate.ear` file.

See Also: The "Accessing Administrative and User Console Customization Files" section on page 1-1 for details on how to access the `\xlWebApp\images` directory and the `xlWebAdmin.properties` file and how to redeploy the EAR file

6

Customizing Self-Registration, User Profile Management, and Service Accounts

This chapter explains how to customize the self-registration process for creating user accounts, and how users edit their account profiles in the Administrative and User Console. It also explains how to customize service accounts, that is, general administrator accounts shared by several users and used for maintenance purpose.

For the self-registration process, you must determine:

- Whether or not self-registration is allowed?
- Which fields can be used during the registration process?
- Which fields are displayed, and which are mandatory when a user is self-registering?
- Whether or not approvals are required for self-registration?

For user-initiated changes to the user profile, you must determine:

- Which fields the user may edit in their own profile?
- Whether or not approvals are required for user-initiated profile changes?
- Which fields the approver may edit or override?

This chapter discusses the following topics:

- Files to Modify
- Customizing Self-Registration Options
- Customizing Self-Initiated Profile Management Options
- Customizing Service Accounts

Files to Modify

To customize self-registration and self-editing of profiles and service accounts, you edit this file:

OIM_HOME`\xellerate\config\FormMetaData.xml`

In addition, you must edit the relevant records within the Oracle Identity Manager Design Console.

Customizing Self-Registration Options

This section explains how to customize the self-registration functions of the Administrative and User Console.

Specifying Whether Self-Registration is Allowed

To specify whether or not self-registrations is allowed:

1. Log in to the Design Console.
2. Access the System Configuration form by navigating to the Administration directory.
3. Query to find the `Is Self-Registration Allowed?` property.
4. Set the value of the `Is Self-Registration Allowed?` property to `TRUE` to allow users to self-register. To disallow users from self-registering, set the value of this property to `FALSE`.
5. Save the changes.

Defining Custom Fields for the User Information Pages

On the user self-registration and profile pages of the Console, there are several fields already defined for user information. You can create custom fields for users to enter information when they are self-registering or editing their profiles.

Note: The default system fields on the self-registration pages are listed in the `FormMetaData.xml` file, in the `<!-- User Self Registration and User Profile Modification section -->` section.

To create custom fields for the user self-registration page:

1. Log in to the Design Console.
2. Access the User Defined Field Definition form by navigating to the Administration directory.
3. Query to find the Users form.
4. Create a user-defined field, for example, a social security number.

   See Also: The "Adding a User-Defined Field to an Oracle Identity Manager Form" section of the *Oracle Identity Manager Design Console Guide* for information about creating a user-defined field

5. Save and close the User Defined Fields form.
6. In a text editor, open the Oracle Identity Manager `FormMetaData.xml` file, and locate the section `<!-- User Self Registration and User Profile Modification section -->`.
7. To define the attribute, add an entry for it in the following format:

   ```
   <Attribute name="identifier" label="field_label"
   displayComponentType="datatype" map="database_column" />
   ```

   In this format:

- *identifier* is the name used to specify this field when you define a page.
- *field_label* is the word displayed to the user next to the field in the Administrative and User Console.
- *datatype* is the data type of the field displayed in the Console.
- *database_column* is the database column name, specified when you created the field definition by using the Design Console. User-defined fields are prefixed with USR_UDF_. Therefore, if you type in SSN in the Design Console, then this value is USR_UDF_SSN. For example:

```
<Attribute name="USR_UDF_SSN" label="SSN" displayComponentType="TextField"
              map="USR_UDF_SSN" />
```

> **Note:** This is the same information you enter when you use the Design Console to create a field definition.

8. Save the changes to this file.

After you create a custom field and define it, you can use it in a page definition.

Defining the User Self-Registration Page

You can specify which fields are displayed, and which are required when users self-register. To display a custom field, you must reference it in the section of the FormMetaData.xml file for the self-registration user page. This is explained in "Defining Custom Fields for the User Information Pages" on page 6-2.

To define the user self-registration page:

1. Open the FormMetaData.xml file.
2. Locate the section called <Form name="SelfRegistrationUserForm">. This section contains the definition of the User Self Registration page. To add a field to the page, add a reference to the field in a new row in the following format:

```
<AttributeReference optional=true/false>identifier</AttributeReference>
```

   In this format:

   - *identifier* is the name you specify in the definition section of the file.
   - optional specifies whether a field is required or optional. Setting optional to false makes the field required, and true makes it optional.

