CA Identity Manager

User Console Design Guide

r12.5 SP11

CA Technologies Product References

This document references the following CA products:

- Identity Manager
- CA SiteMinder®
- CA Directory
- CA User Activity Reporting Module (UARM)
- CA Role & Compliance Manager

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

If you would like to provide feedback about CA Technologies product documentation, complete our short customer survey, which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 1: Customizing the User Console

This section contains the following topics:

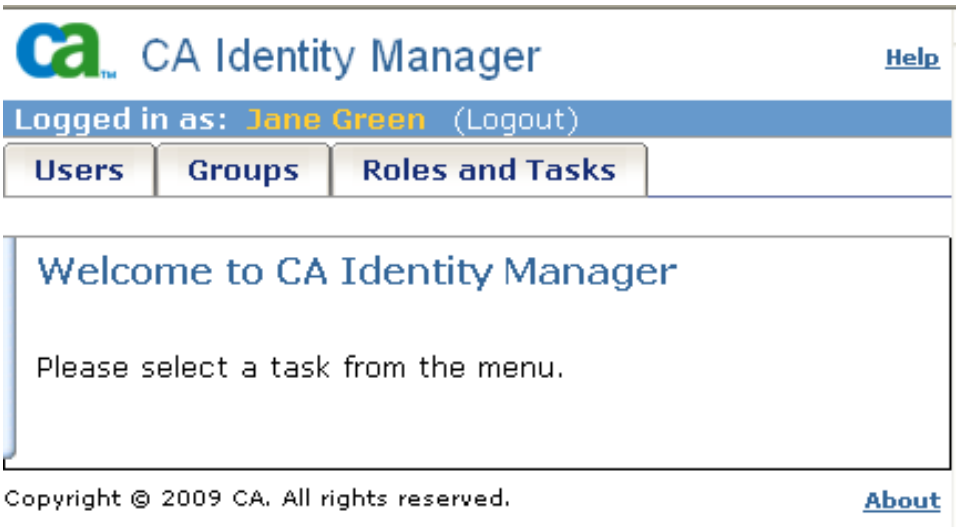
Default User Console

When you create Identity Manager environment, Identity Manager creates a default User Console that you use to manage the environment. The User Console includes a set of default tasks and roles.

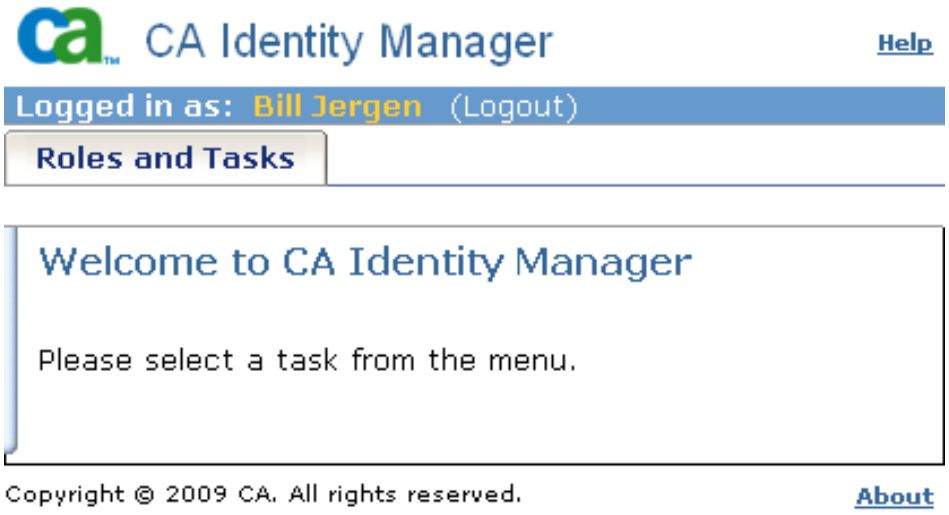
- Tasks are actions performed by Identity Manager users. There are two types of tasks:
  - Admin tasks, which you use to manage users, organizations, groups, roles, and tasks.
  - External tasks, which perform functions in business applications, such as passing user attributes to a reporting application
- Roles associate users and privileges to Identity Manager or other applications. Roles are made up of tasks. A user who has a role can perform its tasks. Users may have multiple roles. For example, a user may have the roles accountant and employee.

  Admin roles are made up of admin tasks.

The tasks that you see when you log into Identity Manager environment depend on your admin roles. In the following example, the user Jane Green has the User Manager role. She sees tabs for the admin tasks that are available for User Managers.



In this example, Bill Jergen has the Role Manager role. When he logs into the User Console, he sees only the Roles and Tasks tab that includes the tasks that he can use.



Note: For more information about tasks and roles, see the *Administration Guide*.

Tasks, Tabs, and Screens

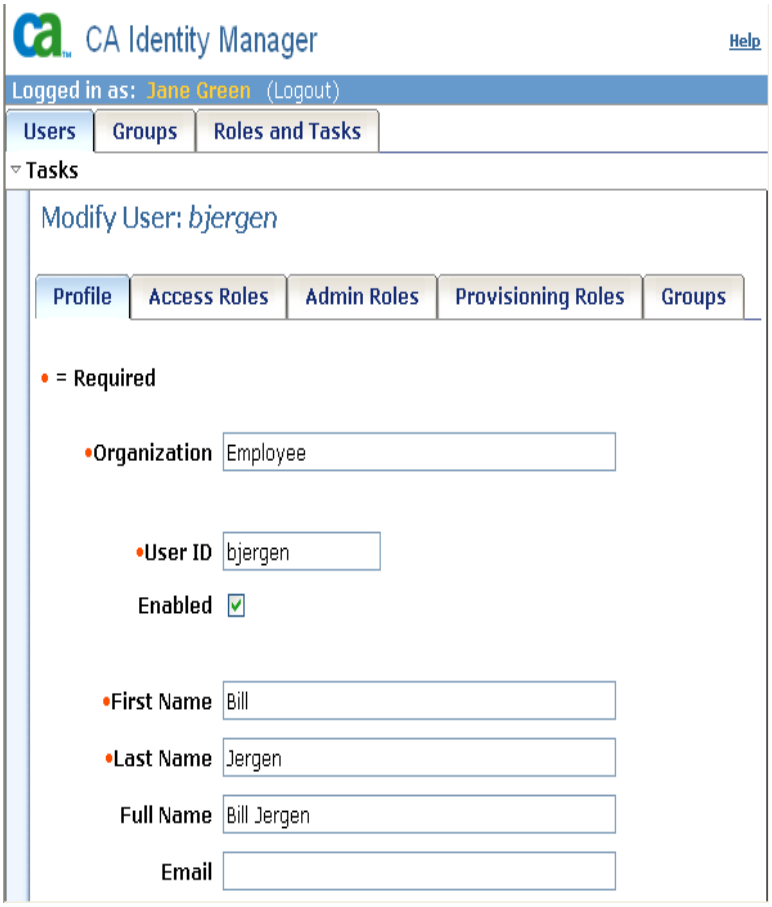
An admin task is an administrative function performed by Identity Manager users. It is comprised of *tabs*, which logically group a set of fields or functionality. For example, the default Modify User task includes the following tabs:

- Profile
- Access Roles
- Admin Roles
- Provisioning Roles
- Groups

When administrators use this task, they select the appropriate tab to enter profile information, manage roles, or manage group membership.

A tab may be associated with multiple tasks.

The following example shows an admin task with multiple tabs.



Tabs may be associated with a configurable *screen*, which determines the appearance and content of the tab. To change a default tab, you can modify the screen that is associated with the tab, or create a new screen.

A screen may be associated with multiple tabs.

More information:

List of Default Tabs (see page 181)

User Console Customizations

Typically, after creating Identity Manager environment, a system administrator performs some initial configuration to ensure that the environment addresses existing business needs. Customizing the User Console also improves usability by creating tasks to match user workflows, increases security by ensuring that users can only access the fields they need, and improves performance.

You can customize the following elements in the User Console:

- Task navigation—Determines how administrators access tasks, and how they access different tabs in those tasks.
- Tabs and screens—Controls the fields that appear on a tab and how those fields are displayed.

  The admin tasks in the default User Console are created based on the information in the directory configuration file (directory.xml), which defines the objects and attributes that Identity Manager manages. For example, the Profile tab for the default Create User task includes all of the attributes that are defined in the directory.xml file for the user object.

  Most users need to manage only a subset of attributes for any object.
- Self-service tasks—Determines how self-service tasks, such as the Forgotten Password or Forgotten User ID tasks function.
- Branding—Displays corporate logos and colors in the User Console.
- Localization—Displays the User Console in different languages.
- Custom Online Help—Allows you to provide online help that is specific to a task or tab that you customize.

Chapter 2: Task Navigation

This section contains the following topics:

Task-based Navigation

To perform an action in Identity Manager, you select a task and an object on which to perform that task. For example, when modifying a user profile, the task is Modify User and the object is the user profile that you want to modify.

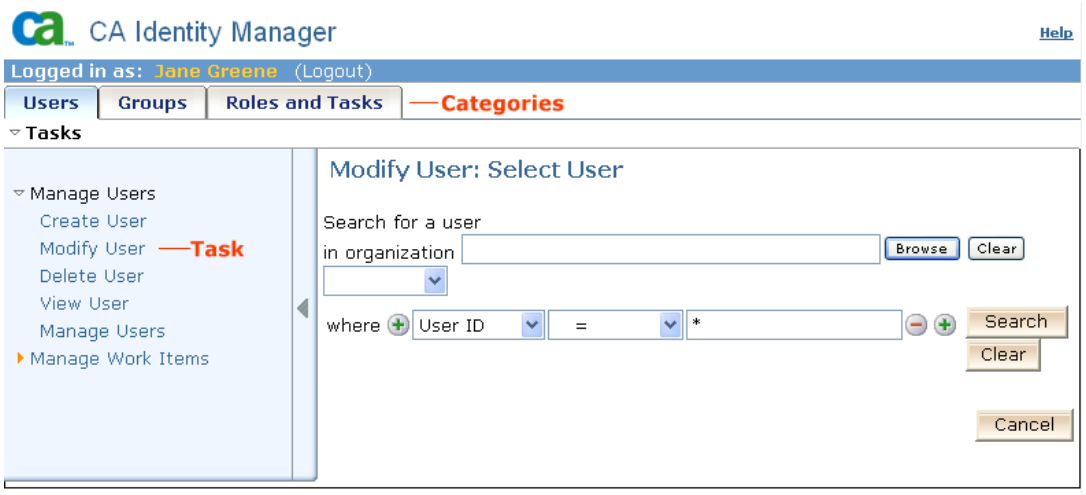
Identity Manager provides two methods for selecting tasks and objects:

- Task-based navigation
- Object-based navigation (see page 16)

In task-based navigation, you select a category and task, and then search for the object to which the task applies.

For example, to modify a user profile, you select the Users category, and then select the Modify User task. You then search for the user to modify.

The following example illustrates categories and tasks in the User Console.



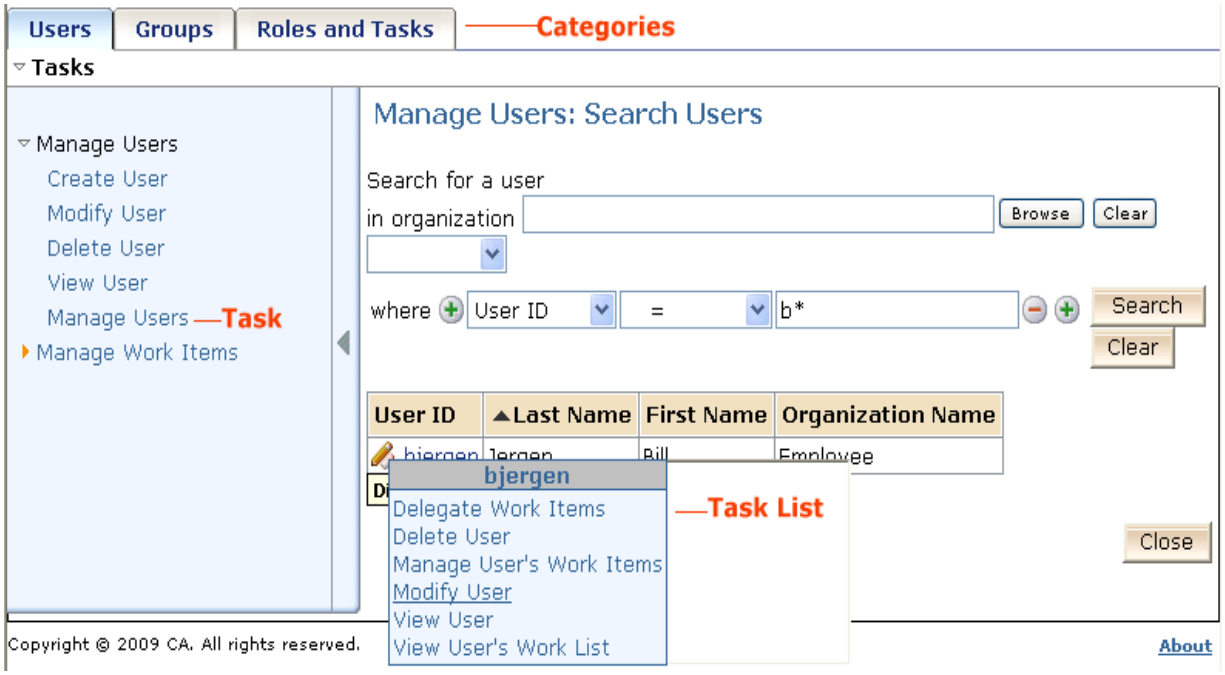
Task-based navigation is the default navigation method. Use task-based navigation when users are more likely to perform a single action on an object.

Object-based Navigation

Object-based navigation is a method that allows users to select an object and view all of the tasks that they can perform on that object in a pop-up menu. From the menu, the user can select the task that they want to use. Once the task is complete, users can select another task from the pop-up menu without having to search for the object again.

For example, to modify a user using this method, you select the Users category, then select the Manage Users task. You search for and select the user that you want to manage. In the search results, you click an icon to see a list of tasks that you can use to manage the selected user. From that list, you can select Modify User or any other appropriate task.

The following example illustrates a pop-up task menu.



Consider implementing object-based navigation when users perform multiple actions on a single object.

Identity Manager includes the following default admin tasks that are configured for object-based navigation:

- Manage Users
- Manage Groups

- Manage Organizations
- Manage Admin Roles / Manage Admin Tasks
- Manage Access Roles
- Manage Provisioning Roles

You can also add pop-up task menus to list and search results screens to enable object-based navigation in existing tasks. For example, you can add object-based navigation to the Modify Admin Role Members task to display a pop-up task menu for each role member. Administrators can use the task menus to manage role members without having to perform a new search for each role member.

Configure Object-based Navigation

Tasks that are configured for object-based navigation include only a search screen. Users search for an object to manage, and then use pop-up task menus to view all the tasks that they can perform on that object.

When you configure object-based navigation, note the following:

- The action for the admin task on the Profile tab must be Search.
- The admin task cannot contain tabs.
- If you want to configure all admin tasks to use object-based navigation, add Create and Delete buttons to the search screen to support these operations. The Create and Delete actions are not supported in the popup task menu.

To configure object-task navigation

1. Complete *one* of the following steps:
   - Select Modify Admin Task from Roles and Tasks, Admin Tasks. Search for and select the admin task to modify.
   - Select Create Admin Task from Roles and Tasks, Admin Tasks. Then, select Create a copy of an Admin Task and search for a task to copy.

   Note: To simplify configuration, consider creating a copy of an existing Manage task, such as Manage User. The default Manage tasks include the configuration settings required for object-task navigation.

   Identity Manager displays the tabs to configure for the task you selected.
2. Configure the settings for the Profile tab as needed. Set the Action for the task to Search.
3. Select the Search tab and click Browse to configure the search screen for the task.

   Identity Manager displays a list of search screens that you can apply to this task.

4. Select the search screen that you need.

   Note: To simplify configuration, consider creating a copy of an existing Manage search screen definition with the same object type, such as Manage Users Search. The default Manage search screens are configured to support object-task navigation.

5. Complete the fields in the search screen configuration screen as needed.

   Note: If you do not want to include separate tasks in the menus for creating an object or deleting multiple objects, you can configure the search screen to have buttons to launch these tasks. You can then hide those tasks in the menus.

Task Categories

Task categories allow you to organize tasks to make them easier to locate and search for in the User Console.

You can specify three task categories:

- Category 1 is the top level category for tasks. These categories are displayed as tabs across the top of the User Console.

  Note: If an administrator has an active task open and then selects a new category 1 tab, Identity Manager cancels the active task. If the user has made changes to that task, Identity Manager displays a warning message. The administrator can choose to complete the current task, or cancel the current task and move to the new category.

- Category 2 is a second level category. This category enables you to group related tasks in a top-level category. If you do not specify a second level category, the task is displayed as a link below the category 1 tabs.

- Category 3 groups tasks in the left navigation pane. When administrators click the Category 3 name in the User Console, a list of tasks in that category is displayed.

  If no category 3 name is specified, the tasks appear in the left navigation pane without a heading.

Within each category, you can control the order in which the items in that category are displayed by specifying a category order. For example, in the following illustration, the Employee tab has a category order of 3.

Note: When a category contains multiple tasks, the category order that is specified in the profile for each task must be the same. If the category order is different, multiple instances of that category tab will appear. For example, the Employee category contains two tasks: Create Employee and Modify Employee. If the category order in the Create Employee task is 3 and the category order in the Modify Employee is 6, the Employee category appears as two tabs.

Disable Automatic Task Cancellation

In the User Console, when a user selects a new task category tab, Identity Manager cancels the active task in the task pane. If the user has made changes to the active task, a message is displayed asking the user to confirm the cancellation. For example, if a user makes changes to information using the Modify User task, and then attempts to access the Home tab before submitting the Modify User changes, Identity Manager informs the user that the task will be cancelled and prompts the user for confirmation.

You can configure Identity Manager to allow administrators to select a new task category tab without cancelling the active task or displaying a confirmation message. In this case, selecting the new category tab displays the menu of tasks for that category in the left navigation pane, but leaves the active task in the task pane. When the user selects a new task, the active task is cancelled without notification.

To change the default behavior so that Identity Manager does not cancel the task before switching to a new tab, add a user-defined property in the Management Console.

To change the default behavior

1. Open the Management Console.
2. Select Environments, and then select the environment that you want to modify.

   The Environment Properties page opens.

3. Select Advanced Settings, Miscellaneous.
4. Enter the following values and click Add:
   - Property: ConsoleDisableAutoTaskCancel
   - Value: true
5. Click Save.
6. Restart the environment.

Task Flow

In Identity Manager, an admin task consists of one or more tabs that represent a logical grouping of functionality. For example, the Modify User task may include a Profile tab, Admin Roles tab, and a Groups tab. *Task flow* determines how users move from one tab to another while using the admin task.

Identity Manager provides three task flow options:

- Independent tabs—Users can use the tabs in any order.
- Wizards—Users are guided through the tabs by a wizard interface.
- Sequences—Users complete one tab in the task, and then Identity Manager automatically opens the next tab.

  The sequence tab flow option supports dynamic page flows using customized logic (see page 22).

The task flow is determined by the tab controller. You specify the tab controller on the Tabs tab when you create or modify an admin task.

Configure Independent Task Tabs

The tabs in the default admin tasks are independent of the other tabs in the task. Users can use the tabs in the task in any order. They do not need to complete each tab before submitting the task.

This tab configuration, shown in the following example, is created by the standard tab controller.

To configure the standard tab controller

1. In the Identity Manager User Console, select Roles and Tasks, Admin Tasks, Modify Admin Task.
2. Search for and select the admin task to modify.

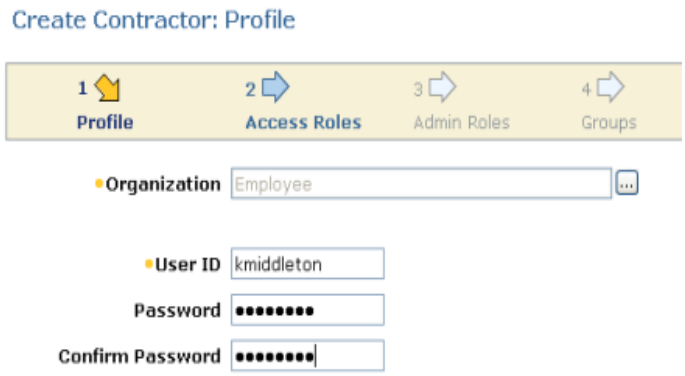
   Identity Manager displays the tabs for modifying an admin task.
3. Select the Tabs tab.
4. Select the Standard Tab Controller from the list box.
5. Click Submit.

   Identity Manager saves the changes to the task.

Configure the Task as a Wizard

Using the Wizard tab controller, you can configure a task as a wizard. In this tab configuration, administrators use each tab in a specified order. When administrators complete one tab, they click the Next button to move to the next tab in the list. A display at the top of the wizard indicates the administrator's progress, and allows that administrator to return to previously visited screens.

The following example shows a custom task, Create Contractor, displayed as a wizard.



To configure the wizard tab controller

1. In the User Console, select Roles and Tasks, Admin Tasks, Modify Admin Task.
2. Search for and select the admin task to modify.

   Identity Manager displays the tabs to configure the task you selected.
3. Select the Tabs tab.
4. Select the Wizard Tab Controller from the list box.
5. Click Submit.

Identity Manager saves the changes to the task.

Configure a Tab Sequence

When a task is configured as a tab sequence, Identity Manager displays one tab as a single page at a time. Users complete one tab and then click a custom button or link to move to the next tab.

The sequence of tabs, and the buttons and links that are displayed are determined programmatically by JavaScript that you write when you configure the sequence tab controller.

In the custom JavaScript, you can specify the appearance and order of tabs based on user input. For example, if a user selects an option on the first tab, Identity Manager displays one page. If a user selects a different option, a different page is displayed.

To configure the sequence tab controller

1. In the User Console, select Roles and Tasks, Admin Tasks, Modify Admin Task.
2. Search for and select the admin task to modify.