   For example, to add a new required field with the identifier, USR_UDF_SSN, add the following line:

```
<AttributeReference optional="false">USR_UDF_SSN</AttributeReference>
```

> **Note:** The default value is true, so if a field is required, you can omit this attribute.

3. Save the changes to this file and close it.

Note: To remove a field from the self registration page of the Administrative and User Console, remove the row from the `<Form name="SelfRegistrationUserForm">` section that represents the field.

Specifying Whether Approvals are Required for Self-Registration

To specify that approvals are required for self-registration, ensure that the User Registration approval process contains at least one manual nonconditional task.

To specify that approvals are not required for self-registration:

1. Ensure that the user registration approval process does not contain any non-conditional tasks.
2. Ensure that no fields on the registration-related approval pages are set as required that are not present on the User Self-Registration page.

   See Also: The "Defining Custom Fields for the User Information Pages" section on page 6-2 to find which section of the `FormMetaData.xml` file to edit

Note: By default, Oracle Identity Manager does not present the Organization and Role fields to the user when they are self-registering. Values for these fields are required to register a user. As a result, the values of these fields must be set on the request, under the User Information branch, by an administrator or approver.

Customizing Self-Initiated Profile Management Options

By default, all users, once registered, are able to edit their own profile information. You can control which fields can be edited by the users within their profile, whether or not approvals must be obtained to allow these edits, and if so, which fields the approver can update.

Specifying Fields that Users May Edit in Their Profiles

To specify the fields that users can edit in their profiles:

1. Open the `FormMetaData.xml` file.
2. Locate the `ProfileModificationUserForm` section.
3. Set the `editable` parameter to `TRUE` for each of the fields that you want to enable for the users to edit.
4. Save and close the file.

Specifying Whether Approvals Are Required for User-initiated Profile Changes

To specify that approvals are required for user-initiated profile changes, ensure that the User Profile Edit approval process contains at least one manual nonconditional task.

To specify that approvals are not required for user-initiated profile changes:

1. Ensure that the User Profile Edit approval process does not contain any nonconditional tasks.
2. Ensure that no fields on the profile update-related approval pages are set as required that are not present on the Modify Account Profile page.

> **See Also:** The "Defining Custom Fields for the User Information Pages" section on page 6-2 for the section of `FormMetaData.xml` file to edit.

Specifying the Fields That the Approver May Edit or Override

To specify the fields that the approver might edit or override:

1. Open the `FormMetaData.xml` file.
2. Locate the section called `ProfileModificationApprovalForm`.
3. Set the `editable` parameter to `TRUE` for each of the fields you want to enable of the users to edit.
4. Save and close the file.

Customizing Service Accounts

Service accounts are general administrator accounts, such as `admin1` and `admin2`, which are used for maintenance purpose. They are typically shared by a set of users. Usually, these accounts enable a system, as opposed to a user, to interact with another system. The model for managing and provisioning service accounts is slightly different from normal provisioning.

> **Note:** Oracle Identity Manager provides only back-end support for service accounts and they can be managed only through APIs.

Service Account Behavior

Here are some features of service account behavior:

- Service accounts are requested, provisioned, and managed in the same way as regular accounts. A service account is similar to a regular account. It uses the same resource object, provisioning processes, and process or object forms as a regular account. It differs from a regular account by a flag. This flag is set by the user requesting the resource, or by the administrator directly provisioning the resource.
- During its lifecycle, a service account can be changed to a regular account, and a regular account can be changed to a service account. When either change occurs, Service Account Changed task functionality is triggered.
- When a user is deleted, the resource is not revoked, which means the provisioning process for the service account will not be canceled, causing the undo tasks to trigger. Instead, a task should get inserted into the provisioning process in the same way Oracle Identity Manager handles disable or enable actions. Therefore, when a user is deleted, the Service Account Alert task functionality is triggered.
- When the user gets disabled, the resource will not be disabled, which means the tasks of effect Disable should not be inserted into the provisioning process for the service account instance. Instead, the Service Account Alert task functionality is triggered and a task gets inserted into the provisioning process.

- Explicitly disabling, enabling, or revoking a service account instance either directly or through a request is managed in the same way as for regular accounts.
- Oracle Identity Manager APIs can be used to transfer a provisioned service account resource, for example a provisioning process and a process form entry, from one user to another. When this happens, the Service Account Move task functionality is triggered.