   Identity Manager displays the tabs to configure the task you selected.
3. Select the Tabs tab.
4. Select the Sequence Tab Controller from the list box.
5. Click Submit.

Identity Manager saves the changes to the task.

Sample Javascript for Tab Controllers

Identity Manager includes sample JavaScript files for Tab Display JavaScript and Active Tab JavaScript.

These files are installed in the samples\WizardSequencerScripts directory where the Administrative tools are installed. The Administrative Tools are placed in the following default locations:

- **Windows:** C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools
- **UNIX:** /opt/CA/IdentityManager/IAM_Suite/Identity_Manager/tools

Chapter 3: Configuring Profile Tabs and Screens

This section contains the following topics:

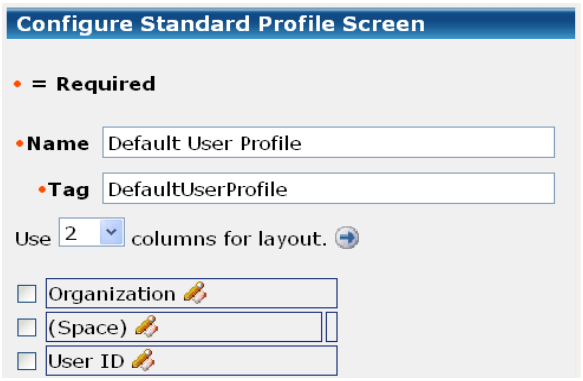
Profile Tabs and Profile Screens

For most tasks, you include a Profile tab, which shows you the attributes for the primary object of the task. The *primary object* is the object to be created, viewed, or modified by the task. For example, in the Modify User task, the primary object is a user. The Profile tab includes user attributes, such as User ID and Last Name.

When you configure a profile tab, you define basic characteristics of that tab and you specify a profile screen. The *profile screen* is the user-visible part of the tab. It controls which attributes of the primary object appear on the tab and their display properties.

Note: You can use the same profile screen on the Profile tab of several tasks.

When you design a Profile screen, you select the fields that apply to that screen. The fields may correspond to profile attributes. For example, the value entered in the User ID field of the Create User task is stored in a user profile attribute.



More Information:

Field Properties on a Profile Screen (see page 30)
Add a History Display Field (see page 75)

Profile Screen Customizations

A profile screen is comprised of fields, which collect and display attribute values. For example, the profile screen for user objects contains fields such as First Name, Last Name, and Email address. A profile screen may also include the following optional components:

- Page separators
- Images
- Attached files
- History fields
- Custom online help text
- Links to nested tasks

When you create Identity Manager environment, Identity Manager creates default profile screens that contain fields for all the attributes specified for that object in the directory configuration file (directory.xml). System administrators should customize the default profile screens to ensure that they meet business requirements, and provide the best Identity Manager performance.

Note: For information about Identity Manager performance, see the *Implementation Guide*.

System administrators can customize the default profile screens by doing the following:

- Determining the fields that appear on the profile screen
- Specifying the style of the fields
- Defining field values for list boxes and other field styles
- Adding page separators to simplify the display
- Adding pictures
- Attaching files
- Adding links to other tasks
- Adding custom online help text

Modify a Profile Screen

You can modify an existing profile screen to:

- Add or remove fields
- Change the layout of fields
- Edit field properties
- Add help text by creating separators that are of type HTML anywhere on the profile screen

To modify a profile screen

1. In the User Console, select Roles and Tasks, Admin Tasks, Modify Admin Task.
2. Search for and select the admin task to modify.

   Identity Manager displays the tabs to configure for the task you selected.
3. Select the Tabs tab, then select the Profile tab.
4. Click Browse next to the Screen field.

   Identity Manager displays a list of existing profile screens.
5. Select the profile screen that you want to modify or copy and then click one of the following buttons:

   - **Select**

     Adds the selected screen to the tab you are configuring
   - **Edit**

     Opens a new screen where you can change the settings, including fields, field properties, and layout for the selected screen

- **Delete**

  Deletes the selected screen

- **New**

  Opens a new screen where you can create a screen. The new screen does not include any default fields.

- **Copy**

  Creates a new screen using the settings from an existing screen. To create a screen which is based on an existing screen, you add a new name and tag to the screen, and modify the settings as needed.

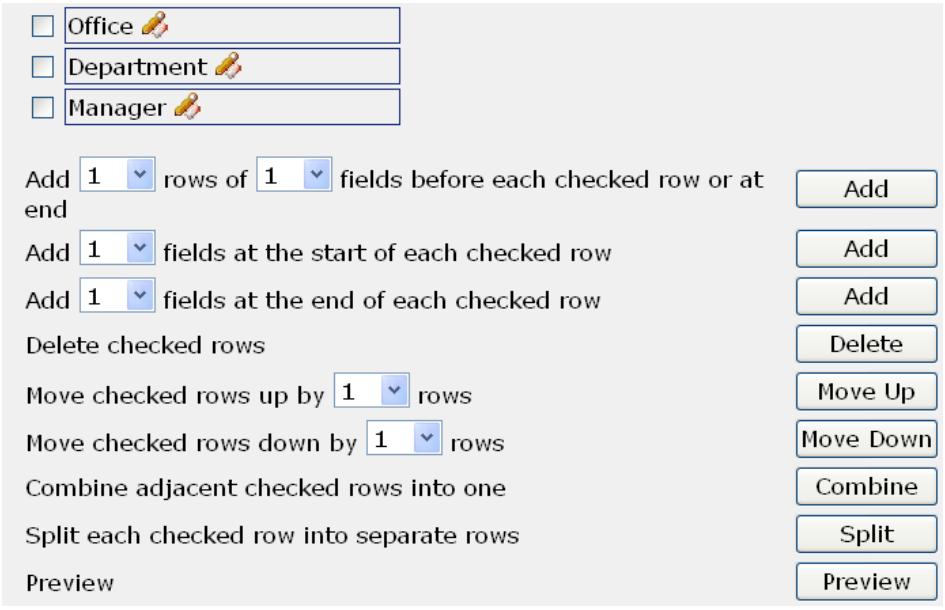
- **Cancel**

  Returns you to the Tab configuration screen.

If you selected Edit, New, or Copy, Identity Manager opens a new screen where you can create or modify a profile screen.

Add or Remove Fields

The default profile screens include all attributes for the object in the order used in the directory configuration (directory.xml). You can add or remove fields, or rearrange the order of fields by using the controls at the bottom of the list of attributes in the profile screen definition.



To add, remove, or rearrange fields

1. Modify a profile screen (see page 27).
2. Select the fields that you want to modify, then select the action to perform.

   Note: When you edit the field properties on the profile screen, you may find it easier to manipulate rows of multiple fields as follows:

   a. Click Split Each Checked Row into Separate Rows to place the fields on separate rows.

   b. Manipulate the fields on separate rows.

   c. Click Combine Adjacent Checked Rows into One to return the fields to a single row.
3. If you select an action that requires a value, select a value as needed.

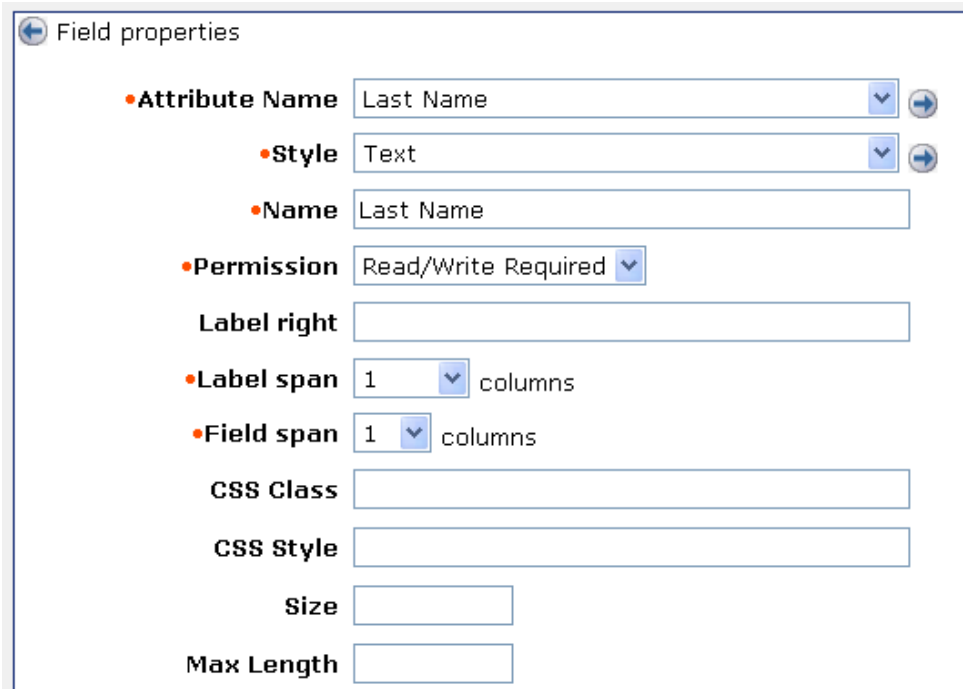
   For example, if you want to move selected rows up by three rows, select 3 from the list box in Move Checked Rows Up by 1 Row.
4. Click Preview to view the changes you made.

   Identity Manager opens a new window and displays the changes you made.
5. Click OK, then click Select to return to the Modify Admin Task task.

Field Properties on a Profile Screen

You select a field to edit its properties. Each field style has different properties, which define the display, permissions, and defaults for that property.

The following example shows the field properties that appear when you edit the Last Name field.



Note: The properties that are displayed in the Field Properties screen are determined by the option you select in the Style field. You may not see all of the properties described in this list.

You can set the following properties (in alphabetical order) for a field:

- **Attribute Name**

  Specifies the name of the object's attribute.

- **Available Values Label**

  Sets the text that appears above the list box that contains the items that are available for selection in the option selector.

- **Checked Value**

  Specifies the value of a field when its checkbox is selected. For example, the checked value for the Enabled field is true.

  The default value is true.

  Note: This field is visible when the checkbox style is selected.

- **Columns**

  The number of characters wide that the text area will be.

  Note: This field is available only when you select the Text Area style.

- **CSS Class**

  Specifies the Cascading Style Sheet class that controls the presentation of this field.

- **CSS Style**

  Specifies the Cascading Style Sheet rules that control the presentation of this field.

  You can use this field to set the width of a field. For example, to set the width of a field with the Drop Down style to 300 pixels, you specify the following in the CSS Style field:

  Note: By default, the width of fields that include a list of values, such as a drop down or multi-selector box, is set to auto ({width:auto}). This setting sizes the field to accommodate the largest value in the field. For example, if the largest value in the City option selector field is San Francisco, the option list is sized to display the entire value.

  You can also use the CSS Style field to control other display properties, such as text size and background color.

- **Current Values Label**

  Sets the text that appears above the list box that contains selected items in the object selector.

- **Date Display Pattern**

  Determines the format of dates displayed in a field and in the Date Picker control.

- **Date Storage Pattern**

  Determines how Identity Manager stores dates in user stores.

- **Default**

  Indicates the value that is displayed by default, and that is stored in the profile if no other value is provided.

  Note: For a checkbox, enter true to make the default enabled; enter false to make the default disabled.

- **Default JavaScript**

  Enables you to use JavaScript to set the default value for a field. Using JavaScript, you can set the value dynamically. For example, you can set a default value based on other attributes.

  Use this field for Create tasks only.

- **Disable AutoComplete**

  Disables the AutoComplete feature in Internet Explorer. If this check box is selected, Internet Explorer does not attempt to provide suggestions for field values based on previous entries.

  For more information on the AutoComplete feature, see the documentation for Internet Explorer.

- **Initialization JavaScript**

  Enables you to use JavaScript to set the default value for a field. Use this field for any task type.

- **Note:** The JavaScript in this field executes after the JavaScript in the Default JavaScript field for Create tasks.

- **Field span**

  Specifies the number of columns that the field will span (excluding the label)

- **Field to Match Against**

  Specifies the field that Identity Manager checks for a matching value. You can use this feature to verify that the value of two fields on a screen match. This is typically used to verify that a password or other critical information is entered correctly. For example, a profile screen may include Password and Confirm Password fields. For the Confirm Password field, the value of the Field to Match Against field would be the Password field.

  Note: Identity Manager uses a screen-defined logical attribute to verify that the value of two fields on a screen match. For the Field to Match Against field to appear, the Attribute Name field must be set to (Screen Logical Attribute).

- **Label right**

  Specifies text that appears to the right of the field. You can use the Label right field to provide a description or help text for fields on a Profile screen.

- **Label span**

  Specifies the number of columns that the label will span.

- **Max Length**

  Sets the maximum number of characters that can be entered for this field.

- **Name**

  Specifies the label you want for this attribute in the screen.

- **Permission**

  Determines the privilege level for the field.

  Note: If a field is required, choose a Required setting. Required fields are indicated by a dot in the screen.

  - **Read**

    An administrator can view but not modify the field.

  - **Read/Write**

    An administrator can see the current value of the field (if one exists), and can enter a value for the field.

  - **Read/Write Required**

    The field is required, but otherwise functions as the Read/Write setting.

  - **Write**

    An administrator cannot see the current value of the field (if there is one), but can enter a value.

    For example, an administrator can change a user's password, but cannot view the user's current password.

  - **Write Once**

    A value can be entered once, but not modified.

    For example, an administrator can specify an organization when a user is created, but cannot modify that organization at a later time.

  - **Write Required**

    A field is required, but otherwise functions as the Write setting.

- **Preserve non-options**

  Controls whether Identity Manager preserves existing values for an attribute, if those values are not valid. For example, a State field includes the options Massachusetts and New York. However, an existing user is from California. If this option is selected, Identity Manager displays California as if it was a valid option for that user. If this option is not selected, Identity Manager displays the first option in the list (Massachusetts). If the field is not required, the value is blank.

  - **Default**

    Identity Manager will force the user to select only the valid options.

- **Rows**

  Specifies the number of rows that a text area for user input should include.

  For example, you may want to define a text area for the Description field, which allows users to enter four rows of text.

  Note: This field is available only when you select the Text Area style.

- **Show Time Picker**

  Displays a calendar control that users can use to select a date and time.

- **Size**

  Specifies the size of the field. Enter a number based on the style of field. For text and password, enter the number of characters. For drop-down, select, multi-select, and multi-text, enter the number of rows.

- **Source of Selection Options**

  Specifies how a field that contains multiple options is populated.

  - **None**

    Identity Manager will not use an external source for selection options.

  - **Select Box Data**

    Specifies that Identity Manager will populate the options in the field using select box data (see page 51).

  - **Simple List**

    Allows you to enter a list of options in a text box. If you select this option the following field appears:

    Selection Options

    Enter options on separate rows. If the option has separate display and storage values, enter them as "storage-value;display-value".

- **Depends on the value of another field**

  Specifies that Identity Manager will populate the options in the field based on options in a other field on the task screen. The other field must also be populated using select box data, or also depend on the value of another field.

  Dependency on another field is defined in the select box data configuration.

- **Javascript**

  Specifies JavaScript that contains the options for the field. If you select this option, the following field appears:

  Selection Options (JavaScript)

  This JavaScript must contain a function with the signature "function getOptions(FieldContext)" and return a pipe delimited string of options. If the option has separate display and storage values, they must be separate by a semicolon (;).

- **Style**

  Determines the presentation of the field.

  Style Options (see page 36) lists styles that you can select for a field.

- **Tip Text**

  Specifies text that describes a field. The text appears on the screen next to the field to which it applies.

- **Unchecked Value**

  Specifies the value of a field when its check box is cleared. For example, the unchecked value for the Enabled field is false.

  The default value is false.

  Note: This field is visible when the checkbox style is selected.

  - **Allow Other Unchecked Values**

    When unchecked, sets the value of the attribute to false in the user store if the attribute is empty. When checked, Identity Manager allows the attribute to be empty.

    Use this field to automatically set empty attributes to false in the user store.

- **Validation Expression**

  Contains a regular expression that performs task-level validation.

- **Validation Java Class**

  Contains the fully-qualified name of a Java class that performs the validation, for example:

  com.mycompany.MyJavaValidator

  Identity Manager expects the class file to be located in the root directory designated for custom Java class files.

- **Validation JavaScript**

  Contains the complete JavaScript code that performs the validation.

  You can also use this field to specify JavaScript code that dynamically hides/shows and enables/disables particular fields based on current values of other fields.

  Note: You must provide JavaScript code in this field. With task-level validation, you cannot reference a file containing JavaScript code.

Field Styles

The Style field enables you to specify how a field is displayed on a Profile screen. You can select the following styles:

Note: The list of styles that are available in the Style field depends on the type of field that you are configuring. Some of these options may not appear in the Style field for the type of field that you are defining.

- **Check Box**

  Adds a check box next to the field name, which enables or disables a setting. For example, use a check box for the Enable User field. If the check box is selected, the user account is enabled. If the check box is clear, the user account is disabled.

- **Check Box Multi-Select**

  Adds a check box next to each option for a field. Users can select multiple options from the option list.

  Use this field for multivalued attributes only.

- **Date Picker** (see page 38)

  Displays a calendar icon next to a date field, such as Start Date. Administrators click the calendar icon to display a calendar control where they can select the date they want.

- **Drop Down**

  Allows the user to select a value for the field. Only one value is visible. Users click an arrow to see additional values in the list.

  The user can select a single value from the list.

- **Drop Down Combo**

  Provides the same choice of values displayed by a Drop Down style, but adds a text box where the user can enter a new value.

- **Group Selector**

  Displays a control for selecting a group.

- **Hidden**

  Retrieves the field's value from the object, but the field's label and value are not displayed on the task screen.

- **Multi-Select**

  Displays a list of values for a field.

  In a Multi-Select box, the possible values for the field are visible in the list box. Users can select multiple values from the list.

  Use only with multivalued attributes.

- **Multi-Text**

  Allows the user to enter multiple values in a text box.

  Use only with multivalued attributes.

- **Object Selector**

  Displays a control for selecting a managed object.

  This style is typically used in account management screens.

  Use only with multivalued attributes.

- **Option Selector**

  Displays two list boxes, which show the available and current values for the field. The user clicks buttons to add or remove current values.

  Use only with multivalued attributes.

- **Option Selector Combo**

  Displays the two list boxes used for an Option Selector style plus a text box where the user can enter a new value.

  Use only with multivalued attributes.

- **Organization Selector**

  Displays a control for selecting an organization.

- **Password**

  Displays the field's value as a series of asterisks. For example, the password secret is displayed as ******.

- **Radio Button Single Select**

  Displays a list of values for a field. A radio button appears next to each value.

- **String**

  Displays the field's value as read only. If no value exists, the field is blank.

- **Text**

  Displays a box where the user can enter a value for the field.

  If the field's permission is read only, the value is displayed as a label.

- **Text Area**

  Displays a box where the user can enter values that are longer than a text field. For example, a description may require a text area.

- **User Selector**

  Displays a control for selecting a user.

Note: You can specify values, called *options*, in drop down menus, drop down combo boxes, multi-select boxes, option selector, option selector combo boxes, and single-select boxes. Users can select one or more options to populate a field value. How to Populate Field Options (see page 46) provides information about the methods you can use to specify field options.

Date Picker Options

The Date Picker style allows you to add a calendar icon to a field on a Profile screen. Users can click the icon to open a calendar that they can use to select a date. The selected date is stored in the profile attribute that is associated with the field. For example, you could add the calendar control to a Start Date field on the Profile tab for a Create Contractor task. When an administrator selects the first day of the contract, Identity Manager stores that date in the user's profile.

The Date Picker style has the following configuration settings:

- **Date Display Pattern (Optional)**

  Determines the format of dates displayed in a field and in the Date Picker control. Specify the date display pattern using Java conventions. For example, the following Java expression is displayed as Oct 2011:

  MMM yyyy

  The Date Display Pattern field appears only when the Date Picker style is selected.

  Note the following when specifying date display patterns:

  - The date picker control supports a *subset* of the Java date formats.

    The complete list of Java date formats appears in the documentation for Java™ 2 Platform Std. Ed. v1.4.2 at the Oracle website (http://java.sun.com/j2se/1.4.2/docs). Search for SimpleDateFormat.

    The following formats, which are supported in the SimpleDateFormat, are *not* supported by the date picker control in Identity Manager:

Symbol	Meaning	Type	Example
G	Era	Text	"GG" -> "AD"
D	Day in year (1-365 or 1-364)	Number	"D" -> "65" "D" ->"065"
W	Week in month (1-5)	Number	"W" -> "3"
k	Hour (1-24)	Number	"k" -> "3" "kk" ->"03"
K	Hour (0-11 AM/PM)	Number	"K" -> "15" "KK"->"15"
S	Millisecond (0-999)	Number	"SSS" -> "007"

  We do not recommend specifying a date display pattern if the Environment supports multiple locales. If a display pattern is not specified, the date is displayed in a format appropriate for the locale of the user.

- **Date Storage Pattern**

  Determines how the date is stored in the user store. Specify the date pattern using Java conventions. (See the description of the Date Display Pattern for more information.)

- **Tip Text**

  Specifies text that appears next to the date picker on the profile screen.

  You can use this field to provide additional information about the date picker control.

- **Show Time Picker**

  Allows users to specify time in addition to the date when using a calendar control on a Profile screen. The time is stored in the user store.

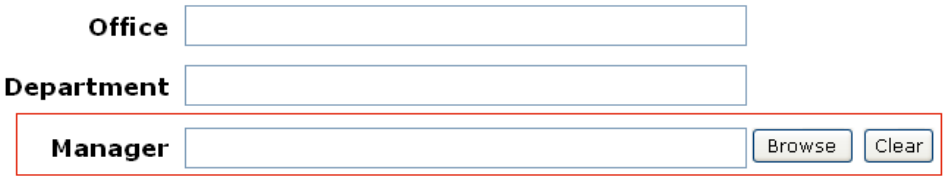
- **Hide Seconds**

  Hides the seconds display in the time picker control.

Note: The Date Storage Pattern, Show Time Picker, and Time Picker Format fields appear only when the attribute that you selected does not have the Date, ISODate, or UnicenterDate value type in the directory configuration file (directory.xml). For more information about value types, see the *Configuration Guide*.