Converting Accounts

You can change a regular account to a service account or change a service account to a regular account. In either case, the Service Account Change task is inserted within the provisioning process and becomes active in the Tasks tab of the Process Definition. Any adapter associated with this provisioning process is executed. If there is no adapter, then a predefined response code is attached.

The relevant APIs for this functionality are:

- `tcUserOperations.changeFromServiceAccount`
- `tcUserOperations.changeToServiceAccount`

Service Account Alerts

When any lifecycle event occurs for the user to whom the service account is linked, the Service Account Alert task is inserted into the provisioning process of that service account instance. You can use this task to initiate the appropriate actions in response to any disabling event that occurred for the user.

An alert task is inserted when a user is disabled or deleted. In these cases, this is the only action that happens to the service account instance.

An alert task is not inserted for events directly on the service account, for example, when a service account is explicitly disabled.

Moving Service Accounts

You can transfer ownership of a service account from one user to another. When you do this, the provisioning instance shows up in the resource profile of the new owner, and no longer in the resource profile of the old user. The Service Account Moved task is inserted into the provisioning process of the resource instance after the account is moved. Any adapter associated with this provisioning process is executed. If there is no adapter, then a predefined response code is attached.

The API method for moving a Service Account is `tcUserOperationsIntf.moveServiceAccount`.

Service Account Flag APIs

The following API methods set the flags associated with service accounts:

- `tcRequestOperations.addRequestObject`
- `tcRequestOperations.setRequestObjectAsServiceAccountFlag`
- `tcUserOperations.changeFromServiceAccount`
- `tcUserOperations.changeToServiceAccount`
- `tcUserOperations.provisionObject`
- `tcUserOperations.moveServiceAccount`

- `tcObjectOperations.getServiceAccountList`

The names of the flags are indicative of the function of each flag.

7

Customizing Field Functionality

This chapter describes how to customize the functionality of the various fields that appear in the Administrative and User Console. You can customize field functionality in the following ways:

- Edit the system fields that appear on the Administrative and User Console pages.
- Add new user-defined fields.
- Control who has the ability to edit the values in various fields.

This chapter discusses the following topics:

- About the FormMetaData.xml File
- Defining a New Field
- Files to Modify
- Setting Field Configuration

About the FormMetaData.xml File

Use the `FormMetaData.xml` file to define form controls and labels in the Administrative and User Console. You can also edit how the default fields behave, for example, you can make them viewable, editable, or required. For all entries in the `FormMetaData.xml` file, the attributes section defines the fields included in the pages while the attribute references sections define the fields on a particular page or group of pages. To define a new value property for this file, you must define the attribute, and then reference it for a page.

The following table lists the forms that are depending on the `FormMetaData.xml` file.

FormMetaData.xml File Sections	
`<!-- User Self Registration and User Profile Modification section -->`	
User self-registration form	`<Form name="SelfRegistrationUserForm">`
Self-registration Approval form	`<Form name="SelfRegistrationApprovalForm">`
User profile modification form	`<Form name="ProfileModificationUserForm">`

FormMetaData.xml File Sections	
User profile modification Approval form	`<Form name="ProfileModificationApproval Form">`
`<!-- User Management section -->`	
User form	`<Form name="3">`
Group form	`<Form name="6">`
`<!-- Organization Management section -->`	
Organization form	`<Form name="2">`
`<!-- Resource Management section -->`	
Resource form	`<Form name="5">`

The following table lists the pages on which you can customize field behavior.

Page	How it is accessed
User Detail	Use the Manage link under Users, then execute a query and select a particular user account.
Create User	Use the **Create** link under Users.
User Self Registration	Use the **Create Request** link under Self-Registration.
Modify Account Profile	Use the **Account Options** link. Then click **Modify Account Profile**.
Edit User	Use the **Manage** link under Users, then execute a query and select a particular user account. Then, click the **Edit** button at the bottom of the User Detail page.
Create User Request	Use the **Track** link under Requests, then execute a query and select a particular request of type Self-Registration. Expand the User Information branch and click **Provide User Information**.
Organization Detail	Use the **Manage** link under Organizations, then execute a query and select a particular organization.
Create Organization	Use the **Create** link under Organizations.
Group Detail	Use the **Manage** link under User Groups, then execute a query and select a particular user group.
Create User Group	Use the **Create** link under User Groups.
Resource Detail	Use the **Manage** link under Resource Management, then execute a query and select a particular resource.

Note: Any pages that are not listed in the preceding table obtain their label names from the `xlWebAdmin.properties` file or the corresponding locale-specific properties files.

For each of these pages, Oracle Identity Manager is preconfigured with certain system fields. In addition, you can define the user-defined fields to be displayed on these forms. The behavior of either of these types of fields can be customized for the pages listed here. However, the procedure for customizing their behavior differs depending on whether the field is system defined field or user defined.

Note: All user-defined fields must first be defined by using the User Defined Field Definition form of the Design Console, as explained in *Oracle Identity Manager Design Console Guide*.

After you create the user-defined field within the Design Console, you must add it to the list of attributes in the section `<!-- User Self Registration and User Profile Modification section -- >` of the `FormMetaData.xml` file. Do not add it to the list in the `<! --User Management -- >` section of that file.

Ensure that your attribute entry provides values for each of the following parameters:

- `Name`: Must be same as column name for this field in the Design Console
- `Label`: Need not conform to label specified for this field in the Design Console
- `DisplayComponentType`: Must be same as the value of Field Type for this field in the Design Console
- `DataLength`: Must be same as the value of Length for this field in the Design Console
- `Map`: Must be same as the column name for this field in the Design Console

Make the entry for the user-defined field consistent with the definition of the field in the Design Console.

Defining a New Field

To define a new field for the Administrative and User Console:

1. Create the field definition by using the Design Console. Note the field definition information for the purpose of editing the `FormMetaData.xml` file.
2. Define the attribute in the `FormMetaData.xml` file.

   To define the attribute, add an entry for it by using the same information you entered when you used the Design Console to create a field definition. The format of the entry must be as follows:

   ```
   <Attribute name="identifier" label="field_label"
   displayComponentType="datatype"
             map="identifier" />
   ```

   In this format:

   - *`identifier`* is the name used to specify this field on a page.
   - *`field_label`* is the associated property name in the `xlWebAdmin.properties` file or the locale-specific file.
   - *`datatype`* is the data type of the field displayed in the Administrative and User Console.
   - `map` is the database column name specified when you create the field definition by using the Design Console. User-defined fields are prefixed with `USR_UDF_`, so if you typed in SSN in the Design Console, the value is `USR_UDF_SSN`.

For example:

```
<Attribute name="social" label="SSN" displayComponentType="TextField"
              map="USR_UDF_SSN" />
```

3. Reference the attribute in the FormMetaData.xml file in the section for the page on which you want the field to appear.

Files to Modify

To customize field behavior and functionality, you edit the FormMetaData.xml file.

Setting Field Configuration

This section explains how to configure fields in various pages. It discusses the following topics:

- For Fields in the User Detail, Create User, and Edit User Pages
- For Fields in the User Self Registration, Create User Request, and Modify Account Profile Pages
- Example: Setting the User ID Field on the Create User Request Page

For Fields in the User Detail, Create User, and Edit User Pages

Fields available for use within the Create User, User Detail, and Edit User pages are controlled by the attribute definition within the <!-- User Management section -- > of the FormMetaData.xml file.

Fields displayed on each of these pages are controlled by using the attribute references in the <!--Fields that will be displayed on the Users form section-- > section of the FormMetaData.xml file.

For each attribute reference, you can specify whether the field is:

- Viewable—By adding the attribute reference to the relevant section
- Editable—By specifying a value of TRUE or FALSE for the editable parameter
- Optional—By specifying a value of TRUE or FALSE for the optional parameter

Note: You need not add an attribute definition or attribute references for user-defined fields associated with these pages.

For Fields in the User Self Registration, Create User Request, and Modify Account Profile Pages

The fields in the User Self Registration, Create User Request, and Modify Account Profile pages are controlled by the attribute definition within the <!-- User Self Registration and User Profile Modification section -- > of the FormMetaData.xml file. The fields that appear on each of these pages are controlled by using the attribute references in the sections of the FormMetaData.xml file, as shown in the following table.