Object Selector Options

The Object Selector style allows administrators to add search functionality to a field on a profile screen. Users can use this functionality to search for and select an object to associate with the attribute described in the profile. For example, an administrator may add an object selector to the Manager field to allow users to search for a new user's manager in a Create User task. When the user selects a manager and submits the Create User task, Identity Manager stores information about that manager in the new user's profile.



In most cases, the object selector search allows users to select and store a single value. However, administrators can also configure the object selector to allow users to search for and select multiple values. In this case, the selected values are stored in a multi-valued attribute on the object.

Configure an Object Selector

The object selector style adds a Browse button to a field. Users can click the button to search for and select an object to populate the field.

The object selector can be applied to fields for single value or multivalue attributes.

To Configure an Object Selector

1. Edit a profile screen definition (see page 27).
2. Add an additional field to the profile screen by using the controls below the list of fields.
3. Click the right arrow icon to open the Field Properties dialog for the attribute that you are adding.
4. Specify values for the following fields:

   Attribute Name

   Select the attribute that is associated with the field.

   For example, if you are configuring the field that specifies a user manager, select the attribute in the user store that stores that information.

   Note: If you are configuring a field for an attribute that stores multiple values, be sure that the attribute is defined as multivalued in the directory configuration file (directory.xml). See the *Configuration Guide* for more information.

   Style

   Select the Object Selector style.

   Object Type

   Specify the type of object that the user will search for. For example, if you are configuring an object selector for the Manager field, select the Users object type.

   Restrict to Single Value

   Allows administrators to select only a single value when they search for an object.

   Note: This option is available only when you specify a multi-valued attribute in the Attribute Name field.

   Display attribute

   Select the attribute of the selected object that is displayed when the value is selected.

   In the Manager example, select Full Name or User ID so that users can easily identify the manager.

   Note: The unique identifier of the object is stored in the attribute.

Search Screen

Select the search screen that administrators use to search for the object.

Size

Specifies the number of items to show in the list box.

Note: This field is available only when you specify a multi-valued attribute in the Attribute Name field.

Default

Select the default object that Identity Manager uses when no other object is selected.

5. Click Apply, then click OK to return to the Select Screen page.

   Identity Manager adds the object selector to the field that you edited.

6. Select the tab that you edited and click OK.

   Identity Manager saves the changes to the screen.

Structured Attribute Display

A structured attribute enables a single attribute value to store multiple related values—for example, a structured attribute can contain a user's first name, last name, and email address in a single value. These types of attributes are used by certain endpoint types, but can be managed in Identity Manager.

You can configure Identity Manager to display the values in a structured attribute as a table, which users can optionally edit. In this case, changes made to values in the attribute are stored in the user store and propagated back to the endpoint account (if synchronization is enabled).

Prerequisites for Structured Attribute Support

To add structured attribute support in the User Console, the definition for the structured attribute in the directory configuration file (directory.xml) must include the following parameters:

- multivalued="true"

  The attribute must be a standard multi-valued attribute in the user store.

- displayhint="*value1;value2;valueN*"

  The displayhint parameter should contain a list of fields that are available in the attribute value, separated by a semicolon (;).

- valuetype="structured"

  The valuetype parameter must be set to "structured" to configure a display table in the User Console. If this parameter is not set correctly, the fields required to configure the display table do not appear.

A completed attribute description for a structured attribute should resemble the following:

<ImsManagedObjectAttr physicalname="emailaddress" required="false" searchable="false" multivalued="true" displayhint="email;type;primary" valuetype="structured">

Note: For more information on how to configure the directory.xml file, see the *Configuration Guide*.

Configure a Structured Attribute Display

To enable users to add or modify values in a structured attribute, you can add a structured attribute display to a profile screen. This display is typically used in account templates for endpoint types that support structured attributes.

To configure a structured attribute display

1. Configure the prerequisites (see page 43) for structured attribute support.
2. Edit a profile screen (see page 27).
3. Add an additional field to the profile screen by using the controls below the list of fields.
4. Click the right arrow icon to open the Field Properties dialog for the field that you are adding.

5. Select a structured attribute from the list of available attributes in the Attribute Name field.

   Note: The attribute you select must have the value type of *structured* in the directory configuration file (directory.xml).

6. Select Nested Structure in the Style field.

   The fields in the Field Properties screen change based on the style selection.

7. Add fields to the display table by clicking the right arrow icon and selecting a value from the list box.

   The values that appear in this list are the values that are available in the structured attribute, as defined in the directory configuration file (see page 43) (directory.xml).

   When you select a value, Identity Manager adds that value to the display table and enables you to configure properties for that value.

8. Specify the following fields for the value in the display table configuration:

   Name

   Specifies the label for the field.

   Style

   Specifies the display properties for the field. You can select one of the following style options:

   - **Checkbox**

     Adds a check box next to the field name, which enables or disables a setting.

   - **Date**

     Displays a text box where administrators can enter a date.

     Identity Manager validates the date format.

   - **Dropdown**

     Allows the user to select a value for the field. Only one value is visible. Users click an arrow to see additional values in the list.

     The user can select a single value from the list.

   - **Dropdown Combo**

     Provides the same choice of values displayed by a Dropdown style, but adds a text box where the user can enter a new value.

   - **Object Selector**

     Allows you to add a search screen for selecting a managed object.

   - **Radio Button**

     Displays a list of values for a field. A radio button appears next to each value. Users can select a single value from the list.

- **String**

  Displays the field's value as read only. If no value exists, the field is blank.

- **Structured**

  Displays an Add button adds a new value to the nested compound attribute table.

- **Text**

  Displays a box where the user can enter a value for the field.

  If the field's permission is read only, the value is displayed as a label.

Sortable

Determines whether users can sort the display table based on the selected field.

9. Select the Allow Reordering of Values check box to allow administrators to reorder the list of structured attributes in the display on the profile screen.

   When selected, this setting adds up and down arrows to the last column of the structured attribute display.

10. Add support for adding information from other managed objects (see page 45) in a structured attribute, if necessary.

    Note: Configuring support for other managed objects adds a search screen in the structured attribute display table that allows users to search for and add information that is stored in other types of managed objects. For example, you may want to allow users to select SAP Roles to add to a structured attribute on a user profile.

11. Click Apply, then click OK.

    The structured attribute display is added to the Profile screen that you edited.

Add Other Managed Objects in a Structured Attribute Display

In some cases, you may want to add other managed objects to a structured attribute. For example, you may have a structured attribute in a user profile that lists SAP roles and a start and end date for when users can use those roles.

To configure support for this use case, you add a structured attribute display table as described in Configure a Structured Attribute Display (see page 43), and then configure additional fields that allow you to search for and store information about another type of managed object in the structured attribute.

When this support is configured, Identity Manager displays a search screen that allows users to search for and select managed object values to add to the structured attribute.

To add managed objects to a structured attribute display

1. Configure a structured attribute display (see page 43).
2. Specify the following fields, as needed:

   Object Field

   Select the field that contains the reference to the managed object. In most cases, this is the unique identifier for the managed object.

   Object Type

   Select the type of object that contains the values to add to the structured attribute.

   For example, to add SAP roles to a structured attribute on the user profile, you would select the SAP roles object.

   Object Attribute (optional)

   Select the attribute of the managed object that will be used to populate the Object Field.

   This attribute is only needed if the field that contains the reference to the managed object is not the unique name for the managed object. If no value is provided for this field, the unique name is used.

   Search Screen

   Specify the search screen that users see when they click the Add button to add additional values to the structured attribute.
3. Click Apply, then click OK.

How to Populate Field Options

There are several field styles that allow you to provide options for users to choose:

- Check Box Multi-Select
- Dropdown
- Dropdown Combo
- Multi-Select
- Option Selector
- Option Selector Combo

- Radio Button Single-Select
- Single-Select

For example, the Office field may contain the list of all the offices that a company has. Users can select the office where they work to populate the field.

Identity Manager provides the following methods for populating options:

Simple List

Allows you to enter a list of options in a text box. Identity Manager uses the text that you enter as the options for the field.

Select Box Data

Allows you to configure field options using a select box data.

JavaScript

Allows you to specify JavaScript that provides the options for the field.

Logical Attribute Handlers

Allows you to specify a logical attribute handler to provide field options.

More Information:

Select Box Data (see page 51)

How to Select a Field Population Method

Identity Manager provides four methods for populating field options:

- Simple Lists
- Select Box Data
- JavaScript
- Logical Attribute Handler

When selecting a method, consider the following criteria:

- Ease of implementation

  Some methods allow you to configure field options in the field properties dialog when you configure a profile screen. Other options require additional configuration or custom code.

- Support for dynamic options

  Certain methods allow you to write custom code to dynamically populate field options, or to retrieve field options from another source, such as a database.

- Support for dependent fields

  Certain methods allow you to configure a dependency between two fields in a task screen. For example, the options that are available in the City field may depend on the option a user chooses in the State field.

The following table summarizes the characteristics of each field population method.

Method	Description	Dynamic?	Supports Dependent Fields?
Simple Lists	Administrators enter static options in the field properties dialog.	No	No
Select Box Data	A static list of options is imported to a database from an XML file, which can be generated dynamically.	Yes. The options in dependant fields can change, based on selected values.	Yes, for hierarchical fields only

Method	Description	Dynamic?	Supports Dependent Fields?
JavaScript	A JavaScript function provides a dynamic list of options. The JavaScript is configured in the field properties dialog. This server-side JavaScript may access any Java APIs available on the application server	Yes	No

Method	Description	Dynamic?	Supports Dependent Fields?
Logical Attribute Handler	A custom Java Logical Attribute Handler provides a dynamic list of options. An administrator writes the Logical Attribute Handler using the Identity Manager Logical Attribute API, and then configures the Identity Manager environment to use the Logical Attribute Handler. The administrator then associates the field with the logical attribute.	Yes	No

Use Simple Lists for Field Options

You can specify a static list of options for fields in a profile screen by using the Simple List selection option style. When users select one or more of the options (depending on the field style), Identity Manager stores that value in the user store.

To use a simple list to populate field options

1. Modify a profile screen (see page 27).
2. Select a field to modify or add a new field.
3. If you are adding a new field, select the attribute that is associated with the field from the list box.
4. Select one of the following styles:
   - Check Box Multi-Select
   - Dropdown
   - Dropdown Combo
   - Multi-Select
   - Option Selector
   - Option Selector Combo

- Radio Button Single-Select
- Single-Select

The fields in the Field Properties dialog change based on the style selection you make.

5. Select Simple List in the Source of Selection Options field.

   An additional field, Selection Options, appears.

6. Enter the options for the field in the Selection Options field.

   Each option should appear on a separate line.

   If you want Identity Manager to store a value in the user store that is different from the value that is displayed in the option list, specify each option as follows:

   "storage-value;display-value"

7. Specify one of the following values in the Preserve Non-Options field:

   - Yes—Existing values that do not match one of the valid options are preserved.
   - No—Users must select a value from the pre-defined option list. Existing values that do not match an existing value are not preserved.

8. Specify values for the remaining required fields.
9. Click Apply, then click OK.

   Identity Manager saves the current field properties.

Select Box Data

Identity Manager task screens include fields that allow users to select a value. These fields include the following:

- Check Box Multi-Select
- Dropdown
- Dropdown Combo
- Multi-Select
- Option Selector
- Option Selector Combo

- Radio Button Single-Select
- Single-Select

You can specify custom data that you want to use to populate select boxes in XML files. For example, you can use the Select Box Data XML files to populate options for a City or State drop down box on a Profile tab for the Create User task.

You can also use the Select Box Data XML file to configure a dependency between two fields in a task screen. For example, the options that are available in the City field may depend on the option a user chooses in the State field.

How to Populate Fields Using Select Box Data

To populate a field using Select Box Data, you must perform the following tasks:

1. Create a Select Box Data XML file.
2. Import the Select Box Data XML file in Identity Manager.
3. Configure fields to use select box data.

Create a Select Box Data XML File

Each Select Box Data XML file contains data that can be used to populate the options for a Identity Manager User Console control.

To create a Select Box Data XML file

1. Create a text file with an .XML extension using a text or XML editor.
2. Create a file using the format described in Select Box Data XML File (see page 52).
3. Save the file.

   You import this file into Identity Manager environment to populate the options in Identity Manager User Console controls.

The Select Box Data XML File

The Select Box Data XML file is a tree-based collection of elements and child elements. You can populate fields in any Identity Manager profile screen with elements or child elements in a Select Box Data XML file. After creating the XML file, you can import it using Import Select Box Data task in the User Console.

The following code shows an example of a Select Data XML file that populates the state names and city names when the country is selected as Australia or UK:

```
<places name="places" displayName="places">
        <country name="AU" displayname="Australia">
                <state name="VIC" displayname="Victoria">
                        <city name="MEL" displayname="Melbourne"/>
                        <city name="GEEL" displayname="Geelong"/>
                        <city name="BAL" displayname="Ballarat"/>
                </state>
                <state name="NSW" displayname="New South Wales">
                        <city name="SYD" displayname="Sydney"/>
                        <city name="NCL" displayname="Newcastle"/>
                        <city name="WOD" displayname="Wodonga"/>
                </state>
                <state name="QLD" displayname="Queensland">
                        <city name="BRIS" displayname="Brisbane"/>
                        <city name="CNS" displayname="Cairns"/>
                        <city name="TVL" displayname="Townesville"/>
                </state>
        </country>
<country name="UK" displayname="UK">
                <state name="SU" displayname="Surrey">
                        <city name="LON" displayname="London"/>
                        <city name="READ" displayname="Reading"/>
                </state>
                <state name="WLS" displayname="Wales">
                        <city name="CDF" displayname="Cardiff"/>
                        <city name="SWN" displayname="Swansea"/>
                </state>
        </country>
</places>
```

Note: The Administrative Tools include a sample Select Box Data XML file, which is installed in the following location:

admin_tools\samples\SelectBoxData

admin_tools is the installed location of the Administrative Tools.The Administrative Tools are placed in the following default locations:

- **Windows:** C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools
- **UNIX:** /opt/CA/IdentityManager/IAM_Suite/Identity_Manager/tools

Note: For more information about the sample, see the readme.txt file in the SelectBoxData directory.

Description of Select Box Data XML File

The Select Box Data XML file is organized as follows:

Root Element

Identifies the Select Box Data XML file. There is only one root element per Select Box Data XML file. The root element is a container for all the elements and child elements. In the example code in the preceding topic, places is the root element. These elements cannot be used to populate the fields.

Provider Element

Specifies the nodes in the tree of a Select Box Data XML file. These elements contain the options that you can use to populate fields in Identity Manager. The provider element does not have a parent element. For example, if you create two dependent fields that have options 'Melbourne' and 'Victoria', the corresponding elements in the Select Box Data XML file must belong to the same provider element. In the illustration below, the city 'Melbourne' is dependent on the state 'Victoria'. The provider element for both the options is 'Australia'.

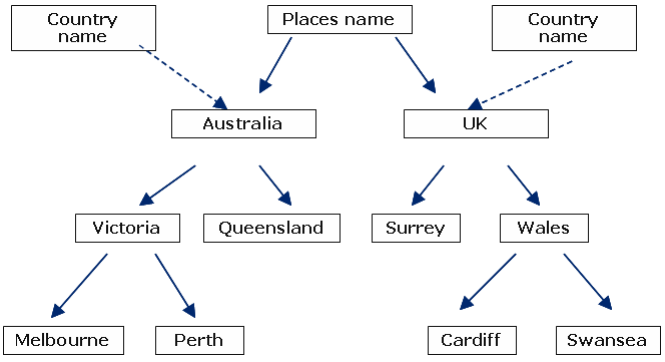
Elements

Any XML element in Select Box Data XML File is identified as an element. An element can be a parent element or a child element. In the following illustration, 'Victoria' is a parent element of 'Melbourne'. Similarly 'Victoria' is a child element of 'Australia'.

Child Elements

All the XML elements that are contained as part of elements higher in the tree structure are called child elements.

The following illustration identifies the Select Box Data structure for the example in the preceding topic.



Each element is identified by the following attributes:

Display Name

Identifies the element name which is displayed when the element appears in the Identity Manager User Console. For example, Melbourne and Queensland are the display names.

Type

Identifies the type of element. In the example in the preceding topic, state name and city name are type elements.

Note: In the Identity Manager User Console Display Name is referred to as Child Name and Type is referred to as Child Type.

Import the Select Box Data XML File

The Select Box Data XML file includes elements that Identity Manager uses as options in select box fields on task screens. You must import the Select Box Data into Identity Manager environment to use the data as the source for selection objects.

To import Select Box Data XML file

1. Click System, Tasks, Select Box Data, Import Select Box Data.

   The Import Select Box Data page appears.

2. Click Create Provider.

   The Create Select Box Data page appears.

3. Complete the following fields:

   Name

   Identifies the unique name of the Select Box Data XML file. The identifier name should be unique. Identity Manager validates the uniqueness of the identifier name.

   Description

   Identifies more information about the Select Box Data XML file.

   Precedence

   Identifies the precedence of the Select Box Data XML file compared to other Select Box Data XML files. Precedence should be numeric.

4. Click Browse to search for the XML file, locate the XML file and click Create.

   The XML file is imported.

5. Click Close.

   The newly imported Select Box Data XML file is displayed in the list.

Use Select Box Data for Field Options

Configure fields in a task screen to use select box data as the source of selection for options fields.

To configure fields to use Select Box Data

1. Modify a profile screen (see page 27).
2. Add or select a field on the profile screen and click the right arrow icon to display the field properties.
3. Complete the properties for the field you added. Set Style to one of the following values:
   - Check Box Multi-Select
   - Dropdown
   - Dropdown Combo
   - Multi-Select
   - Option Selector
   - Option Selector Combo
   - Radio Button Single-Select
   - Single-Select

   The Source of selection options field appears.
4. Select *one* of the following options for the Source of selection options field.

   Select Box Data

   Specifies that the field is populated with select box data. If you select this option, the Edit for the Select Box Data for Options appears.

   A list of Root Elements for the imported select box data also appears.

   Click the Edit button to display a Select Box Data Options page that allows you to browse the imported select box data. Click an Element Name to view information about the child elements for that element. When the list of elements displayed represents the list of options to use for the profile field, click OK.

When there are large lists of elements, the following two fields may be used to filter the list. These fields support a wildcard character (*). Click the Refresh button to filter the results.

Child Name Filter

Identifies the name of the element or child element in select box data.

Child Type Filter

Identifies the type of the element or child element in select box data XML file.

For more information about element name and element types, see The Select Box Data XML File (see page 52) .

Note: The values you can select from this field are populated from the select box data XML file you have imported into Identity Manager.

Depends on the value of another field

Specifies that the field is populated based on the values selected in another field. If you have selected this option, see How to Create Dependent Fields Using Select Box Data (see page 57).

Select an existing field on the form Options source field.

5. Select one of the values for Preserve non-options field.
6. Complete the other necessary field properties and click Apply.
7. Click OK to save the changes.

   The fields that you have configured will appear to the user on the profile tab of the selected admin task. The values for select-based controls that are configured to use select box data as source will be populated with the values from the Select Box Data XML file.

How to Create Dependent fields Using Select Box Data

You can create a dependency between two fields on Identity Manager task screen. The following process describes the steps that you must follow to create dependency between two fields:

1. Create a field that uses select box data file to populate the options.

   For example, you can create a field called State that uses the Dropdown style and Select Box Data for options. Each option in State corresponds to an element or a child element in the select box data. Each element in State has City child elements.

2. Create another field which is populated based on the option that you have selected in Step 1.

   For example, you can create a field called City, which has a Dropdown style and is dependent on the State field.

More Information:

The Select Box Data XML File (see page 52)

Use JavaScript For Field Options

You can specify the options that appear in fields on a profile screen by writing custom JavaScript.

To use a JavaScript to populate field options

1. Modify a profile screen (see page 27).
2. Select a field to modify or add a new field.
3. If you are adding a new field, select the attribute that is associated with the field from the list box.
4. Select one of the following styles:
   - Check Box Multi-Select
   - Dropdown
   - Dropdown Combo
   - Multi-Select
   - Option Selector
   - Option Selector Combo
   - Radio Button Single-Select
   - Single-Select

   The fields in the Field Properties dialog change based on the style selection you make.
5. Select JavaScript in the Source of Selection Options field.

   An additional field, Selection Options (JavaScript), appears.

6. Enter JavaScript to provide the options for the field in the Selection Options (JavaScript) field.

   The JavaScript you enter must contain a function with the signature "function getOptions(FieldContext)" and return a pipe delimited string of options. If the option has separate display and storage values, enter as "storage-value;display-value"

   For example:

```
function getOptions(FieldContext) {
  return "1;one|2;two|3;three|4;four";
}
```

7. Specify one of the following values in the Preserve Non-Options field:
   - Yes—Existing values that do not match one of the valid options are preserved.
   - No—Users must select a value from the pre-defined option list. Existing values that do not match an existing value are not preserved.
8. Specify values for the remaining required fields.

   Note: For information on required fields, see the User Console online help.
9. Click Apply, then click OK.

Identity Manager saves the current field properties

Use Logical Attribute Handlers For Field Options

You can use a logical attribute to populate a list of field options. Logical attribute values (in this case, the options) are not directly associated with or written to the user store. The logical attribute values are presented in a profile screen field. When a user selects an option and submits a task, the selected value is processed by a logical attribute handler, which stores the value in the physical attribute associated with the logical attribute.

Note: Identity Manager includes a sample logical attribute handler, called StateSelector, that you can use as a base for creating a logical attribute handler that populates field options. The StateSelector sample folder is installed under samples\LogicalAttributes in the Administrative Tools. The Administrative Tools are placed in the following default locations:

- **Windows:** C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools
- **UNIX:** /opt/CA/IdentityManager/IAM_Suite/Identity_Manager/tools

For information on using the sample, see the readme.txt file in the StateSelector directory.

To use a simple list to populate field options

1. Create a logical attribute handler.

   Note: You use the Logical Attribute API to write a logical attribute handler. For more information, see the *Programming Guide for Java*.

2. In the Identity Manager User Console, modify a profile screen (see page 27).

3. Add a new field.

4. Select the logical attribute that is associated with logical attribute handler that you created.

   Note: Logical attributes are indicated by a preceding and trailing pipe (|) character.

5. Select one of the following styles:

   - Check Box Multi-Select
   - Dropdown
   - Dropdown Combo
   - Multi-Select
   - Option Selector
   - Option Selector Combo
   - Radio Button Single-Select
   - Single-Select

   The fields in the Field Properties dialog change based on the style selection you make.

6. Select None in the Source of Selection Options field.

   An additional field, Selection Options, appears.

7. Specify one of the following values in the Preserve Non-Options field:

   - Yes—Existing values that do not match one of the valid options are preserved.
   - No—Users must select a value from the pre-defined option list. Existing values that do not match an existing value are not preserved.

8. Specify values for the remaining required fields.

   Note: For information on required fields, see the User Console online help.

9. Click Apply, then click OK.

Identity Manager saves the current field properties.

Dynamically Populating the Organization Field

If the user store that Identity Manager manages includes organizations, the default Create User task includes an Organization field. An administrator must search for and select the appropriate organization before creating a user profile.

To simplify the Create User task, you can configure Identity Manager to populate the Organization field dynamically, based on the administrator who is executing the task. In this case, the administrator does not have to specify an organization. The user is created in the organization where the administrator's profile exists. For example, if an administrator, whose profile exists in the Employees organization, creates a user profile for a new hire, Identity Manager creates the new profile in the Employees organization. If an administrator in the Suppliers organization uses the same Create User task, the profile for the new user that the second administrator creates would exist in the Suppliers organization.

Configure a Dynamic Organization Field

When you configure a dynamic organization field for the Create User task, Identity Manager creates new users in the organization where the profile for the administrator who is creating the user exists.

To configure a dynamic organization field

1. In the User Console, go to Roles and Tasks, Admin Tasks, Modify Admin Task.
2. Search for and select the Create User task.
3. On the Tabs tab, click the edit icon to edit the Profile tab.
4. In the Screen field, click Browse to display a list of screens to edit.
5. Select the Create User Profile screen and click Edit.
6. Locate the Organization and click the edit icon to edit its properties.
7. Set Style to Hidden.
8. In the Default JavaScript field, enter the following:

```
function defaultValue(FieldContext)
{
return FieldContext.getAdministrator().getOrg(null).getUniqueName();
}
```

9. Click Apply.
10. Click the left arrow next to Field Properties to return to the screen.

How to Change Field Display Properties Dynamically

Identity Manager can set certain field display properties based on the value of other fields in a profile screen. Using JavaScript, you can hide and show a field, or enable and disable a field. For example, you can use JavaScript to show an Agency field if the Employee Type is set to Temp. If the Employee Type is Full Time or Part Time, the Agency field is hidden.

You enter the JavaScript in the Initialization JavaScript or Validation JavaScript fields in the Field Properties dialog in the profile screen definition. The methods that control the display of a field are available in the FieldContext class of the init and validate methods.

For example, to control the display of the Agency field described above, you would enter the following JavaScript code in the Validation JavaScript field in the Field Properties for the Employee Type field, since the changes to the Employee Type field control the display of the Agency field:

```
function validate(FieldContext, attributeValue, changedValue, errorMessage) {
    if (attributeValue == "Temp") {
        FieldContext.showField("Agency");
      }
    else {
        FieldContext.hideField("Agency");
    }
    return true;
}
```

To ensure that the JavaScript is triggered when the field value changes, set the Validate on Change field to Yes.

Configure Dynamic Field Display Properties

You can configure Identity Manager to hide and show, or enable and disable a field on a profile screen based on the value of another field on that screen.

To configure dynamic field display properties

1. Edit the profile screen (see page 27).

   Identity Manager displays a list of fields configured for the screen.

2. Add the field for which you are configuring dynamic field properties, if necessary.

3. Click the Edit icon next to the field name to edit it.

   Identity Manager displays the Field Properties dialog.

4. Enter JavaScript code in the Validation JavaScript field using the following method:
   `function validate(FieldContext, attributeValue, changedValue, errorMessage)`

   The FieldContext class includes the following methods for showing/hiding and enabling/disabling a field:

   public void hide();

   Hides the field.

   public void show();

   Displays the field.

   public void hideField(String attrName);

   Hides the current field.

   public void showField(String attrName);

   Displays the current field.

   public void disable();

   Disables the current field.

   public void enable();

   Enables the current field.

   public void disableField(String attrName);

   Disables a field for a specific attribute.

   public void enableField(String attrName);

   Enables a field for a specific attribute.

5. Click Apply, then Click OK.

Screen-Defined Logical Attributes

Screen-defined logical attributes are fields on a Profile tab that are defined locally for the current task. You can use these screen-defined logical attributes to manipulate objects in a task screen or *modify* physical attributes that are stored in the user store.

Screen-defined logical attributes are defined, initialized, validated, populated, and implemented using JavaScript.

For example, if you had 3 physical attributes that stored a date (month, day, year), but you wanted to present the user with a single field to enter the date, you can configure a screen logical attribute for the date field. Once the user enters a date, you can configure validation JavaScript to parse the date into month, day, and year values and set them into the physical attributes (which would likely be hidden attributes on the screen).

Note: Attributes enclosed within '|' are identified as screen-defined logical attributes.

The screen-defined logical attributes are useful when you are creating generic tasks that are not bound to any primary object. In this case, you create the fields on the Profile tab using only screen defined logical attributes. You cannot specify physical attributes.

Add Screen-Defined Logical Attributes

Any field on a profile task can be defined as a screen-defined logical attribute. You can use these screen-defined logical attributes to manipulate objects locally within the scope of that profile screen or modify physical attributes in the object store. For example, you can use screen-defined logical attributes to capture a note or a warning on a profile screen, or process a user-supplied value before storing it in the physical attribute.

To define fields as Screen-Defined Logical Attributes

1. Modify Profile Screen (see page 27) to add or modify fields to use screen-defined logical attribute.
2. Create or update field properties with the screen-defined logical attribute-specific values:

   Attribute Name

   Select (Screen Logical Attributes) in the Attribute Name field.

   |Attribute Name|

   Identifies the attribute name for the field. This can be any name you choose.

   Multi-valued

   Specifies that the screen-defined logical attribute is multi-valued.

   Note: By default, this option is unchecked. If this field is unchecked, then the attribute is only single-valued.

   Name

   Enter the display name for the screen-defined logical attribute name.

   Note: If the screen-defined logical attribute has the same name as a Logical Attribute Handler, the screen defined logical attribute will override the Logical Attribute Handler.
3. Complete all the necessary field properties (see page 30).

Screen-Defined Logical Attributes in View Submitted Tasks

When you submit a task that contains screen-defined logical attributes, the original and updated values for the screen-defined logical attributes are displayed in the Task Details screen from the View Submitted Tasks tab.

Additional Components in a Profile Screen

In addition to fields, a profile screen can include one or more of the following components:

- Page separators
- Images
- Attached files
- History display
- Custom HTML text
- Links or buttons to launch tasks

Options for the Separator Attribute

When you select Separator in the Attribute field of the field properties dialog, you can add additional components to a profile screen. The Separator attribute has the following style options:

Binary (for LDAP user directories only)

Allows you to add a binary file, such as a certificate or other document, to the User Profile screen.

HTML

Displays HTML on a Profile screen.

History Display

Displays a read-only table containing details of previous history entries in chronological order.

History entries are annotations that can be added to a submitted task. They can be added as the task moves through workflow and viewed using the View Submitted Tasks task.

History Editor

Displays a text box for entering new history entries and an optional button for submitting the new entry.

Page Section

Allows you to divide the profile screen into multiple sections, which can have a different number of columns than other page sections on the same screen.

For example, the Page Section style allows you to create a profile screen that has an initial page section with a single column, and another page section with two columns.

Picture (for LDAP user directories only)

Allows you to add an image to a User Profile screen.

Space

Adds a blank space to the screen to visually separate a set of fields.

Task

Adds a link or button to a different task to the Profile tab.

Add a Binary Attribute or Picture to a Profile Screen

You can configure Identity Manager to include a binary file or display a picture on a user profile screen. For example, you can configure a user profile screen to allow users to attach a document, such as a certificate, to the profile screen or display a digital photograph of the user being managed.

Note: This functionality is available only for user profile screens. The user store must be an LDAP directory and the binary attribute or picture must be stored in attribute that is defined in the directory configuration file (directory.xml).

To add a binary attribute or picture to a Profile screen

1. Modify the profile screen (see page 27).
2. Select the field below the row where you want to add the picture and click the Add button to add one row with one field above the row you selected.

   Identity Manager adds a new field above the field you selected.
3. Click the Edit icon to edit the new field.

   The Field Properties dialog opens.
4. Select the (Separator) attribute in the Attribute Name field.
5. Select one of the following options in the Style field:
   - Binary
   - Picture

   Identity Manager displays new configuration fields in the Field Properties dialog.

6. Complete the following fields as needed:

   - **Name**

     The label you want for this field in the profile screen.

   - **Permission**

     The privilege level for the field.

     Note: If a field is required by the user store, choose a Required setting. Required fields are indicated by a red dot in the screen.

     - **Read**

       An administrator can view but not modify the field.

     - **Read/Write**

       An administrator can see the current value of the field (if one exists), and can enter a value for the field.

     - **Read/Write Required**

       A required field, but otherwise functions as the Read/Write setting.

     - **Write Once**

       An administrator cannot see the current value of the field (if there is one), but can enter a value.

       For example, an administrator can change a user's password, but cannot view the user's current password.

     - **Write Required**

       A required field, but otherwise functions as the Write setting.

   - **Label span**

     The number of columns that the label will span.

   - **Field span**

     The number of columns that the field will span (excluding the label)

   - **CSS Class**

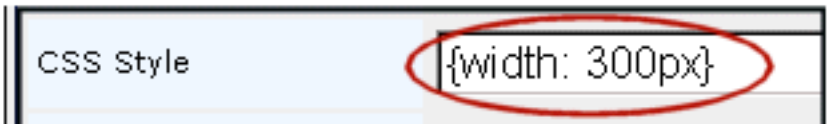
     The Cascading Style Sheet class that controls the presentation of this field.

     Note: This field is available for picture attributes only.

- **CSS Style**

  Field properties and style defined using CSS rules.

  You can use this field to set the width of a field. For example, to set the width of the field where the picture will display to 300 pixels, you specify the following in the CSS Style field:



  Note: This field is available for picture attributes only.

- **Binary Attribute Name**

  Specifies the name of the attribute that stores the image. This attribute must exist in the user store, but should not be defined in the directory configuration file (directory.xml).

- **Content Type**

  Specifies the MIME type of image that will be displayed. For example, for a binary file, you may specify application/octet-stream. For a picture, you may specify image/gif or image/jpg.

- **Alternate Content**

  Specifies the URI of an alternate image that Identity Manager displays when an image is not available for a certain user.

  Identity Manager includes a default image that is displayed when another image is not available; however, you can use this field to override the default image.

  The default image is located in *iam_im.ear*\user_console.war\ui\images\user_photo_default.jpg

  iam_im.ear is the deployed location of Identity Manager on the application server.

  Specify the path to alternate image, which has the same content type as the attribute, relative to user_console.war.

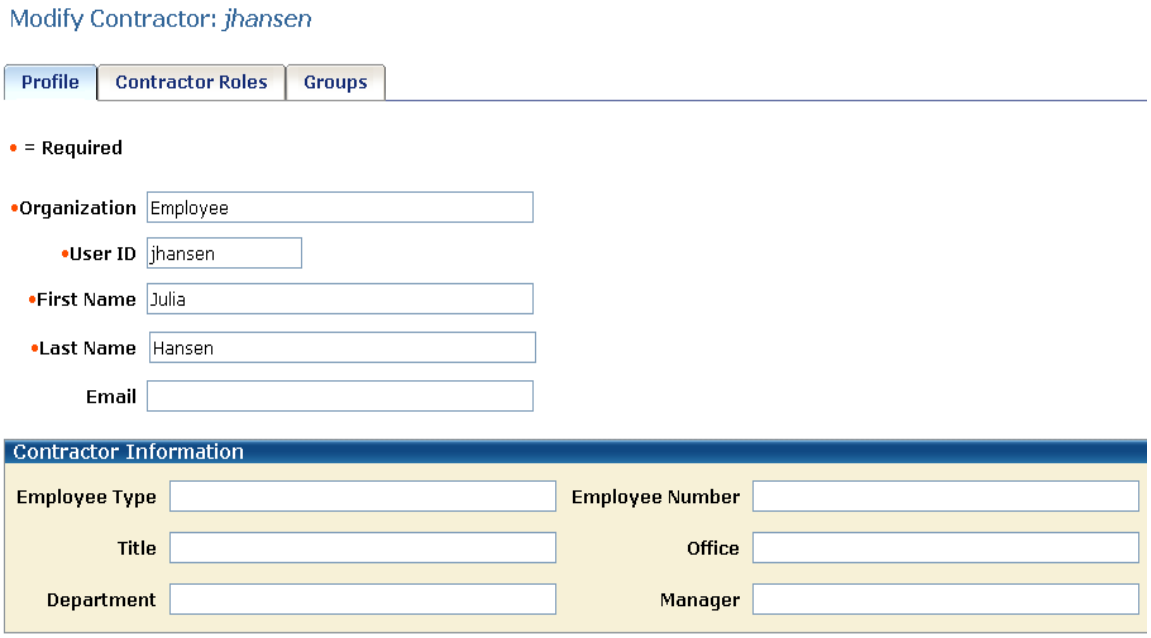
  Note: This field is only available when you select the Picture style.

7. Click Apply, then click OK to save changes.

Add Page Sections

Page sections visually separate fields in a profile screen by adding a header and applying a different number of columns to part of a profile screen. The page section layout applies until another page section is defined for the profile screen.

The following sample profile tab shows two page sections.



To add a page section

1. Modify the profile screen (see page 27).
2. Select the first field that appears in the page section you are creating and add one row with one field before the selected field.

   Identity Manager adds a new field above the field you selected. This field indicates where the page section starts.
3. Click the right arrow icon to edit the new field.

   The Field Properties dialog opens.
4. Select the (Separator) attribute in the Attribute Name field.
5. Select Page Section in the Style field.

   Identity Manager adds additional fields to the Field Properties dialog.

6. Specify values for the following fields:

   - **Columns for Layout**

     Specifies the number of columns that the page section contains.

     After you specify the number of columns, click the right arrow icon to apply the changes. Once the changes are applied, additional fields appear that enable you to specify the width of each column.

     Note: Each field includes two columns: one column for the field label, and one column for field values. To display two fields on a single row, add four columns.

   - **Table Header**

     Specifies the text that appears above the page section as a heading.

   - **Tip Text**

     Specifies text that appears below the page separator.

     You can use this field to provide a description about the page section, or to provide instructions for completing fields in the page section.

   - **Enable Hide/Show Buttons**

     Determines whether users can choose to hide a page section. When this option is selected, Identity Manager adds an arrow icon in the table header that allows them to show or hide the page section.

   - **Hide Initially**

     Specifies that a page section is hidden by default.

     If users can use the page section, select the Enable Hide/Show Buttons option when you select the Hide Initially option to allow users to show the page section.

   - **Specify Column Widths**

     Determines the width of each column in the page section. Each column width is specified as a percentage of the profile screen.

     For example, to add four columns of equal width, you would specify each column width as 25%.

     Note: The total width of the columns must be 100%.

7. Click OK to save the changes to the field properties.
8. Click Select to choose the screen that you edited or copied.
9. Click OK, then click Submit to save changes to the task.

Add a Nested Task

A nested task is an admin task that can be opened from the Profile tab of another task. Users of the first task open the nested task by clicking a link or button. For example, you can add a Delete User button to the Modify User task. If the user account is no longer valid, an administrator can click the Delete User button to remove the account without having to return to the navigation pane to select a new task.

Note: The nested task does not appear if the administrator does not have adequate privileges to access it.

To add a nested task

1. Modify the profile screen (see page 27).
2. Select the field below the row where you want to add the nested task and click the Add button to add one row with one field above the row you selected.

   Identity Manager adds a new field above the field you selected.
3. Click the Edit icon to edit the new field.

   The Field Properties dialog opens.
4. Select the (Separator) attribute in the Attribute Name field.
5. Select Task in the Style field.

   Identity Manager displays new configuration fields in the Field Properties dialog.
6. Complete the following fields as needed:
   - **Field Span**

     The number of columns that the field will span (excluding the label)
   - **Default Task**

     Specifies the task that is added to the existing task.
   - **Override Task Name**

     Specifies the name of the task link or button that will appear in the Profile screen for the active task.
   - **Task link**

     Determines whether the nested task appears as a link or a button.

- **Use current object as task subject**

  When this option is selected, Identity Identity Manager uses the subject of the active task as the subject of the task. For example, suppose the Modify User task includes a link to the Delete User task. An administrator uses the Modify User task to modify John Smith's profile. The administrator decides that John Smith's profile is no longer necessary, so she uses the Delete User link to open the Delete User task. When the task opens, Identity Manager asks the administrator if she wants to delete John Smith's profile. She does not need to search for the profile to delete.

- **Task Behavior**

  Determines how Identity Manager opens the task.

  - **Replace active task**

    Opens a new task before the active task completes. The new task replaces the previous task. When the nested task completes, users are not returned to the original task.

  - **Nest task within active task**

    Submits the new task before the active task completes. When users complete the new task, they are returned to the original task.

  - **Nest task within active task and execute when only after the active task completes**

    Submits the new task after the original task completes. This is called a post-task.

Add Help Text to Profile Screens

You can add text anywhere in a profile screen to provide additional information, such as online help text for a field, to users.

To add help text to a profile screen

1. Modify the profile screen (see page 27).
2. Select the field below the row where you want to add the online help text and click the Add button to add one row with one field above the row you selected.

   Identity Manager adds a new field above the field you selected.
3. Click the edit icon to edit the new field.

   The Field Properties dialog opens.
4. Select the (Separator) attribute in the Attribute Name field.
5. Select HTML in the Style field.

   The HTML field appears.

6. Enter the text that you want to appear in HTML tags, for example:

   <h1>Add your online help text here</h1>

7. Click OK.

Note: To display custom HTML in a different language, specify a resource key with the following format in the custom HTML field:
`${bundle=`*`ResourceBundle`*`:key=`*`keyID`*`}`

ResourceBundle

Identifies the resource bundle that includes the text string mapping for the key ID.

keyID

Identifies the key ID that maps to the text string to display. The mapping must exist in a resource bundle.

For example, the HTML for a localized field should resemble the following:

```
<p>
${bundle=MyResourceBundle;key=MyResourceKey}
</p>
```

For more information about resource bundles, see the *User Console Design Guide*.

Add a History Editor Field

The history editor is a text area which creates new history entries, if this text area contains text when the task is submitted. The history editor can include an optional submit button, which allows the creation of history entries without submitting the task.

To add a history editor field to a profile screen

1. Modify a profile screen (see page 27).
2. Select a field to modify or add a new field.
3. Select (Separator) in the Attribute Name field.

   Identity Manager changes the fields that are displayed.

4. Select History Editor in the Style field.
5. In the Label field, enter the name of the history editor field that appears in the profile screen.

6. Enter text that is attached to history log entries which describes the role of the user who created the log entry in the Stakeholder field.

   For example, the following description would appear in the Source column of a history display for a user with an Approver stakeholder label:

   User comment by SalesMgr (John Doe) acting as Approver

   This can be a string or a localization key, specified according to Identity Manager localization rules. The stakeholder type is blank by default, and is optional.

7. Enter the number of rows and columns for the history editor.

   Note: If you do not specify a value for rows and columns, the history editor does not display properly in the profile screen.

8. Select one of the following options in the History Level field:

   - Task Level—For approval tasks it is the task belonging to the event being approved. For non-approval tasks, this is the current task.
   - Event Level—For approval tasks, this is the event being approved. For non-approval tasks, this returns no results.

9. Specify the text that appears on the submission button in the Add Button Label field.

   The text can be a string or a localization key, specified according to Identity Manager localization rules. If it is blank (the default value), then the button label is "Add History Event".

10. Specify the CSS class to use to for the Add button in the Add Button CSS Class field.

    These strings will be included in the <input> element in the profile screen, as the contents of the class and style elements respectively.

11. Specify the CSS class to use to for the Add button in the Add Button CSS Style field.

    These strings will be included in the <input> element in the profile screen, as the contents of the ‘style’ and ‘class’ elements respectively.

12. Specify whether the history editor includes its own independent Add button by checking or unchecking the Enable Add Button field.

    If checked, this button submits only the new history entry, not the entire task.

More Information:

Add a History Display Field (see page 75)

Add a History Display Field

The history display is a list of text entries created using the history editor. The history display can appear on any profile screen, regardless of subject type. The history display has the following field property settings:

To add a history display field to a profile screen

1. Modify a profile screen (see page 27).
2. Select a field to modify or add a new field.
3. Select (Separator) in the Attribute Name field.

   Identity Manager changes the fields that are displayed.
4. Select History Display in the Style field.
5. In the Label field, enter the name of the history editor field that appears in the profile screen.
6. In the History Level field, select one of the following options:
   - Task Level—For approval tasks it is the task belonging to the event being approved. For non-approval tasks, this is the current task.
   - Event Level—For approval tasks, this is the event being approved. For non-approval tasks, this returns no results.
7. In the Show Entry Types field, select one of the following options:
   - User Created Entries Only—Show only runtime entries created using the history editor.
   - All Entries—Show all entries, including those created by workflow or the task controller.

Configure Task-Level Validation

You configure task-level validation in the User Console, when defining field properties on a profile task screen.

To configure task-level validation

1. On the profile screen, select the field to be validated and click Field properties.

   You define a Profile screen as part of defining tabs for the task.

2. Specify a value in one of the following fields, depending on how the validation rule is to be implemented:

   - Validation Expression—Contains a regular expression that performs the validation.
   - Validation Java Class—Contains the fully qualified name of a Java class that performs the validation--for example:

     com.mycompany.MyJavaValidator

     Identity Manager expects the class file to be located in the root directory designated for custom Java class files. For information on deploying Java class files, see the *Programming Guide for Java*.
   - Validation JavaScript—Contains the complete JavaScript code that performs the validation.