Section	Page Affected
<Form name="SelfRegistrationUserForm">	User Self Registration

Section	Page Affected
`<Form name="SelfRegistrationApprovalForm">`	Create User Request
`<Form name="ProfileModificationUserForm">`	Modify Account Profile

For each attribute reference, you can specify whether the field is:

- Viewable—By adding the attribute reference to the relevant section
- Editable—By specifying a value of `TRUE` or `FALSE` for the `editable` parameter
- Optional—By specifying a value of `TRUE` or `FALSE` for the `optional` parameter

Example: Setting the User ID Field on the Create User Request Page

This example shows how to make the system field User ID on the Create User Request page viewable, editable, and required.

1. Open the `FormMetaData.xml` file.
2. Locate the section `<!-- Definition of the form that will be displayed to the approver(s) for self registration approvals -- >`.
3. Add an attribute reference to this section to represent behavior of this field on the Create User Request page:

```
<AttributeReference editable="true" optional="false">
Users.User ID
</AttributeReference>
```

This entry causes the User ID field to be visible, editable, and required on the Create User Request page.

> **Note:** To edit the label applied to a field, edit the value of the label parameter of the applicable attribute definition.

8 Customizing Menus

Menu items are displayed as links in the left panel of the Administrative and User Console. This chapter describes how to customize the menus in the Administrative and User Console.

To customize the menus in the Administrative and User Console:

1. Identify the menu group or item that you want to customize and obtain the class name of the menu group or item by using the Form Information form in the Oracle Identity Manager Design Console.

   For example, the class name for the Request New Resources item on the My Resources menu is `My Resources.Request New Resources`. This name is stored in the database `WIN` table. The corresponding menu item shows that the menu group code is `My Resources` and the menu item code is `Request New Resources`.

2. Open the `xlWebAdmin.properties` or `xlDefaultAdmin.properties` files. Table 8–1 identifies the locations of the files for each supported application server.

Table 8–1 Location of xlWebAdmin Properties and xlDefaultAdmin.properties Files

Application Server	Location
Oracle Application Server	In a nonclustered environment, the files are available in `*OAS_HOME*\product\10.1.3.1\OracleAS_1\j2ee\home\applications\Xellerate\xlWebApp\WEB-INF\classes\`. In a clustered environment, the files are available in `*OAS_HOME*\product\10.1.3.1\OracleAS_1\j2ee\*member*\applications\Xellerate\xlWebApp\WEB-INF\classes\`.

Table 8–1 (Cont.) Location of xlWebAdmin Properties and xlDefaultAdmin.properties

Application Server	Location
JBoss Application Server	In a nonclustered environment: **1.** Extract `xlWebApp.war` from the EAR file located in *`JBOSS_HOME`*`\server\default\deploy\XellerateFull.ear`. **2.** Extract `xlWebApp.war\WEB-INF\classes\xlWebAdmin.properties` and `xlWebApp.war\WEB-INF\classes\xlDefaultAdmin.properties` from the WAR file. **3.** Edit the `xlWebAdmin.properties` file or `xlDefaultAdmin.properties` file. **4.** Repackage the WAR file. **5.** Repackage the EAR file. In a clustered environment, extract `xlWebApp.war` from the EAR file located in *`JBOSS_HOME`*`\server\all\deploy\XellerateFull.ear`, and then perform steps 2 through 6. **Note:** JBoss Application Server clustered environments are not supported in Oracle Identity Manager release 9.1.0. See "Certified Components" in *Oracle Identity Manager Release Notes* for information about certified components.
IBM WebSphere Application Server	Files are available in *`WAS_HOME`*`\AppServer\profiles\profile_name\installedApps\`*`node_name`*`\XellerateFull.ear\xlWebApp.war\WEB-INF\classes`.
BEA WebLogic Server	In a nonclustered environment, the files are available in *`BEA_HOME`*`\user_projects\domains\`*`domain_name`*`\XLApplications\WLXellerateFull.ear\xlWebApp.war`. In a clustered environment: **Admin Server:** The files are available in `BEA_HOME\user_projects\domains\xldomain\XLApplications\WLXellerateFull.ear\xlWebApp.war\WEB-INF\classes`. **Managed Server:** The files are available in *`WL_HOME`*`\weblogic81\common\nodemanager\`*`MANGED_SERVER_NAME`*`\stage\Xellerate\xlWebApp.war\WEB-INF\classes\`. For the manager server, the files must be modified on all the nodes.

3. To modify a menu group, locate in the `xlWebAdmin.properties` file the property name for the menu group that you want to customize. Each menu group has a single property associated with it, which represents the display label.