     JavaScript code must be provided in this field. With task-level validation, you cannot reference a file containing JavaScript code.

3. (Optional) Enable Validate On Change, so that the field validation occurs as soon as it is changed.

4. (Optional) For a user, group, or organization, you can use a Validate button on the profile tab. The Validate button is hidden by default. To make this button visible, clear the Hide Validate button option when you configure the task's profile tab.

   If Validate On Change is enabled on a field and the value of that field changes, the Validate button updates other fields on the screen.

   Note: The Validate button also executes Logical Attribute Handlers that include the validate method. For more information about Logical Attribute Handlers, see the *Programming Guide for Java*.

Directory-Level Validation validates fields based on the content of the directory.xml file.

Note: For more information on Directory-Level Validation or understanding the default validation included with Identity Manager, see the *Configuration Guide*.

User-defined Custom Attributes for Roles

Identity Manager supports user-defined custom attributes that allow you to specify additional information about roles. You can use this information to filter roles in your organization. For example, a corporate environment may have more than a thousand roles. That organization can specify additional information, such as business unit or geographical location, for each role. Administrators can then use that information to facilitate role searches.

You can use custom attributes in the Create, Modify, and View tasks for the following roles:

- Admin Roles
- Provisioning Roles
- Access Roles

To configure custom attributes for roles, you complete the following high-level steps:

1. Add support for custom attributes to the profile tab for the tasks that create, modify, or view admin roles, provisioning roles, or access roles.
2. Configure search and list screens for the roles to include the custom attributes.

More Information:

Configure Custom Attributes in Profile Tab for Roles (see page 77)
Add Custom Attributes to Search Screen Definitions (see page 78)

Configure Custom Attributes in Profile Tab for Roles

Identity Manager allows you to configure up to 10 custom attributes on the Profile tab of tasks that allow you to create, modify, or view roles.

To configure custom attributes in the Profile tab

1. Click Roles and Tasks, Admin Tasks, Modify Admin Tasks.

   The Select Admin Task page appears.
2. Search for and select the admin task that you want to modify.

   Identity Manager displays the task details for the selected admin task.
3. Click the Tabs tab.

   The tabs that are configured for use with this admin task appear.

4. Click the arrow icon to edit the Profile tab.

   The Configure Profile screen appears.

5. Select the checkbox next to each custom field to add to the Profile tab and enter a meaningful label.

6. Click OK.

   The custom attributes will be available in the Profile tab of the modified task after you submit the task.

   Note: To use the custom attributes in role searches, configure the search screen (see page 78) to display these custom attributes.

Add Custom Attributes to Search Screen Definitions

When you want to filter roles in Identity Manager, you can only use the attributes that are available in the search screen. To filter the roles based on the custom attributes that you have defined, you must add the custom attributes to the search screen of the roles.

To add Custom Attributes to the Search Screens of roles

1. Click Roles and Tasks, Admin Tasks, Modify Admin Tasks.

   The Select Admin Task page appears.

2. Search for and select that admin task that you want to modify.

   To add custom attributes to search screens, select the Modify or View task for the type of role (admin, provisioning, or access) that includes custom attributes.

   Identity Manager displays the task details for the selected admin task.

3. Click the Search tab in the Modify Admin Role screen.

   The search screen details appear.

4. Click the Browse button to display a list of search screen definitions that are available for the task.

   The Select Screen Definition page appears.

5. Select a search screen definition to edit, or create a copy of an existing search screen definition.

   The Configure Standard Search Screen appears.

6. Add the custom attributes to the following tables:

   - Select the fields that a user can search on
   - Select the fields that appear in the search results

7. Change the name of the custom attribute to match the name you specified when you configured the Profile tab.
8. Click OK to save the changes to the search screen definition.

   The Select Screen Definition page displays again.
9. Select the screen that you created or edited, then click Select.
10. Select All Admin Roles from the Search Options list.
11. Click Submit.

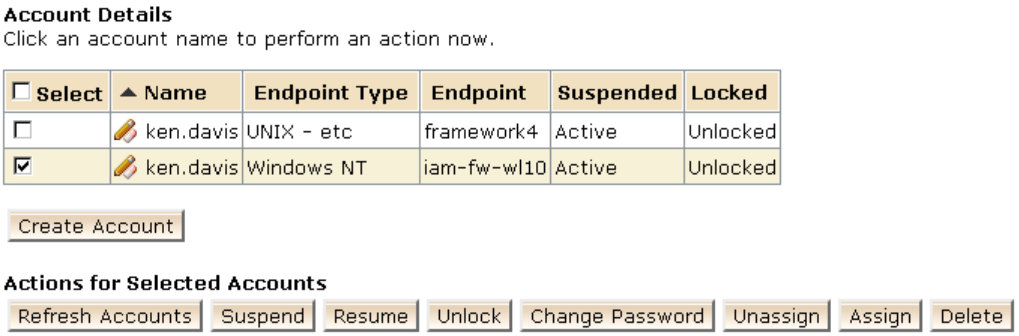
    The search screen will now include the custom attributes in the search options and display the attributes in the search results.

Chapter 4: Configuring Account Tabs

This section contains the following topics:

Account Tabs

The Accounts tab lists accounts in managed endpoints for users who have been assigned provisioning roles. Typically, this tab is added to tasks that allow you to view or modify a user.



When the Accounts tab is added to a Modify User task, administrators can perform other actions on the user's accounts. For example:

- Suspend or resume an account
- Unlock an account that has been automatically locked because of incorrect or inappropriate access. For example, an account may be locked when a user exceeds the acceptable number of failed login attempts set in a Identity Manager password policy.
- Change the user's password in one or more accounts.
- Assign and unassign accounts to a user.

For details on the other options you can provide on the Accounts tab, see the user console help for the Configure Accounts tab.

Prerequisite for Using the Accounts Tab

To use the Accounts tab, Identity Manager must be configured with provisioning support, and the Identity Manager environment must include a provisioning directory.

Note: To configure provisioning support for an environment, see the *Configuration Guide*.

Fields on the Accounts Tab

The Accounts tab displays details about the accounts the user has on endpoint systems.

The following are some of the more significant fields:

- Name—The login name, email name, or other name for the account.
- Endpoint Type—The type of endpoint, such as an LDAP directory, that is associated with the account.
- Endpoint—The specific endpoint that is associated with account.
- Suspended—One of three states.
  - Active appears if the account is enabled.
  - Suspended appears if the account is disabled.
  - Activation Pending (Manual) appears if it cannot be resumed or suspended. Log into the endpoint system to resume or suspend the account.
  - Unavailable appears if the state cannot be retrieved because of no communication with the endpoint.
- Locked—Shows if the account is locked. Locking occurs when a user makes several attempts to log into the account with the wrong password. Unavailable appears if the state cannot be retrieved because of no communication with the endpoint.

Additional Functions on the Accounts Tab

When the Accounts tab is included in a task that modifies a user, administrators can use that task to perform functions on the user's accounts. The available functions are determined by the tab configuration.

You can select which functions are available by using the Modify Admin Task on a tasks containing the Accounts tab. You edit the Accounts tab to determine if functions such as Assign Account and Unassign Account are available in the tab.

Note: See the online help for the Configure Accounts tab for more information.

Chapter 5: Search and List Screens

This section contains the following topics:

Search Screen Configuration

You configure a search screen to limit the scope of the task and control the fields that users can search on. Search screens apply to two types of objects:

- A *primary object*—The object to be modified or viewed by the task.
- A *secondary object*—The object that is related to the primary object.

  For example, if you include a group tab on a create user task, the user is the primary object and the group is the secondary object. The group tab needs a search screen for groups.

  Note: After configuring a search screen, you can use it for any task to search for a primary or secondary object.

Modify a Search Screen

You can modify an existing search screen to:

- Configure search filter defaults
- Modify the fields in search filters
- Modify the fields in search results
- Add help text on the search screen

To modify a search screen

1. In the User Console, select Roles and Tasks, Admin Tasks, Modify Admin Task.
2. Search for and select the admin task to modify.

   Identity Manager displays the tabs to configure for the task you selected.

3. Select the Search tab.
4. (Optional) Select the Modified objects must remain in administrator's scope check box.

   When this check box is selected, Identity Manager displays an error if changes to the task cause the administrator to lose scope over the primary object. For example, an administrator may use Modify User to change a user's Employee Type attribute to Manager. This change may put the user outside the administrator's scope.

   Note: This option does not appear for tasks that manage roles.
5. Click Browse next to the Screen field.

   Identity Manager displays a list of applicable screens.
6. Select the search screen that you want to modify or copy and then click one of the following buttons:

   - **Select**

     Adds the selected screen to the search that you are configuring
   - **Edit**

     Opens a new screen where you can change the settings, including fields, field properties, and layout for the selected screen
   - **Delete**

     Deletes the selected screen
   - **New**

     Opens a new screen where you can create a screen. The new screen does not include any default fields.
   - **Copy**

     Creates a new screen using the settings from an existing screen. To create a screen which is based on an existing screen, you add a new name and tag to the screen, and modify the settings as needed.
   - **Cancel**

     Returns you to the Search configuration screen.

If you selected Edit, New, or Copy, Identity Manager opens a new screen where you can create or modify a search screen.

Search Filters

Search filters limit which objects the search returns. For example, if the object is users, you can limit the search to find only contractors. You can configure a filter to find users with the Employee Type of Contractor.

You can configure the following fields for searches:

Show only objects meeting the following rules

Defines additional criteria to be combined with the user-defined filter to constrain the search.

Note the following when using this field**:**

- Due to limitations with provisioning roles searches, these criteria override filter fields with the same name entered by the user.
- Attributes that are used when you configure this field should not be added as available search fields on the search screen.

  For example, if you configure the search screen to display only roles where the Enabled attribute is set to Yes, remove the Enabled attribute from the list of attributes that users can specify in search criteria.

  Otherwise, the user-entered criteria is ignored.

Default search filter

Defines a filter that appears by default when an administrator uses the search screen. For example, if you are configuring a search screen for the Modify Contractor task and you know that administrators typically search for contractors based on the contract firm name, you can set the default filter to Contract Firm = *. Administrators can override the default filter by specifying different search criteria. Setting a default filter improves performance by limiting the number of results returned if an administrator does not specify a filter before beginning a search.

Auto select all search results when used with multi-select tasks

Specifies that all search results are selected by default. If you select this check box, all the objects in the search results list appear with a checked box next to the object name.

Automatically perform search

Specifies that a search field is displayed with the search results.

Automatically set subject of task when there is only a single search result

Sets the primary object of the task automatically when only one object matches the search filter.

For example, suppose that this option is selected for a user search screen which is associated with the Modify User task. When an administrator opens the Modify User task and enters a search filter that returns only one user, Identity Manager opens the Modify User task for that user. The administrator does not have to select the user to open the Modify User task.

Note: For this setting to apply, Automatically perform search must also be selected.

Save search filter

Specifies that the search filter for the task is saved for the user in the current session. The next time that user searches in the task, the saved search filter will be displayed.

Note: Identity Manager saves the search filter for the duration of the user session. When the user logs out, the search filter is cleared.

Search in organization

Displays an organization filter on the search screen. If this check box is selected, administrators can specify a filter that limits the organizations in which Identity Manager searches for an object. You can specify defaults for the organization search filter by specifying a search screen in the Organization Search field.

Save search organization

Specifies that the organization for the task is saved if an organization was established for the search. The next time a user searches in the task, the organization will be displayed.

Organization Search

Specifies the search screen that Identity Manager uses to allow administrators to search for an organization.

Default Organization Search Scope

Specifies the default organization search scope that appears when an administrator uses a search screen. The search scope determines the levels in an organization tree that are included in the search. Administrators can override the default organization search scope by specifying different search criteria on the search screen.

For example, if you configure a search screen for a custom Modify Contractor task in an environment that stores contractor information at various levels in the organization tree, you can set the default organization search scope to And Lower.

Single expression search

Defines the type of search filter that appears on the search screen. When you select this checkbox, users can specify a single search filter, such as <attribute><comparator><value>. When you clear this checkbox, users can specify multiple search filters. For example, <attribute1><comparator><value1> AND <attribute2><comparator> <value2>. Objects that meet the conditions in all the filters are returned in the search results. In the previous example, objects that include <value1> and <value2> would be returned as search results.

Equals Only Search

Prohibits administrators from using search operators other than equals.

Display the number of results

Displays the number of matching search results. When this check box is selected, all searches return the message, "There are X number of results".

Add task button for <task name>

Adds a link to another task to the search screen. The link is displayed as a button. This field is typically used to add a Create task to a search screen that is configured for object-task navigation.

Optional label

Specifies a label for the task that you selected in the previous field. This label appears on the button for the task.

Add multi-delete button for <task name>

Adds a link to a task that allows administrators to select multiple objects to delete. The link is displayed as a button.
This field is typically with object-task navigation.

Search Fields and Search Results

On another part of the search screen, you select fields that an administrator can use in a search query and fields to display in search results.

Select the fields that a user can search on

Select the fields that an administrator can use to create a search query.

To add additional fields, select the fields in the list box below the search fields table.

After you select the fields, you can change the order in which they appear by using the up and down arrow icons to the right of the field.

Note: If you do not specify fields that an administrator can search on, Identity Manager starts the search automatically.

Select the fields that appear in the search results

Select the fields that Identity Manager displays in the search results. You can select fields that are not available in the search query.

To add additional fields, select the fields in the list box below the search fields table.

Style

When you select a field to display in the search results, you can select one of the following style options:

- **Boolean Display Name**

  Displays the name of the field for all results that are true. For example, if you enter Enabled as the name of the attribute that indicates a user's account status, "Enabled" would appear in the search results for all active user accounts.

- **Checkmark**

  Displays the value as a selected check mark, based on the value of the attribute. For example, if you select the check mark style to represent the Enabled/Disabled state of user accounts, Identity Manager displays a selected check mark for all active accounts.

- **Multi-Value String**

  Displays the values in a multi-value attribute on separate lines. The values are listed alphabetically.

- **Read-Only Checkbox**

  Displays the value as a read only checkbox.

- **String**

  Displays the value as a text string.

- **Task**

  Adds a task list to a field. Users click an arrow icon to see a list of tasks that they can perform on the object associated with the search field. For example, if you add a task list to a Last Name field in the search results, users can click on the arrow icon in that field to see a list of tasks they can perform on the user they select.

  This setting can also be used to make an attribute value appear as a link to a task.

  If you select the Task style, a right arrow icon appears next to the Style column. Click the arrow to open a Field Properties dialog. Use this dialog to configure a task list (see page 96).

- **Task List**

  Adds additional tasks that users can perform on objects in search and list screens. For example, you can configure the search screen in the Modify User task to enable users to perform a task, such as disabling a user, from the list of users returned by the search.

  When you select this option, you determine whether users access the task by clicking an icon, or a text link.

- **Task Menu**

  Adds additional tasks (similar to the Task List style) as pop-up menu items.

  When you select this option, an Action button appears next to each object in a search or list screen. Users click the Action button to see the list of tasks they can perform for that object.

Note: To see the Task List and Task Menu style options, select (Separator) when you add a field to the search results table. For more information about adding additional tasks to search and list screens, see the *User Console Design Guide*.

Sortable

Select this checkbox to allow administrators to sort search results by a field or fields.

Set the default sort order for the search results

Specifies the order in which search results are displayed. Search results are sorted initially by the first field in the list and then by each additional field in the order in which they appear. Select the Descending checkbox to sort the results in descending order.

Select objects with changes to field *name*

Specifies that objects in which the specified field has changed are selected when the user clicks the Select button.

Return *N* results per page

Select the number of results to display per page. When search results exceed the number you specify, Identity Manager displays a link to each page of results.

User-Defined Help on Search Screens

If you want to add custom text to your search screen, you can define text in the corresponding HTML text box. You can add text in the following areas:

- Beginning or end of the page
- Before or after the create
- Before or after the results

Types of Search Screens

Identity Manager includes these pre-configured search screens.

Access Role Search Screen

The Access Role Search Screen lets you configure search filters to find access roles that match specific criteria.

Access Task Search Screen

The Access Task Search Screen lets you configure search filters to find access tasks that match specific criteria. This search screen is used to find an access task to view or modify, or to add a task to an access role.

Admin Role Search Screen

The Admin Role Search Screen lets you configure search filters to find admin roles that match specific criteria.

Admin Task Search Screen

The Admin Task Search Screen lets you configure search filters to find admin tasks that match specific criteria. This search screen is used to find an admin task to view or modify, or to add a task to an admin role.

Approval Search Screen

The Approval Search Screen lets you configure the display that appears at the top of an approval task.

Begin Certification User Search Screen

The Begin Certification User Search Screen lets you configure search filters to find users to set to require certification. Users selected will have their certification status set to *requiring certification*.

Certify User Search Screen

The Certify User Search Screen lets you configure the search filters to find users who require certification.

Delegation Search Screen

The Delegation Search Screen lets you configure search filters to find additional users to add as delegates. A delegate is another user that you can temporarily grant permission to view and resolve your workflow work items.

Enable/Disable User Search Screen

The Enable/Disable User Search Screen lets you configure search filters to enable/disable users who match specific criteria.

EndCertification User Search Screen

The EndCertification User Search Screen lets you configure search filters to identify users whose certification cycle should be completed.

End User License Agreement Search Screen

The End User License Agreement Search Screen lets you configure the Self Registration task with a page that is specific to your identity-based application.

Explore and Correlate Search

The Explore and Correlate Search Screen lets you configure search filters for explore and correlate definitions that match specific criteria.

Feeder File Upload Search

The Feeder File Upload Search Screen lets you browse for the feeder file to upload. A feeder file is used to automate repeated actions performed on large number of managed objects.

Forgotten Password Search Screen/Forgotten User ID Search Screen

The Forgotten Password Search Screen lets you configure the Forgotten Password task to prompt users for information that verifies their identity.

Group Search Screen

The Group Search Screen lets you configure search filters for groups, such as groups within the finance organization.

Identity Policy Set Search Screen

The Identity Policy Set Search Screen lets you configure search filters to find identity policy sets that match specific criteria.

Logical Attribute Handler Search Screen

The Logical Attribute Handler Search Screen lets you configure search filters to find logical attribute handlers. This search screen is used to find a logical attribute handler to view or modify its configuration.

Manage Reports Search Screen

The Manage Reports Search Screen lets you configure search filters to find a report to view or delete.

NonCertified User Search Screen

The NonCertified User Search Screen lets you configure search filters to find users who were not certified by the end of the certification period.

Organization Search Screen

The Organization search screen lets you configure search filters to limit the choice of organizations to certain sub-organizations.

Provisioning Role Search Screen

The Provisioning Role Search Screen lets you configure the search filters for retrieving provisioning roles.

Account Template Search Screen

The Account Template Search Screen lets you configure the search filters for retrieving account templates.

Password Policy Search Screen

The Password Policy Search Screen lets you configure the search filters to find password policies that match specific criteria.

Snapshot Definition Search Screen

The Snapshot Definition Search Screen lets you configure the search filters to find a snapshot definition to view, modify, or delete.

Standard Search Screen

The Standard Search Screen lets you configure filters to find custom managed objects.

User Search Screen

The User search screen lets you configure search filters to find users that match specific criteria. For example, you can search for users who are contractors.

Once you complete the Search tab, Choose Tabs for the Task.

List Screens

In configuring tabs, you often need to show a list of items, such as a list of users or roles. The list appears on the tab that you are configuring. In these situations, create a List Screen to control the columns and sorting of the objects on the tab.

You can configure the following fields for a List Screen:

Name

Defines the name of the task.

Tag

An identifier that is unique within the task. It can contain ASCII characters (a-z, A-Z), numbers (0-9), or underscore characters, beginning with a letter or underscore. The tag is used for setting data values through XML documents or HTTP parameters.

Field

Specifies the attributes that appear as fields in the search results.

Name

Specifies the label for the field in the search results.

Style

Determines the format of the field in the search results. You can specify the following style options:

Boolean Display Name

Displays the name of the field for all results that are true. For example, if you enter Enabled as the name of the attribute that indicates a user's account status, "Enabled" would appear in the search results for all active user accounts.

Checkmark

Displays the value as a selected or deselected checkmark based on the value of the attribute. For example, if you select the checkmark style to represent the Enabled/Disabled state of user accounts, Identity Manager displays a selected checkmark for all active accounts.

Multi-Value String

Displays the values in a multi-value attribute on separate lines. The values are listed alphabetically.

Read-Only Checkbox

Displays the value as a read only checkbox.

String

Displays the value as a text string.

Task

Adds a task list to a field. Users click a right arrow icon to see a list of tasks that they can perform on the object associated with the search field. For example, if you add a task list to a Last Name field in the search results, users can click on the arrow icon in that field to see a list of tasks they can perform on the user they select.

Sortable

Determines whether users can sort search results based on the selected field.

Descending

Determines the order in which search results are displayed. When the Descending checkbox is selected, the search results are sorted alphabetically in descending order. The results are sorted in the order in which they appear in the list.

Results per page

Indicates the number of search results to display in the search results.

Enter HTML to appear before the list

Specifies text that appears above the list of search results.

Enter HTML to appear after the list

Specifies text that appears below the list of search results.

You can also add text above and below a list screen.

Add a Task List

A task list is a menu of tasks that you access from a list of objects, such as a list or search results screen. Task lists allow you to view and use the tasks that apply to an object without having to search for that object each time you use a new task. For example, you can configure Identity Manager to display a task menu for each role member listed on the Membership tab of the Modify Admin Role Members task. Administrators can use the task menus to manage role members without having to perform a new search for each role member.

To add a task list

1. Complete *one* of the following steps:
   - Select Modify Admin Task from Roles and Tasks, Admin Tasks. Search for and select the admin task to modify.
   - Select Create Admin Task from Roles and Tasks, Admin Tasks. Then, select Create a copy of an Admin Task and search for a task to copy.

   Identity Manager displays the tabs to configure for the task you selected.
2. Select the tab where you want to add the task list.

   Typically, this is a tab that includes a search or list screen, such as the Membership tab.
3. Search for a list or search screen to edit by clicking Browse.
4. Select the field for the task list from the list of fields that appear in search results.
5. Select Task in the Style field.
6. Click the right arrow icon to open a Field Properties section where you can configure the task list.
7. Complete the following fields as needed:
   - **Default Task**

     Specifies the task that opens when a user clicks a value in the field. When you configure a field to support task lists, and specify a default task, the field value appears in blue text, indicating that it is a link.

     For example, if you configure the Last Name field to include a task list, an administrator can click a user's last name to open the default task.
   - **Alternate Task**

     Specifies the task that opens when a user clicks the field value and does not have privileges to use the default task.

- **Enable popup task menu**

  Displays a right arrow icon next to the field. Users click the icon to view the list of tasks they can perform on that object in that field.

  When you select this checkbox, the following options appear:

  - **Include all tasks that the administrator can perform on the object**
  - **Include all tasks that the administrator can perform on the object unless hidden in menus**
  - **Include only the specified tasks**

    Displays only tasks that you select in the Task field.

    Note: Users will not see a specified task if they do not have privileges to use it.
  - **Exclude the specified tasks**

    Displays the tasks that an administrator can perform on the object *except* tasks listed in the Task field.
  - **Task**

    Specifies the tasks that appear or do not appear in a task list, depending on whether the Include Only Specified Tasks or Exclude the Specified Tasks checkbox is selected.

- **Nest Task**

  When checked, specifies that Identity Manager should open the task as a nested task. When users complete the nested task, they are returned to the original task.

  If this option is not selected, the new task replaces the original task.

8. Click OK.

Additional Tasks in Search and List Screens

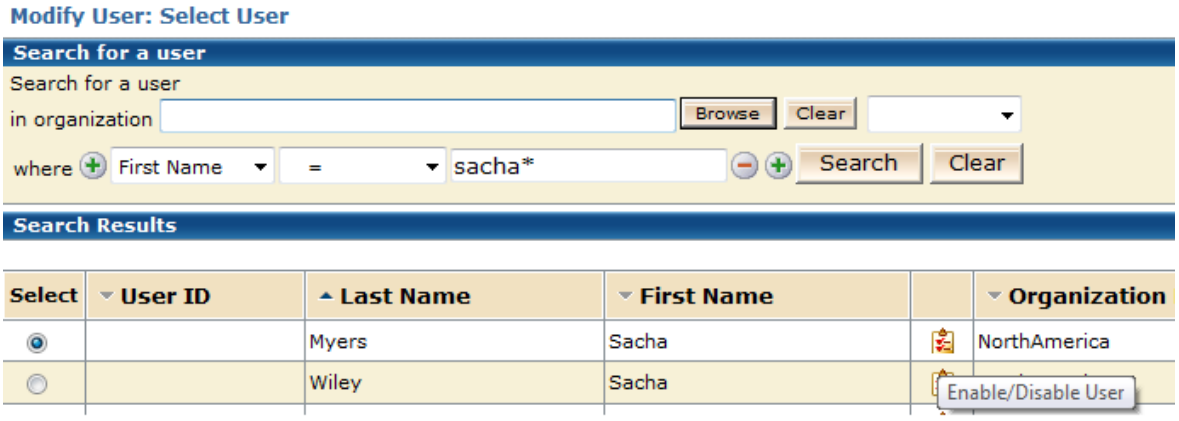
You can configure Identity Manager to add additional actions that users can perform in search and list screens. For example, you can configure the search screen in the Modify User task to enable users to perform a task, such as disabling a user, from the list of users returned by the search.

Adding tasks to search and list screens reduces the number of clicks required to complete a task, and simplifies the user console.

Tasks on search and list screens can be displayed using one of the following methods:

- Task links or icons

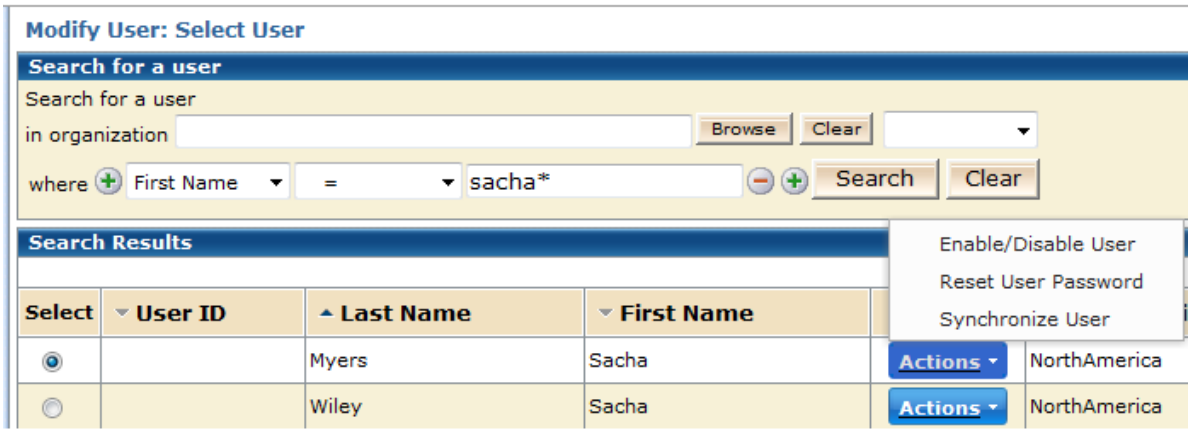
  Displays each task as a links or icon in the search results or list screens. Use this method to display a small number of tasks.



- Task Menus

  Displays an Action button in each row in search results or list screens. Administrators click the Action button to see the list of tasks that they can perform for that user.

  Use this method if users are able to perform more than two or three tasks.



Add Additional Tasks to Search and List Screens

You can configure Identity Manager to launch additional tasks from search or list screens to reduce the number of steps users take to complete certain tasks.

To add additional tasks to search and list screens

1. Modify a search or list screen.

   The Configure Standard Search Screen window appears.

2. Add a new row in the search results fields section as follows:

   a. Add a new row by clicking the plus icon below the search results table.

   b. Select the separator style.

   c. Select one of the following options, and then click the edit icon to configure the additional tasks:

      – Task Link

        Displays the additional tasks as icons or text links.

      – Task Menu

        Displays an Action button that users click to view a menu of the tasks they can perform.

3. If you selected Task Link, complete the following steps:
   a. Specify the task that opens when users click the task icon or link in the Default Task field.
   b. Specify an alternate task that opens if users do not have privileges to open the default task.
   c. Determine how Identity Manager opens the task by selecting or clearing the Nest Task field.

      When this option is selected, the task opens as a nested task. When users complete the nested task, they return to the search or list screen.
   d. Determine whether the additional tasks will be displayed as icons or text links be selecting or clearing the Task Icon field.

      If you clear this option, Identity Manager displays the task as a text link.
4. If you selected Task Menu, complete the following steps:
   a. Select the type of tasks that Identity Manager displays in the task menu.
   b. Specify the tasks to display *if* you selected the Include Only the Specified Tasks or Exclude Specified Tasks options in step a.
   c. Determine whether Identity Manager opens menu tasks as nested tasks by selecting or clearing the Nest Task option.

      When you select the Nest Task option, Identity Manager returns users to the location where they launched the task when the additional task completes.
5. Click OK, then click Select.
6. Click OK, Submit to save changes to the screens.

Chapter 6: Self-Service Tasks

This section contains the following topics:

Identity Manager Self-Service Tasks

Self-service tasks are Identity Manager tasks that users can use to manage their own profiles. These tasks are divided into two types:

- Public tasks--Tasks that users can access without providing login credentials. Examples of public tasks are self-registration, forgotten password, and forgotten user ID tasks.
- Protected tasks--Tasks for which users provide valid credentials. Examples include tasks for changing passwords or profile information. To gain access to these tasks, users must be given a role, such as the Self Manager role.

The following table lists the default self-service tasks, which are available when Identity Manager is installed.

Task Type	Tasks
Public Task	■ Self-registration--Allows users to register at a corporate Web site ■ Forgotten Password Reset--Allows users to reset a forgotten password ■ Forgotten Password--Displays a temporary password that users can use to login to Identity Manager. When the users log in, they are prompted to enter a new password ■ Forgotten User ID--Retrieves or resets a forgotten user ID

Task Type	Tasks
Protected Task	■ Change My Password--Allows users to reset their password ■ Modify My Profile--Maintains profile information, such as address and phone number ■ Modify My Groups--Enables users to subscribe to groups ■ View My Roles--Displays a user's roles ■ View My Submitted Tasks--Displays Identity Manager tasks that the user initiated

How to Configure Self-Service Tasks

The following table describes the steps to configure self-service tasks for Identity Manager environment. Some of the steps are optional.

Step	Refer to...
1. Configure a public alias in the Management Console to allow users to access public tasks, such as the self-registration, forgotten password reset, and forgotten user ID tasks.	*Configuration Guide*
2. Configure the self-service tasks that apply in your environment.	■ Configure the Self-Registration Task (see page 103) ■ Configure the Forgotten Password Reset and Forgotten User ID Tasks (see page 105)
3. Customize the self service tasks for your environment.	Customize Self-Service Tasks
4. Add links for accessing self service tasks to your corporate Web site.	Access Self Service Tasks

5. Configure the Self Manager role. (Optional). By default, the Self Manager role is assigned to all users. Complete this step only if you want to restrict the users who have access to the role.	*Administration Guide*

Configure the Self-Registration Task

To provide self-registration for users, first make sure that you have an alias for public tasks for the Identity Manager environment. (See the *Configuration Guide*). Then, configure the self-registration task.

Note: To avoid overwriting the default Self Registration task, create a copy of the task. Customize the new task as needed.

1. In the User Console, choose Roles and Tasks, Admin Tasks, Modify Admin Task.
2. Select the Self Registration task.
3. On the Search tab, select the End User License Agreement screen by clicking Browse.

   Edit the screen to present an appropriate title and a Message URL.

   For the Message URL, use a page that you create to request that new users agree to license restrictions for your application.
4. On the Tabs tab, edit the Profile and Groups tabs as needed:
   - If the Identity Manager environment supports organizations, supply a default organization where self-registered users' profiles are stored. (see page 104)
   - If the default tasks do not suit your business requirements, customize the profile and list screens.
   - If the Identity Manager environment includes forgotten passwords or forgotten User ID support, add fields for collecting password questions and answers (see page 104).

Set Up a Default Organization for Self-Registered Users

If your Identity Manager environment supports organizations, you can specify the organization where Identity Manager creates accounts for self-registered users.

Note: To store profiles for different types of users, such as customers and suppliers, in different environments, create multiple self-registration tasks with different default organizations. For example, if customers self-register in the customers organization, and suppliers register in the supplier organization, create two self-registration tasks, such as Customer Registration and Supplier Registration. In each task, define the appropriate default organization.

1. Navigate to the Configure Profile screen for the Self-Registration task if necessary:
    a. In the User Console, choose Roles and Tasks, Admin Tasks, Modify Admin Task.
    b. Select the Self Registration task.
    c. Select the Tabs tab.
    d. Click the right arrow next to the Profile tab.
2. On the Configure Profile screen, click Browse next to Default Organization.
3. Select the organization where new users should be created.
4. Save your changes.

Add Verification Questions and Answers

To enable users to specify question and answer pairs, which can be used to retrieve a forgotten password or user ID, add question and answer fields to the self registration screen.

Note: Before adding question and answer fields to collect verification information, verify that the logical attributes for the question and answer pairs are configured in the forgotten password logical attribute handler. You configure logical attribute handlers in the User Console or Management Console. For more information, see the online help in the console that you want to use.

To add verification questions and answers

1. Navigate to the Configure Profile screen for the Self Registration task if necessary.
    a. In the User Console, select Roles and Tasks, Admin Tasks, Modify Admin Task.
    b. Select the Self Registration task.
    c. Select the Tabs tab.
    d. Click the Edit icon next to the Profile tab.

2. On the Configure Standard Profile screen, click the Browse button next to the Screen field.

   The Select Screen Definition screen opens.

3. Select the Self Registration Profile and click Copy.

4. Supply a new name and tag for the custom self registration profile screen that you are creating.

   The tag can contain ASCII characters (a-z, A-Z), numbers (0-9), or underscore characters, beginning with a letter or underscore.

5. Add the number of rows and fields that you want to appear for the verification questions and answers.

   For example, if users should supply two question/answer pairs, add two rows of two fields.

6. In the field properties for the first question, select |Question 1| from the list of available attributes. Configure the field properties as needed.

   Note: If the ForgottenPasswordHandler logical attribute handler is configured to display a list of questions that users can select, specify the Option Selector style.

7. Repeat step 6 for each of the new fields that you added.

8. Click Apply.

   The Select Screen Definition screen opens again.

9. Verify that the screen definition is selected and click Select.

   The Configure Profile screen appears.

10. Click OK to close the Configure Profile screen and return to the Tabs tab.

Configure the Forgotten Password Reset and Forgotten User ID Tasks

Identity Manager includes default tasks for users who cannot access their accounts due to a forgotten password or user ID:

- The Forgotten Password Reset Task (see page 106)
- The Forgotten User ID Task (see page 106)

You can use these tasks as installed or customize them to suit your needs.

The Forgotten Password Reset Task

The Forgotten Password Reset task enables a user to reset a password after Identity Manager verifies his identity. Identity Manager uses two types of questions to verify a user's identity:

- Identification questions--Determine who a user is. Examples include a user's full name, user ID, or email address.
- Verification questions--Confirm a user's identity. Depending on how Identity Manager is configured, users can specify their own verification questions, or they can select questions from a predefined list.

In the default Forgotten Password Reset task, a user must provide a user ID and answer five verification questions. Each verification question, which is presented on a separate screen, is randomly chosen from a list of five questions that the user supplies during registration.

Once Identity Manager verifies a user's identity, a screen where the user can enter a new password is displayed.

The Forgotten User ID Task

In the default Forgotten User ID task, a user must provide an email address and answer one verification question to view their user ID in the User Console. The verification question, which is presented on a separate screen, is randomly chosen from a list of five questions that the user supplies during registration.

Custom Forgotten Password Reset and Forgotten User ID Tasks

You can use the Forgotten Password Reset or User ID task as installed, or customize the task for your environment. You can:

- Specify the number of questions (see page 107) users must answer successfully to verify their identity.
- Determine whether users supply their own verification questions (see page 107), or whether they select questions from a pre-defined list.
- Define the presentation (see page 110) of the verification questions on the screen.
- Require users to provide additional information, such as a social security number, to verify their identity (see page 112).

- Determine how users receive their password (see page 114) or user ID (see page 115).
- Specify criteria, such as failing more than three verification attempts, for locking a user out of the task (see page 112).

Note: The Forgotten Password Reset task should often not be configured for outbound synchronization. The temporary password may not match the password composition rules on each account associated with the provisioning user. For this reason, the ForgottenPasswordEvent is not included in the default Provisioning Outbound Mappings

Collect Question and Answer Pairs for User Verification

Users must supply the question and answer pairs that are used to verify their identity.

You can allow users to create their own questions, or require them to select predefined questions from a list.

To configure Identity Manager to collect question and answer pairs, complete the following actions:

- Add fields for collecting the questions and answers to the Self Registration, Modify My Profile, and Change My Password tasks (see page 103).
- Configure the ForgottenPasswordHandler handler in the User Console or Management Console. For configuration instructions, see the online help in the console that you want to use.

Set Up the Forgotten Password Reset or User ID Task

The configuration for the Forgotten Password Reset and Forgotten User ID tasks is similar.

To configure these tasks

1. Verify that the following items are configured in the Management Console:
   - Public Alias

     A text string that Identity Manager adds to the URL for accessing public tasks, including the Forgotten Password Reset and Forgotten User ID tasks.

     Note: See the *Configuration Guide* for more information.

- ForgottenPasswordHandler

  A logical attribute handler that enables users to create one or more verification questions, or choose questions from a predefined list.

  See the *Programming Guide for Java* for more information.

Note: You can also configure the ForgottenPasswordHandler in the User Console. Click the Help button in the User Console for more information.

2. In the User Console, do one of the following:
   - To create a copy of the Forgotten Password Reset or Forgotten User ID task (recommended), select Roles and Tasks, Admin Tasks, Create Admin Task. Select Create a copy of an admin task, and search for the task to copy.
   - To modify the default task, select Roles and Tasks, Admin Tasks, Modify Admin Task. Search for the task to modify.

   Identity Manager displays the tasks that match the criteria you entered.
3. Select the Forgotten Password Reset or Forgotten User ID task.
4. On the Search tab, click Browse to display a list of screens to edit.
5. Select one of the following screens, and click Edit:
   - Forgotten Password Search
   - Forgotten User ID Search
6. Configure the following based on your needs:
   - Identification screen

     Determines who a user is. This is the first screen that users see when they access the Forgotten Password Reset or Forgotten User ID tasks.
   - Verification screen(s)

     Presents one or more verification questions to users.
7. Enter the number of questions users must answer to verify their identity.

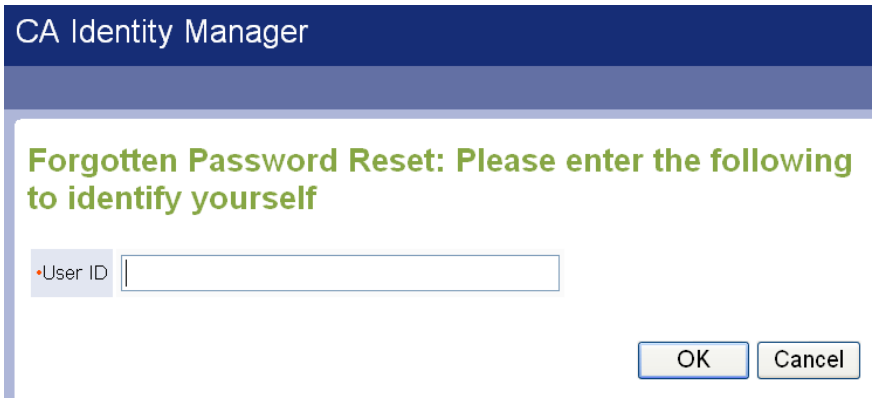
   Note: If you configure Identity Manager to display multiple verification questions on a single screen, the number of questions is determined by the logical attribute handler associated with the task. The Number of Questions setting does not apply.
8. Configure the criteria for locking the Forgotten User ID or Password task.
9. Submit the task.

Design Identification Screens

The identification screen is the first screen that users see when they access the Forgotten Password Reset or User ID task.

The default identification screen prompts users to supply a user ID. You can add or change the fields on the identification screen to suit your needs.

The default identification screen resembles the following:



To customize the identification screen

1. Navigate to the Configure Forgotten Password Search screen or the Forgotten User ID Search screen in the Identity Manager User Console, if necessary:
   a. Choose Roles and Tasks, Admin Tasks, Modify Admin Task.
   b. Select the Forgotten Password Reset or User ID task.
   c. On the Search tab, click Browse to display a list of screens to edit.
   d. Select one of the following screens, and click Edit:

      Forgotten Password Search

      Forgotten User ID Search
2. Enter the text that will appear above the area where users supply account information in the Prompt field.
3. Select the appropriate screen in the Profile Screen for Identification field.
4. Modify the screen to include your choice of attributes that users must enter.

Design Verification Screens

After a user successfully completes the identification screen, he is redirected to a verification screen where he must provide information to verify his identity. The user may be required to answer one or more questions, or provide an attribute, such as a social security number.

If users must answer multiple verification questions, Identity Manager can display those questions on the same screen, or on separate screens.

The default verification screen resembles the following:

Display Multiple Verification Questions At One Time

If users answer multiple questions to verify their identity, you can display those questions on a single screen.

Note: If a single screen displays multiple questions, the number of questions that a user has to answer is determined by the number of question and answer pairs that you add to the profile screen for primary verification, not the number of questions that you configure in the search screen for the task.

To display multiple verification questions on a single screen

1. Configure the Forgotten Password Logical Attribute Handler for multiple question and answer pairs.

   You can configure the ForgottenPasswordHandler in the User Console or the Management Console. For instructions, see the online help in the console that you want to use.

   Add |VerifyQuestion1| , |VerifyAnswer1| pairs depending upon the number of questions you want to set.

2. Navigate to the Configure Forgotten Password Search screen or Configure Forgotten User ID Search screen, if necessary.

3. Enter the text that appears above the area where users supply verification information in the Prompt for Primary Verification Screen field.

4. In the Profile Screen for Primary Verification field, select a screen definition, such as the Forgotten Password Verify screen.
5. Modify the screen definition to include the Logical Attributes for each of the verification question and answer pairs that should appear on the screen. For example, add fields as follows:

   |VerifyQuestion1| - Read only.

   |VerifyAnswer1| - Write Required.

   Note: For more information, see the online help for the ForgottenPasswordLogical Attribute Handler.
6. Make sure that the Prompt for Secondary Verification Screen and Profile Screen for Secondary Verification fields are blank in the Configure Forgotten Password Search or Configure Forgotten User ID Search screen.
7. Enter the number of questions that user must answer correctly in the Number of Questions field.
8. Click OK.

Display One Verification Question at a Time

For increased security, you can display only one verification question at a time. Subsequent questions are displayed only after the preceding question is answered successfully.

To display each verification question on a separate page, define a Primary Verification Screen and a Secondary Verification Screen.

The Primary Verification Screen is displayed after users provide valid identification, such as a user ID. When the user successfully answers one question on the primary verification screen, Identity Manager displays the secondary verification screen for each remaining question.

To configure the primary and secondary configuration screens:

1. Make sure that the |VerifyQuestion| and |VerifyAnswer| logical attributes are configured in the ForgottenPasswordHandler logical attribute handler. See the *Programming Guide for Java*.
2. Navigate to one of the following screens, if necessary:
   - Configure Forgotten Password Search Screen
   - Configure Forgotten User ID Search Screen
3. Enter the text that appears above the area where users supply verification information in the Prompt for Primary Verification Screen field.

4. In the Profile Screen for Primary Verification field, select a screen definition, such as the Forgotten Password Verify screen.

   Note: Modify the screen definition to include the Logical Attributes for each of the question and answer pairs that should appear on the screen.

5. Enter the text that appears above the area where users supply verification information in the Prompt for Secondary Verification Screen field.