   The property name for the menu group display label has the form `menuGroup.menu group code` with the spaces replaced by hyphens (-). For example, for a menu group with code `My Resources`, the property name is `menuGroup.My-Resources`. Modify the value assigned to the property to the label you want to display.

   Each menu item has two properties associated with it: the menu item display label property and the menu item link property, which identifies the target page that is displayed when a menu item is selected. The menu item display label property is defined in the `xlWebAdmin.properties` file and the menu item link property is defined in the `xlDefaultAdmin.properties` file.

   The property name for a menu item display label has the form `menuItem.menu group code.menu item code` with the spaces replaced by hyphens. This means that a menu item with the code `Request-New-Resources` under a menu group with the code `My-Resources` has a property name of `menuItem.My-Resources.Request-New-Resources`. The menu item link property is in the form `menuItem.menu group code.menu item code.link`.

4. To create new menu items or groups, you first create the correct entries in the system by using the Form Information form in the Oracle Identity Manager Design Console. Next, you add the corresponding new properties mentioned in step 3. For each new menu group, one new property is required, and for each new menu item, two new properties are needed.

   For instance, you want to add a new menu group with the code My Links and two new menu items under it, with codes Favorites and Non-Work Links. You must create two new entries in the system by using the Form Information form in the Design Console with the class names `My Links.Favorites` and `My Links.Non-Work Links`.

   > **Note:** You can specify any class names you want, and the labels that are displayed do not depend on these names.

5. Add the following properties to the `xlWebAdmin.properties` file by using the class names specified in step 4:

```
menuGroup.My-Links=My Links
menuGroup.My-Links.Favorites=Favorites
menuGroup.My-Links.Non-Work-Links=Non-Work Links
```

6. Add the following properties to the `xlDefaultAdmin.properties` file by using the class names specified in step 4:

```
menuGroup.My-Links.Favorites.link=favorites.do?myfavorites
menuGroup.My-Links Non-Work-Links.link=nonworklinks.do?nonworklinks
```

> **Note:** To change the menu group or item names, only the `xlWebAdmin.properties` and `xlDefaultAdmin.properties` files must be changed. No change is required in the database or code.

9 Customizing Group Menu Item Access

This chapter describes how to specify which Administrative and User Console menu items are available to a particular group.

This chapter discusses the following topics:

- Setting Menu Items for Groups
- Creating New Menu Items for Groups
- Assigning a Menu Item to a Group
- Removing a Menu Item from a User Group
- Renaming a Menu Item

Setting Menu Items for Groups

You can set the menu items for groups that you manage by using the group menu items page. The menu items listed in the group menu items page are available to that group. You can add items to the page and remove items from it. If a user is a member of multiple groups, then they have access to all of the menus associated with each of the groups to which they belong.

> **Note:** To ensure that all users of the Administrative and User Console have access to at least the most basic menu items, assign those menu items to the `All Users` group.

For example, if User A is a member of two groups, `ABC` and `XYZ`, and members of the `ABC` group are entitled to access `Requests.Create` and members of `XYZ` are entitled to access `Requests.Track`, then User A must be able to access both the Create and Track pages, which are under Requests, of the Administrative and User Console.

Each menu group has the `menuGroup.menu group code` property associated with it. Each menu item has the following associated properties:

- `menuItem.menu group code.menu item code`: Provides the display label of the menu item
- `menuItem.menu group code.menu item code.link`: Defines the link destination

Creating New Menu Items for Groups

To create new menu items for groups, you add entries by using the Form Information form. Then you add the corresponding properties.

Table 9–1 presents the complete list of Administrative and User Console menu items to which a user group can be granted access.

Table 9–1 Administrative and User Console Menu Items to Which Groups Can Be Added