6. Select the Forgotten Password Secondary Verify screen in the Profile Screen for Secondary Verification field.

   Modify the screen to include |VerifyQuestion| and |VerifyAnswer| logical attributes.

   Note: To use a secondary verification screen, you must configure a primary verification screen.

7. Enter the number of questions that user must answer correctly in the Number of Questions field.

8. Click OK.

Verify a User Attribute

Identity Manager can verify a user identity by requiring the user to supply one or more profile attributes. You can require these attributes in addition to verification questions, or instead of them.

To use user attributes in the verification process

1. Configure the verification screen as described in one of the following sections:
   - Display Multiple Verification Questions At One Time (see page 110)
   - Display One Verification Question at a Time (see page 111)
2. Add one or more fields to collect the user attribute in the Forgotten Password Verify screen, or in a custom primary verification screen, if you designed one.

   For example, to collect a user's employee number in addition to a user ID, modify the Forgotten Password Identify profile screen. Add one row containing a single field before or after the user ID field. Click the right arrow for the new field to define its properties.

Lock the Forgotten Password Reset or Forgotten User ID Task

To secure the Forgotten Password Reset or Forgotten User ID task, you can limit the number of failed verification attempts a user makes. Once a user exceeds the failed attempt limit, the task locks, and the user can no longer access it.

You can determine what Identity Manager considers a failed verification attempt. The definition of a failed attempt may be very strict, such as answering one verification question incorrectly, or more lenient to allow for mistakes, such as mis-typing an answer.

Note: You can also configure Identity Manager to lock the Forgotten Password Reset or Forgotten User ID task after a specified number of successful verification attempts (see page 114).This prevents users from using the Forgotten Password Reset or Forgotten User ID task instead of remembering login credentials.

Configure a Failed Attempt Limit

To configure Identity Manager to lock the Forgotten Password Reset or Forgotten User ID task after failed verification attempts:

1. Navigate to the Configure Forgotten Password Search Screen, if necessary.
2. Configure the criteria for verification failure, as needed:
   - Number of acceptable incorrect answers--The number of incorrect answers a user can provide before Identity Manager records a verification failure.
   - Verification page timeout--The amount of time a user has to answer all of the questions on a page.

     Verification page attempt limit--The number of times a user can attempt to answer the questions on a page.

     If only one question appears per page, the Verification page attempt limit is the number of times a user can try to answer that question.

   Note: Specify 0 for the options that do not apply.

   If a user exceeds any of the specified criteria, Identity Manager records a verification failure.
3. In the Failed Attempt Limit field, enter the number of consecutive times a user can fail the verification process before they are locked out of the task.

   Identity Manager locks the user out of the task, and optionally disables the user's account, if the user attempts to verify his identity when the Failed Attempt Limit has been reached. For example, if the failed attempt limit is 3, the user is locked and disabled on the third failed attempt.
4. Select the Disable User check box to disable a user's account in addition to locking the task when the failed attempt limit is exceeded.

5. In the Failed Attempt Lockout Length field, enter the length of time that a user is locked out of the task if they exceed the failed attempt limit.

   You can specify minutes, hours, and days. To indicate that a particular limit does not apply, enter 0.

   Note: The attribute you specify must be defined in the directory configuration file (directory.xml) for the Identity Manager environment.

6. Select the attribute that Identity Manager will use to track verification attempts in the Attempt Tracking Attribute field.

Configure a Successful Attempt Limit

Limiting the number of successful verification attempts prevents users from misusing the Forgotten Password Reset or Forgotten User ID task. For example, a user may rely on the Forgotten Password Reset task to reset a password instead of having to remember a password that conforms to a strict password policy.

To limit successful verification attempts:

1. Navigate to the Configure Forgotten Password Search Screen, if necessary.
2. Select the attribute that Identity Manager will use to track verification attempts in the Attempt Tracking Attribute field.
3. Enter the number of days that users must wait before using the task in the Successful Attempt Limit field.

Determine How Users Reset Passwords

Once Identity Manager verifies a user's identity in the Forgotten Password task, it performs *one* of the following actions:

- Redirects users to a screen where they can enter a new password. (default)
- Emails or displays a temporary password. Users can use the temporary password to log into Identity Manager, where they are forced to set a new password.

To configure Identity Manager to display or email a temporary password, use the Forgotten Password task instead of the Forgotten Password Reset task.

The Forgotten Password task is associated with a business logic task handler, a Java object that forms custom business logic, which generates a temporary password.

By default, the Forgotten Password task displays the temporary password in the User Console.

To configure the Forgotten Password task to email the temporary password:

1. In the Management Console, configure email notifications for the Identity Manager environment. See the *Configuration Guide* for instructions.
2. In the User Console, choose Roles and Tasks, Admin Tasks, Modify Admin Task.
3. Select the Forgotten Password task.
4. On the Profile tab, click Business Logic Task Handlers.

   The Business Logic Task Handlers screen opens. The BLTHGenerateTemporaryPassword handler should appear in the list of handlers.
5. Click the right arrow icon to edit the properties for the handler.
6. In the Property field, click the minus icon to delete the ShowPwdOnScreen property.
7. In the Property field, type in ShowPwdOnScreen again.
8. In the Value field, enter:

   false
9. Click Add.

Determine How Users Retrieve a Forgotten User ID

Once Identity Manager successfully verifies a user's identity, it displays the user's ID on the screen.

For additional security, you can configure Identity Manager to email the user's ID.

To configure Identity Manager to email a user ID

1. Configure the Identity Manager environment to support email notifications.
2. Choose Roles and Tasks, Admin Tasks, Modify Admin Task.
3. Select the Forgotten User ID task.
4. On the Profile tab, click Business Logic Task Handlers.

   The Business Logic Task Handlers screen opens.
5. Click the Delete ( ) icon next to the BLTHDisplayUserID handler to delete it.

   Deleting the BLTHDisplayUserID handler prevents Identity Manager from displaying the user ID in the User Console. If you want Identity Manager to display the user ID in the User Console *and* email the user ID, do not delete the BLTHDisplayUserID handler.

Logout Pages

A logout page is a page to which a user is directed after performing an action in certain Identity Manager task screens, such as clicking a Logout link from the User Console.

For self-service tasks, such as self-registration or forgotten password tasks, users are redirected to a logout page when they click Cancel to exit the task, or when they click OK in a confirmation or error message.

You can configure a custom logout page for the following Identity Manager screens:

- User Console
- Self-registration tasks
- Forgotten Password tasks

Important! If Identity Manager integrates with SiteMinder, configure the SiteMinder Web Agent to terminate the user session after the user logs out of Identity Manager. If you do not configure the Web Agent, SiteMinder may reopen the user session.

Configuring Logout Pages

Follow these steps:

1. Create one or more custom logout pages.

   To ensure that an HTML logout page is loaded from the Web server and not from the browser's cache, set up the logout page so that it cannot be cached in the browser. For example, for HTML pages, you can add the following meta tags to the page:

   ```
   <META HTTP-EQUIV="Pragma" CONTENT="no-cache">
   <META HTTP-EQUIV="Expires" CONTENT="-1">
   ```

   Important! Meta tags may not always work with an Internet Explorer browser. If not, use a cache-control HTTP header.

2. In the Identity Manager environments screen, click the name of the appropriate environment.

   The Environment Properties screen appears.

3. Click Advanced Settings, and click Miscellaneous.

   The Miscellaneous Properties screen appears.

4. In the Property field, type one of the following properties:

   - MainConsoleLogoutUrl—Overrides the default logout URL in the main console.

     This URL is also displayed for self-registration and forgotten password tasks if you do not specify custom logout pages using the *tasktag*LogoutUrl property.

- *tasktag*LogoutUrl—Specifies a logout page for a public task.

  In this property, *tasktag* identifies the task for which you are configuring a custom logout page.

  For example, to configure a logout page for the default self-registration task, enter the following in the Property field:

  SelfRegistrationLogoutUrl

  You can define multiple *tasktag*LogoutUrl properties to configure different logout pages for different tasks. For example, when you have different self-registration pages for customers and suppliers, you can define a different logout page for each task.

Note: You specify the task tag when you configure a task in the User Console. For more information, see the *Administration Guide*.

5. In the Value field, type the URL that users are redirected to at logout.
6. Click Save.

Chapter 7: Branding the User Console

This section contains the following topics:

Skins

Identity Manager users perform most admin tasks in a user console. Identity Manager includes the following user consoles:

- ca12—The default user console
- ui7—The user console that conforms to the new user interface standard from CA Technologies.
- imcss—An alternate user console that you can customize for your business needs

The console name is added to the URL for each console page. For example, the following URLs are for the Create User task in the ca12 and imcss consoles, respectively:

- http://myserver.ca.com:8080/iam/im/neteauto/ca12/index.jsp?task.tag=CreateUser
- http://myserver.ca.com:8080/iam/im/neteauto/imcss/index.jsp?task.tag=CreateUser

Note: Identity Manager generates these URLs automatically, once an administrator logs in to the user console.

You can change the appearance of a console with skins. A *skin* is a set of components that you can use to customize the appearance of the user interface (UI) fort a given audience.

You can change the following aspects:

- Images
- Colors and fonts
- Page headers and footers

In Identity Manager environment, you can have more than one skin, and you can create different skins for different audiences. For example, employees may see one skin and partners may see another skin.

Components of a Skin

The components of a skin are as follows:

Cascading Style Sheet

Defines the CSS files that contain color and font definitions.

Location:

iam_im.ear/user_console_war/app/imcss/*skin_name*/*fileName.css* *

Images

Defines a graphic file that can be displayed on a web page, such as a .gif, .jpeg, or .png file.

Location:

iam_im.ear/user_console_war/app/imcss/*skin_name*/image/*

.properties file

Defines the text file that defines the components for the skin.

Location:

iam_im.ear/user_console_war/app/imcss/*skin_name*/

Iam_im.ear

Defines one of the following installed locations of the Iam_im.ear directory:

- WebLogic:

  WebLogic_home\user_projects\domains*Identity-Manager-Domain-Name*\applications\iam_im.ear\

- JBoss:

  JBoss_home\jboss-*version*\server\default\deploy\iam_im.ear\

- WebSphere:

  was_im_tools_dir\WebSphere-ear\iam_im.ear\

was_im_tools_dir

Defines the installed location of the Identity Manager Tools for WebSphere.

Note: After making changes in the WebSphere-ear\iam_im.ear directory, you must package the directory into an Enterprise Archive (EAR) file and install the EAR file in the WebSphere Application Server. For more information, see the *Installation Guide*.

Identity Manager Skins

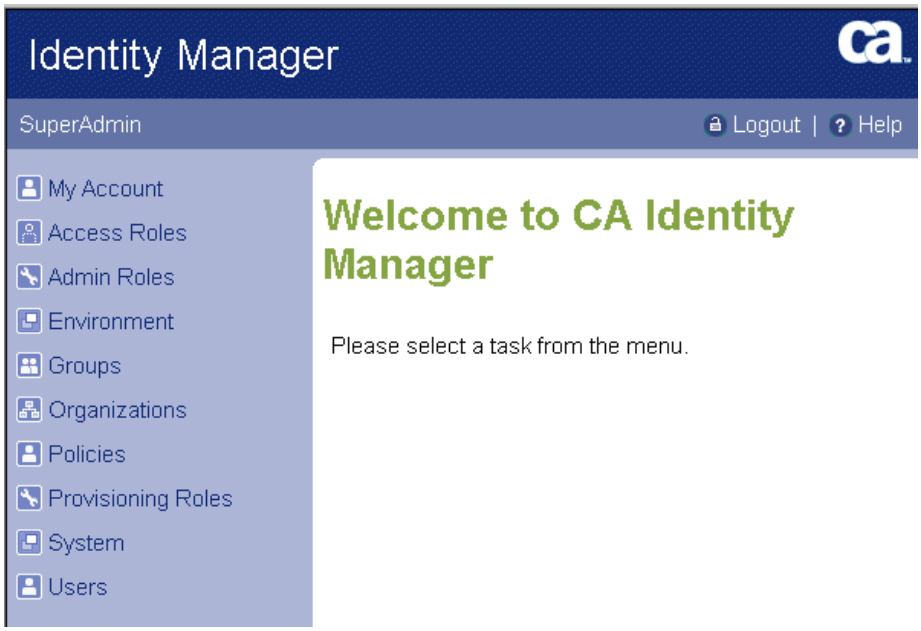
Identity Manager includes the following skins:

- idm—Defines the components of the default idm User Console. Skin components that are not defined by custom skins default to the *idm* components. For example, if a skin does not include a custom CSS file, Identity Manager uses the default CSS file from the *idm* skin.

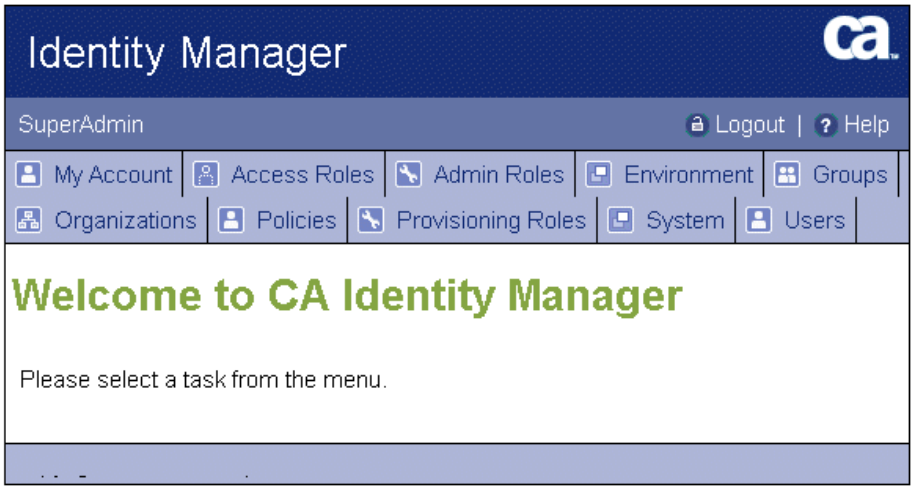
This skin is installed in the following folder:

iam_im.ear/user_console_war/app/imcss/idm

The following illustration shows the idm User Console:



- horizontal—Defines the components of the *horizontal* User Console, as shown in the following illustration:

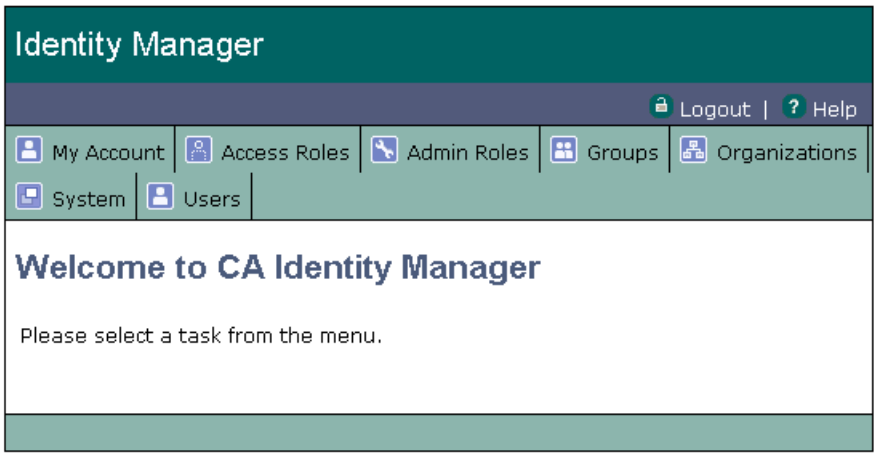


This interface looks similar to the *idm* skin but displays menu options in a horizontal format.

This skin is installed in the following folder:

iam_im.ear/user_console_war/app/imcss/horizontal

- horizontal2—Defines the components of the *horizontal2* User Console, as shown in the following illustration:

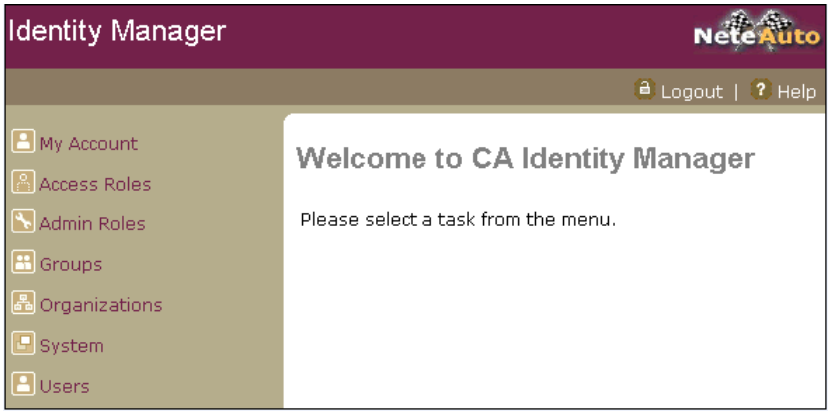


This interface looks similar to the *horizontal* skin but displays green and purple colors in the same horizontal format.

This skin is installed in the following folder:

iam_im.ear/user_console_war/app/imcss/horizontal2

- neteauto—Defines the components of the *neteauto* User Console, as shown in the following illustration:



This interface has the same layout as the *idm* skin but displays red and light brown colors and the NeteAuto image.

This skin is installed in the following folder:

iam_im.ear/user_console_war/app/imcss/neteauto

This chapter uses the *neteauto* skin for demonstration purposes.

Note: The skins in the various illustrations apply to the imcss console.

Skin Associations

A skin can inherit properties from a *parent skin*. Therefore, you can define a component once and reuse it in multiple skins. For example, a company logo, defined in a parent skin, can propagate to all the company skins.

When Identity Manager renders a user console, it checks for the appropriate skin for the user. When a parent skin is specified, its components are added.

For example, suppose the *corporate* skin includes only the company logo. A *sales* skin includes a custom footer JSP and points to the parent skin, *corporate*. When salespersons access Identity Manager, they see the footer defined in the *sales* skin, and the logo, colors, and fonts defined in the *corporate* skin.

How to Create a Skin

Creating a skin involves the following steps:

1. Copying an existing skin
2. Updating the skin's images
3. Updating the skin's colors and fonts
4. Modifying the .properties file
5. Modifying the index.jsp file

Copy an Existing Skin

Perform the following procedure to copy an existing skin.

Follow these steps:

1. Look at the illustrations in Identity Manager Skins, and decide which of the following skins is similar to the one that you want to create for your company:
   - *idm* skin
   - *horizontal* skin
   - *horizontal2* skin
   - *neteauto* skin

   For illustrative purposes, this section assumes that you chose the look of the *neteauto* skin and will call it *yourskin*. The name *yourskin* is the name of your company's skin.
2. Copy the following file:

   iam_im.ear/user_console_war/app/imcss/neteauto

   iam_im.ear/

   Give it the following name:

   iam_im.ear/user_console_war/app/imcss/yourskin
3. In the yourskin directory, rename the files in the following table:

Current Name	New Name
neteauto.css	yourskin.css
neteauto_task.css	yourskin_task.css
neteauto.properties	yourskin.properties

   To simplify administration, the name of the .properties file and the .css files should correspond to the name of the skin.

   Note: Skin names are case-sensitive.

Update the Skin Images

Update any of the images (.gif, .jpg, or .png) in the following directories with the images that match the look of your company's skin:

- *iam_im.ear*/user_console_war/app/imcss/yourskin/image
- *iam_im.ear*/user_console_war/app/imcss/yourskin/image/category

- *iam_im.ear*/user_console_war/app/imcss/yourskin/image/tab
- *iam_im.ear*/user_console_war/app/imcss/yourskin/image/tasks

For example, to put your company's logo in the *yourskin* skin, replace logo.jpg from the image directory with your company's logo.

We recommend using the same image file names because many of these images are already mapped with the appropriate paths in the yourskin.properties file.

Update the Skin Colors and Fonts

Update the *yourskin* skin's cascading style sheets with the fonts and menu, background, margin, and other colors of your choice.

The paths to the style sheets are as follows:

- *iam_im.ear*/user_console_war/app/imcss/yourskin/yourskin.css
- *iam_im.ear*/user_console_war/app/imcss/yourskin/yourskin_task.css

Modify the .properties File

Perform the following procedure to modify the .properties file.

Follow these steps:

1. In the yourskin.properties file, replace every *neteauto* reference with *yourskin*.
2. Make the following additional modifications in the yourskin.properties file:
   a. If it is not already specified, set the following in the parent section:

      ```
      parent=/app/imcss/idm/im.properties
      ```

      The parent section specifies the location to the properties file of the parent skin. In this example, the parent skin is *idm*.

   b. Set the cascading style sheets as follows:

      ```
      stylesheet/skin.css=yourskin/yourskin.css
      stylesheet/skin_task.css=yourskin/yourskin_task.css
      ```

c. Proceed as follows:

If you replaced the images in the skin without renaming them, go to Step d. If you renamed images, do the following:

- In the header images section, set the path to header images from Update the Skin Images (see page 125).

  You can add multiple image entries to the .properties file by adding an entry like the following:

  image/logo.gif=yourskin/image/logo.jpg

- In the standard tabs section, set the path to tab images from Update the Skin Images (see page 125).
- In the task body buttons section, set the path to task body button images from Update the Skin Images (see page 125).

d. Save the file.

Modify the index.jsp File

Perform the following procedure to modify the index.jsp file.

Follow these steps:

1. Open the index.jsp file in a text editor. This file is located in the following folder:

   <iam_im.ear>/user_console_war/app/imcss

2. Under the <skin:update> section, add the following as the first entry:

   ```
   <skin:skin name="yourskin" filename="/app/imcss/yourskin/yourskin.properties"
   />
   ```

3. Save the file.

Identity Manager selects the default skin for the Identity Manager User Console based on which skin tag is listed first in the <skin:update> section of the index.jsp file.

As noted in the following <skin:update> section from the index.jsp file, Identity Manager will display the user console for the *yourskin* skin:

```
<skin:update>
    <skin:skin name="yourskin" filename="/app/imcss/yourskin/yourskin.properties"
    />
    <skin:skin name="idm" filename="/app/imcss/idm/im.properties" />
    <skin:skin name="neteauto" filename="/app/imcss/neteauto/neteauto.properties"
    />
    <skin:skin name="horizontal"
    filename="/app/imcss/horizontal/horizontal.properties" />
    <skin:skin name="horizontal2"
    filename="/app/imcss/horizontal2/horizontal2.properties" />
</skin:update>
```

When you want to use a different skin, you need to change the order of the skin tags listed.