Menu Group	Menu Item Name	Function
My Account	**Account Profile**	Allows users to access their account profiles.
	Change Password	Allows users to change their passwords.
	Challenge Q&A	Allows users to change their challenge questions.
	My Proxy	Allows users to view their proxy details.
My Resources	**My Resources**	Allows users to view their provisioned resources.
	My Requests	Allows users to view their requests (raised for them or by them).
	Request New Resources	Allows users to initialize and/or submit requests to provision resources to themselves.
Requests	**Resources**	Allows users to make requests.
	Track	Allows users to track existing requests.
To-Do List	**Pending Approvals**	Allows users approve tasks within existing requests.
	Open Tasks	Allows users to view open tasks assigned to them, or to users they manage.
	Attestation	Allows users to display all open attestation tasks that are assigned to them, and certify, reject, decline, or delegate attestation tasks.
Users	**Create**	Allows users to create user accounts.
	Manage	Allows users to view, edit, and otherwise manage users they administer.
Organizations	**Create**	Allows users to create organizations.
	Manage	Allows users to manage organizations they administer.
User Groups	**Create**	Allows users to create groups.
	Manage	Allows users to view, edit, and otherwise manage groups they administer.
Access Policies	**Create**	Allows users to create access policies.
	Manage	Allows users to view, edit, and otherwise manage access policies they administer.
Resource Management	**Manage**	Allows users to view, edit, and otherwise manage resources they administer.
	Create IT Resource	Allows users to create IT resources and set access permissions to user groups on the IT resource.

Table 9–1 (Cont.) Administrative and User Console Menu Items to Which Groups Can Be

Menu Group	Menu Item Name	Function
	Manage IT Resource	Allows users to view, modify, and delete IT resources.
	Create Scheduled Task	Allows users to create scheduled tasks by specifying the schedule and adding scheduled task attributes.
	Manage Scheduled Task	Allows users to view and modify scheduled tasks.
Deployment Management	**Export**	Allows users to export Xellerate entities as XML files.
	Import	Allows users to import Xellerate entities from XML files created by using the Deployment Manager Export feature.
	Install Connector	Allows users to install predefined connectors, and automate copying connector files to the specified installation directory, importing the connector XML files, and compiling adapters.
Reports	**Operational Reports**	Allows users to view a list of available Operational Reports and run them.
	Historical Reports	Allows users to view a list of available Historical Reports and run them.
Generic Technology Connector	**Create**	Allows users to create generic technology connectors
	Manage	Allows users to view and edit generic technology connectors
Attestation	**Create**	Allows users to create an attestation process.
	Manage	Allows users to view, edit and otherwise manage the attestation processes.
	Dashboard	Allows users to view active attestation processes.

Assigning a Menu Item to a Group

To assign a menu item to a user group, perform the following steps:

1. Log in to the Administrative and User Console as an administrator.
2. Click **User Groups**, and then click **Manage**.
3. Search to find the group to which you want to assign a menu item.
4. Select the group name.
5. From the additional details menu, select **Menu Items**.
6. Click **Assign Menu Items**. The Assign Menu Items page is displayed.
7. From the Assign Menu Items page, select the **Assign** option for any menu item that you want to assign to the selected group.

> **Note:** If there are more items than are displayed in the table, then click **Next** to view the rest of the items.

8. Click **Assign.**
9. Click **Cancel** to return to the Assign Menu Items page.

   or

   Click **Confirm Assign**.

The items are added to the group users, which they will be able to access the next time they log in.

Removing a Menu Item from a User Group

To remove a menu item from a user group, perform the following steps:

1. Log in to the Administrative and User Console as an administrator.
2. Click **User Groups**, then click **Manage**.
3. Search to find the group from which you want to remove the menu item.
4. Click the group name.
5. From the **Additional Details** menu, select **Menu Items.**
6. Select the **Delete** option for each of the desired menu items.
7. Highlight the menu item you wish to prevent the user group from accessing.
8. Click **Delete**.
9. Click **Confirm Delete**.

Renaming a Menu Item

To rename a menu item, perform the following steps:

> **Note:** To change the menu group or item names, only the `xlWebAdmin.properties` file or locale-specific file, such as the `xlWebAdmin_jp.properties` file for Japanese, must be changed. No change is required in the database or code.

1. Identify the group item name that must be changed.
2. Get the class name of that menu item from the system by using the Form Information form in the client.

   For example, for the menu item Request New Resources from My Resources, the class name is `My Resources.Request New Resources`. This name is in the database in the `WIN` table. This menu item shows that the menu group code is My Resources and the menu item code is Request New Resources.
3. Record the menu group and menu item codes, and open the `xlWebAdmin.properties` file.
4. Search for the property name in the `xlWebAdmin.properties` file.

5. Change the right hand side value of this property to the new label. This changes the name of the menu group.

   In the `xlWebAdmin.properties` file, the property name for a menu item display label has the form `menuItem.menu group code.menu item code` with the spaces replaced by hyphens. For menu item with code Request New Resources under menu group with code My Resources, the property name is `menuItem.My-Resources.Request-New-Resources`.

6. Locate the property name in the `xlWebAdmin.properties` file.

7. Change the right hand side value of this property to the new label. This changes the name of the menu item.

10

Customizing Search Pages

In the Administrative and User Console, when you click a menu item to perform tasks, such as managing users and groups, a search page is displayed. For example, when you click the **Manage** link under the **Users** menu item, the Manage Users page is displayed with four drop-down menus for searching users. You can customize the number of drop-down menus, and what the items in the drop-down menus are.

When the search results display, you can determine the maximum number of rows in the results table displayed on each page. After a user selects an item from the results table, a detail page is displayed such as the User Detail page or the Group Detail page. The detail page contains an additional details menu. You can customize the items in these menus.

Customizing Drop-Down Menu Item Content

Use the Design Console to change the lookup codes for search pages and additional details. To customize drop-down menus:

1. Log in to the Design Console.
2. Open the Lookup Definition form by navigating to Administration, then to Lookup Definition.
3. Search to locate the desired lookup definition.

   Tip: For your search criteria, use **lookup.webclient*** search to find the search pages, or ***additional_details** to find the additional details.

4. Make the desired changes to the lookup codes to set the options displayed in the drop-down menu for each search page.
   - The Code Key is the meta data for each column.
   - The Decode value is what is displayed in the Administrative and User Console.
   - The order the items appear in the Code Key list are the order they appear in the Administrative and User Console drop-down list. If you delete an entry and add it back, it is displayed last in the list.
5. Save your changes.

Customizing the Number of Drop-down Menu Items and Search Results

To change the number of drop-down menus, and the maximum number of search results on each page, edit the `xlDefaultAdmin.properties` file.

See Also: The "Accessing Administrative and User Console Customization Files" section on page 1-1 for details on how to access the `xlDefaultAdmin.properties` file and how to redeploy the EAR file.

To set the number of drop-down menus:

1. Open the `xlDefaultAdmin.properties` file.
2. Locate the property from the following table, and edit it as required.

Table 10–1 Properties that Determine the Number of Menus on a Search Page

Property Name	Default	Page
`global.property.numsearchuserfields`	4	Search Users
`global.property.numsearchaccesspolicyfields`	2	Access Policies
`global.property.numsearchresourcefields`	3	Search Resources
`global.property.numsearchgroupfields`	1	Search groups
`global.property.numsearchopentaskfields`	2	My Open Tasks
`global.property.numsearchattestationprocessfields`	3	Attestation Process

3. To change the maximum number of search results on each page, change the value of the property `global.displayrecordNum.value` to the desired value. The default value is 10.
4. Save the file.
5. Redeploy the EAR file.

A

Customizing the Workflow Designer

There are a number of properties that can be useful in customizing the Workflow Designer user interface based on what a particular locale demands. This appendix discusses some of the examples of Workflow Designer customization.

The label widths for the task names can be customized by using:

```
global.workflowRenderer.labelWidthFactor=7
```

If in a particular locale the text is seen to be truncated, then this property can be changed accordingly to modify the width of the labels.

Similarly, the width and height of the icons representing the tasks and responses can be controlled by:

```
workflowRenderer.referenceMarker.defaultHeight=22
workflowRenderer.referenceMarker.defaultWidth=22
workflowRenderer.referenceMarker.defaultDistance=6
workflowRenderer.referenceMarker.maxLabelLength=5

workflowRenderer.response.defaultHeight=20
workflowRenderer.response.defaultWidth=250
workflowRenderer.response.maxLabelLength=25

workflowRenderer.task.defaultHeight=38
workflowRenderer.task.defaultWidth=38
workflowRenderer.task.maxLabelLength=200
```

In the search functionality at different places in the designer implementation, the number of results could be high in cases where users and groups are searched. To control the number of results to be displayed, the following properties can be used.

```
workflowDesigner.label.limitUserSearchResults=200
workflowDesigner.label.limitGroupSearchResults=200
workflowDesigner.label.limitAdapterSearchResults=200
workflowDesigner.label.limitEmailTemplateSearchResults=200
workflowDesigner.label.limitRulesSearchResults=200
workflowDesigner.label.limitAssignTypeSearchResults=200
workflowDesigner.label.limitDependentDataTaskSearchResults=200
workflowDesigner.label.limitExistingTasksSearchResults=200
```

When the number of results returned are more than these numbers, a message is shown to narrow the search criteria.

Index