Recompile the .jsp Files and Restart the Application Server

If you are running Identity Manager on a JBoss application server, you must recompile the JSP files and restart the application server after you have created a skin, as follows:

1. Configure your application server to recompile all the console.jsp files (see page 185).
2. Restart the application server so that the *yourskin* skin's changes can take effect.

Note: If you are using a different application server, you do not need to complete this step.

How to Use Multiple Skins

You may want different sets of users to see different skins. For example, the skin for employees may include the company logo and colors. The skin for suppliers could include a different logo and color scheme, and display a company-specific footer.

Note: This functionality requires integration with SiteMinder, an optional component.

If Identity Manager integrates with SiteMinder, you can use a SiteMinder response to associate the *yourskin* skin with a set of users. The response is paired with a rule in a policy, which is associated with a set of users. When the rule fires, it triggers the response to pass information about the skin to Identity Manager, to build the user console.

The following steps are involved:

1. A user tries to access a Identity Manager environment. (A Identity Manager environment is protected by SiteMinder.)
2. SiteMinder authenticates and authorizes the user. The rule in the policy fires, triggering a response that contains information about the skin.
3. The response is passed as an HTTP header variable to Identity Manager through the Web agent.
4. Identity Manager builds the User Console based on the skin.

Prioritize Skins

Multiple policies (and therefore multiple skins) may apply to a single user. To make sure that the user sees the correct skin, assign priority numbers to skins.

When you create a skin response, add the priority number to the header variable name. The priority number can be from 1 to 999, with 1 having the highest priority.

For example, the header variable name in the response for the *sales* skin is skin3. For the *management* skin, the header variable name is skin2. When a user logs in to Identity Manager as a sales manager, he or she will see the management skin because it has a higher priority.

Create a Skin Response

Perform the following procedure to create a skin response.

Follow these steps:

1. Log in to one of the following interfaces as an administrator with Domain privileges:
   - For CA SiteMinder Web Access Manager r12 or higher, log in to the Administrative UI.
   - For CA eTrust SiteMinder 6.0 SP5, log in to the Policy Server User Interface.

   Note: For information on using these interfaces, see the documentation for the version of SiteMinder that you are using.
2. Locate the policy domain that protects the Identity Manager Environment. The domain name has the following format:

   *environment_name*Domain

   environment_name

   Defines the name of the Identity Manager Environment.

For example, when the Identity Manager environment name is *employees*, the domain name is *employees*Domain.

Note: The policy domain must be associated with the user directory that contains the Identity Manager users who will use the skin.

3. View the realms for the domain.

   You see the following realms depending on the features you configured for the environment:

   - *environment_name*_ims_realm—Protects the Identity Manager Environment.
   - *environment_name*_pub_realm—Enables support for public tasks, such as self-registration and forgotten password tasks.

     This realm appears only when you have configured a public alias.

4. Create a rule in each of the realms. Specify the following:

   - Resource: /*
   - Actions: GET, POST

   To simplify administration, include the skin in the rule name. For example, when the skin name is *yourskin*, create a rule named *yourskin* skin.

5. Create a response for the domain with the following response attributes:

   - Attribute: WebAgent-HTTP-Header-Variable

     This adds a new HTTP header to the response.

   - Attribute Kind: Static
   - Variable Name: Enter one of the following variable names:
     - skin—Use this variable when all Identity Manager users see the same skin.
     - skin*x*—Use this variable if you have multiple skins. The **x** represents a priority value where 1 is the highest priority and 999 is the lowest priority.
   - Variable Value: Specify the name of the skin.

6. Create a policy in the policy domain with the following values:

   - Users: Specify users who should see the skin. For example, if members of the sales organization should see the skin, be sure to include ou=sales in the policy.
   - Rules: Add the rules you created in Step 4. Associate each rule with the response you created in Step 5.

How to Edit Console Pages

To change the layout of the User Console or to add links to a page's headers or footers, you can edit the following console pages:

- index.jsp (see page 132)—Hides or displays headers and footers in the main console. Also defines the available Identity Manager skins.

  The index.jsp file is located in *<iam_im.ear>* /user_console_war/app/imcss/

- head.jsp (see page 131)—Contains the code for the header in the User Console main page.
- home.jsp—Contains the confirmation message for a particular task.

  The head.jsp and home.jsp files are located in *<iam_im.ear>* /user_console_war/app/imcss/component

Note: If Identity Manager runs on a JBoss application server, you must manually recompile the JSPs (see page 185) that you modify before changes take effect.

How to Modify the Header in the Identity Manager User Console

You use the head.jsp file (located in *<iam_im.ear>*/user_console_war\app\imcss\component) to modify the header in the User Console. The header can contain images, links, title, colors, and fonts. The following illustration shows the <div id> markers in the head.jsp file, which generate the header for the *idm* skin:

To update the header, modify the code for the following entries in the head.jsp file:

-
-
-
-
-

For example, to change the link users are directed to when they click on the logo image that appears in the header for the User Console, replace the URL in the following code with the URL for your company's home page.

```
<div id="logo">
<a href="http://www.yourcompany.com/"><skin:img altKey="console.logo"
titleKey="console.logo" src="logo.gif" width="93" height="33" border="0" /></a>
</div>
```

Note: After modifying the head.jsp file, be sure to recompile the .jsp files (see page 128).

How to Modify and Show the Footer

To modify and show the footer, you can do the following:

- Insert the footer's HTML code into the index.jsp file (located in *iam_im.ear*/user_console_war/app/imcss).
- Modify the cascading style sheet file from one of the skins—*idm*, *horizontal*, *horizontal2*, and *neteauto*—so that Identity Manager displays the footer.

Insert the Footer's HTML into the index.jsp File

Perform the following procedure to insert the footer's HTML into the index.jsp file.

Follow these steps:

1. Open the index.jsp file in a text editor.
2. Under the section at the end of the file, add the HTML for *yourskin* skin's footer between the and markers.
3. Save the file.

Modify the Cascading Style Sheet to Display a Footer

Use the cascading style sheet file from each skin—*idm*, *horizontal*, *horizontal2*, and *neteauto*—to configure Identity Manager to display the footer or not. These cascading style sheet files are stored in the following folder:

iam_im.ear/user_console_war/app/imcss/

By default, the footer shows in the *horizontal* and *horizontal2* skins but not in *idm* or *neteauto*.

Presently, *yourskin* skin does not display the footer because it is a copy of *neteauto*, which uses the idm.css file to display footers. The neteauto.css and yourskin.css files contain color overrides only and import the idm.css file to handle other display functions like showing footers. In the idm skin that is shown as follows, the #foot section in the idm.css file shows the footer as being disabled:

```
#foot {
    display: none;
```

In the horizontal skin that is shown as follows, the #foot section in the horizontal.css file shows the footer as enabled:

```
#foot {
             border: 1px solid black;
             background-color: #AEB6D8;
             padding: 4px;
         display: block; /* to override parent */
```

The display: block reference means that Identity Manager displays the footer.

To get *yourskin* skin to display the footer:

- In yourskin.css, add the following code:

```
/*CONSOLE FOOTER*/
#foot {
    display: block;
}
```

In this section, you can also modify the border at the footer, background color, and padding.

Chapter 8: Custom Help

Identity Manager allows you to create your own custom help for tasks and tabs that you have customized in the User Console. To implement custom help, you can create a context-sensitive help system with custom HTML help files or Wiki pages and redirect help links within the User Console to access your custom help.

This feature also allows you to translate any of the default help (written in English) into another language.

This section contains the following topics:

How Customized Help Works

Identity Manager uses resource bundles to override default help files and to provide the ability to link to custom context-sensitive help.

Using the defined format, you can create a resource bundle and place it in the iam_im.ear. When a user clicks on a help link, Identity Manager will search through any applicable resource bundles for matches to custom help. If there is no match to custom help, Identity Manager will provide the default help to the user.

If an international user clicks on a help link and a language-specific resource bundle has been created, Identity Manager will check the browser locale settings of the user and open the language-specific help link. If there is no match within the language-specific resource bundle, the user will be directed to the default English help.

Note: Content on the custom pages will not show up when you search the default help or view the help index.

Custom Help Format

The help link resource bundle uses a key/value pair to determine which help page to direct the user to. For custom help, tags are used for Key IDs and the custom web page URL is the value. The syntax for the help link resource bundle may be one of the following:

TaskTag.PageTag=Help URL
TaskTag.@PageDefinitionTag=Help URL

A tag is the unique identifier for a task, screen, or tab. The parameters for the key/value pair are defined as follows:

Task Tag

The active task tag.

Page Tag

The active search or tab tag.

Page Definition Tag

The active search definition or tab definition tag.

Help URL

The help URL is either an absolute URL (http://www.neteauto.com) or a relative URL that points to content on the Identity Manager server (/iam/im/help/customhelp.html)

Example:

```
ModifyUser.Profile=/iam/im/userprofile.html
```

Custom Help Expressions

The following expressions have specific meaning when used in custom help resource bundles:

Expression	Meaning
*	Used to match any task tag, page tag, or page definition tag.
${task}	Used in the help URL. Replaced with the task tag of the active page.
${page}	Used in the help URL. Replaced with the page tag of the active page.
${pagedef}	Used in the help URL. Replaced with the page definition tag of the active page.

Example:

```
*.*=http://www.help.com/Wiki.jsp?page=${task}_${page}
```

The $task and $page expressions are replaced with the task tag and page tag for the current web page where the user clicked the help link. For example, if a user clicks the help link from the Profile tab of the Create Group task, the help URL would open the following help page: http://www.help.com/Wiki.jsp?page=CreateGroup_Profile

How Help Determines Which Link To Use

When a user clicks on a help link, Identity Manager performs checks to determine which help URL to use. These checks are made for the most specific case and then become more general. The bundle with the best match to the locale of the user is used.

The order of checks for a help link is as follows:

1. [TaskTag].[PageTag]
2. [TaskTag].@[PageDefinitionTag]
3. *.[PageTag]
4. *.@[PageDefinitionTag]
5. [TaskTag].*
6. *.*

Example:

```
*.*=http://www.help.com/Wiki.jsp?page=${task}_${page}
```

Matches anything without a more specific match and includes active task and page tags in the generated help link.

Example:

```
ModifyUser.Profile=/iam/im/userprofile.html
```

Matches the Profile tab on the Modify User task and directs the user to the relative URL /iam/im/userprofile.html

Example:

```
*.@Profile=/iam/im/profile.html
```

Matches all tabs that are derived from the Profile tab definition without a more specific match and directs the users to the relative URL /iam/im/profile.html.

How to Customize the Help

Custom help allows you to provide specific help pages for your users when accessing highly-customized or localized Identity Manager environments.

Follow these steps:

1. Create custom help pages and host them on a web site.
2. Create a resource bundle with key IDs mapped to the custom help pages. Name the resource bundle as follows:
   `help_EnvironmentName_languageidentifier.properties`

   Note: The language identifier is an optional, two-character abbreviation for a specific language.
3. Place the .properties file in the iam_im.ear/config/com/netegrity/config directory.

   Note: If you are localizing the help to more than one language, create a resource bundle for each language.
4. Restart the Identity Manager server.

   Your custom help will now override the default help when your users click on the help links in the Identity Manager User Console.

For internationalized help, users with their browser locale preferences set to another language will be directed to the appropriate custom help.

Examples of How to Use Custom Help

This section provides some examples of how custom help can address your business needs and localization requirements.

Example: Customize the Help

A customer has deployed a Identity Manager environment with heavily customized screens for Users, Groups, and Organizations. Unfortunately, the help content is static and pertains only to the default environment. They want to write their own help content to reflect the customization in their environment.

Follow these steps:

1. Write custom help pages for each custom task and tab.
2. Host the pages on a web site.

3. Create a resource bundle and place the .properties file in the iam_im.ear/config/com/netegrity/config directory.
4. Restart the Identity Manager server.

For example, if the environment name is neteauto, create the following resource bundle named help_neteauto.properties:

*.UserProfile=http://www.neteauto.com/imhelp/user.html

*.GroupProfile=http://www.neteauto.com/imhelp/group.html

*.OrgProfile=http://www.neteauto.com/imhelp/org.html

The help links for the neteauto environment access the default help except for the Profile tabs in the Modify User, Modify Organization, and Modify Group tasks.

Example: Create Wiki Help

A customer has a highly customized Identity Manager environment and would like to provide their users with access to a rich context-sensitive help system. They want to create a Wiki for their environment that is available to users through the help links.

Follow these steps:

1. Write the Wiki content for your customized environment.
2. Create a resource bundle and place the .properties file in the iam_im.ear/config/com/netegrity/config directory. The resource bundle must have a single entry:

   ```
   *.*=http://www.neteauto.com/wiki.jsp?page=${task}-${page}
   ```

   The help links launches the Wiki with a page specific to the task and tab.
3. (Optional) If the page has not been created yet, the user can enter the details for the task and tab.
4. Restart the Identity Manager Server.

Example: Localize the Help

A customer in Japan has purchased a localized version of Identity Manager. Unfortunately, all of the help is in English. They want to write their own help in Japanese and implement it for their User Console.

Follow these steps:

1. Write custom help pages for each task and tab.
2. Host the custom help pages on a web site.

3. Create a resource bundle and place the .properties file in the iam_im.ear/config/com/netegrity/config directory.

   For example, if the environment is named neteauto, create the following resource bundle named help_neteauto.properties:

   ```
   ModifyUser.*=http://www.neteauto.jp/modifyuser.html
   ModifyGroup.*=http://www.neteauto.jp/modifygroup.html
   ```

4. Restart the Identity Manager Server.

   The help links in the User Console will redirect to the language-specific, custom help pages that are hosted on the www.neteauto.jp web site.

Example: Internationalize the Help

An international company has purchased Identity Manager and must support users in English, Spanish, and French. They are able to internationalize their Identity Manager environments, but the help is written in English. They want to write a version of help in each language and make the correct help available depending on what language the user requires.

Follow these steps:

1. Write custom help pages for each task and tab, and in each language.
2. Host the custom help pages on a web site.
3. Create two locale-specific resource bundles and place the .properties files in the iam_im.ear/config/com/netegrity/config directory.

   For example, if the environment is named neteauto, create two resource bundles such as help_neteauto_es.properties (for Spanish) and help_neteauto_fr.properties (for French).

4. Restart the Identity Manager Server.

   Users with their browser locale preferences set to Spanish or French are directed to the appropriate custom help in their language. All other users are directed to the default English user help.

Chapter 9: Identity Manager Localization

This section contains the following topics:

Introduction

You can localize the following elements in the User Console to a particular language:

- Prompts, buttons, text, tabs, and titles that appear in the User Console
- Validation, status, and error messages and instructions that are displayed to Identity Manager users
- Email notifications
- Graphics
- Custom features created with the Identity Manager APIs

Note: Identity Manager reports are not localized.

Resource Bundles

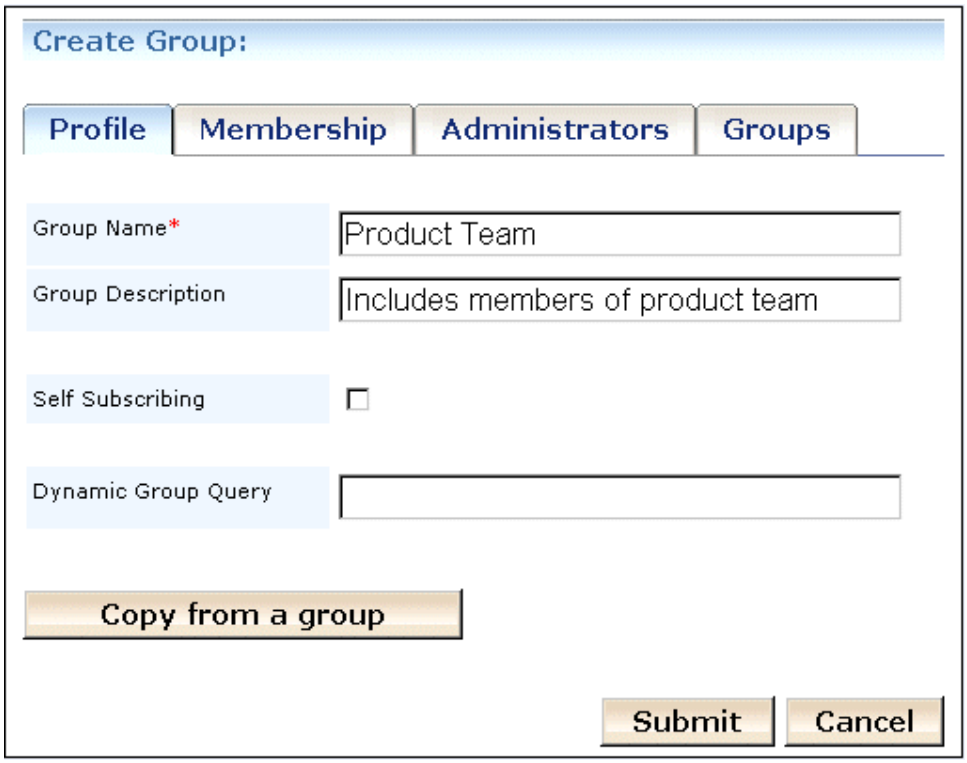
A *resource bundle* is Identity Manager file that associates user interface elements with locale-specific text strings. In this file, a user interface element is assigned a key ID, which is paired to a locale-specific text string.

When Identity Manager encounters an element with a key ID on a user interface screen, such as an admin task profile screen, it displays the paired value from the resource bundle.

For example, the entry for a Submit button in a Identity Manager resource bundle is as follows:

```
global.button.submit=Submit
```

The key ID is global.button.submit and the value is Submit. When Identity Manager encounters the global.button.submit key ID in a user interface screen definition, it displays the word Submit, as shown in the following screen:



Note the following about key IDs:

- Elements in the User Console can have only one key ID, and the key ID must be unique for a particular resource bundle.

- Key IDs are case sensitive, must contain only US-ASCII characters, and cannot contain spaces. However, key IDs may contain multiple strings separated by periods (.). For example, the key ID for the Department field name may resemble the following:

  attribute.displayname.department

- Identity Manager supports two formats for specifying key IDs in admin task screens:

  - Using a syntax that includes the resource bundle name

    This method allows you to specify the name of a custom resource bundle. You can use a custom resource bundle to separate custom mappings from the default resource bundle.

    The format for this syntax is as follows:

    ${bundle=*ResourceBundle*:key=*keyID*}

    In this syntax example:

    bundle=*ResourceBundle*

    Identifies the resource bundle that includes the text string mapping for the key ID, if you are using a custom resource bundle.

    Note: Create the custom resource bundles in iam_im.ear/custom/resourceBundles and then refer to it as resourceBundles.MyResourceBundle.

    Bundle names use standard Java localization syntax. The bundle name does not need to include the .properties extension, but must be fully qualified, relative to iam_im.ear/custom.

    This name/value pair is optional. You do not need to specify a resource bundle if the key ID is defined in the base resource bundle, IMSResources.properties. In this case, the key ID in the admin task screen would be:

    ${key=*keyID*}

    key=*keyID*

    Identifies the key ID that maps to the text string to display. The mapping must exist in a resource bundle.

    A completed key ID entry resembles the following:

    ${bundle=resourceBundles.MyResourceBundle:key=attribute.displayname.department}

  - Using the prefix imstask.label as follows:

    imstask.label.*elementname*

    For example:

    imstask.label.attribute.displayname.department

When you use this method, attribute.displayname.department must appear in the resource bundle.

Note: If you are creating custom localization files (see page 151), you use a translation tool that changes the value mapped to a key ID for a different language. For example, to display the Submit button in French, the translation tool changes the value paired with the global.button.submit key ID in the French resource bundle as follows:

global.button.submit=Enter

Sample Localized Identity Manager Environments

Identity Manager includes the files to create translated versions of the User Console. You can use these samples as defined, or use the samples as templates for creating a custom translated User Console. The following languages are supported:

- French
- German
- Italian
- Japanese
- Korean
- Polish
- Simplified Chinese
- Spanish

The files that support these language versions are installed at the following location:

admin_tools\samples\Localization*language*

admin_tools

Specifies the installed location of the Identity Manager Administrative Tools.

The Administrative Tools are placed in the following default locations:

Windows: C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools

UNIX: /opt/CA/IdentityManager/IAM_Suite/Identity_Manager/tools

language

Specifies the language that you want to use.

Restrictions on the Use of International Character Sets

The following input must contain ASCII characters only:

- Environment names and aliases
- Directory names
- Task tags
- Class names used in the following APIs:
  - Event Listener API
  - Notification Rule API
  - Logical Attribute API
  - Workflow Organization Resolver API
- Logical attribute names and physical attribute names used by the Logical Attribute
- The URL for the end-user license agreement that appears when users self-register

  Note: The end-user license agreement can contain internationalized character sets.

How to Configure Support for Localization

To configure the system to use a translated version of Identity Manager, you perform the following steps:

1. Verify that the system hosting Identity Manager meets the prerequisites defined in Localization Prerequisites (see page 146).
2. If Identity Manager integrates with SiteMinder, configure the SiteMinder Web Agent for localization.
3. If Identity Manager runs on a JBoss application server, modify the Tomcat server.xml file (see page 148).
4. Configure the PATH Variable.
5. Complete *one* of the following steps:
   - If you are using one of the sample localized Environments, go to How to Use the Sample Localized Environments (see page 149).

     Identity Manager installs translated versions of the User Console and online help in the following languages:
     - French
     - German
     - Italian

- Korean
- Japanese
- Polish
- Simplified Chinese
- Spanish

You can use these files as installed, or use these files as a template to create a custom Environment in one of these languages.

- If you want to translate Identity Manager into a language for which a sample is not available, or you want to translate a highly customized Environment, go to How to Create Custom Localization Files (see page 151).

6. Translate the following additional components, as needed:
   - Email templates (see page 152)
   - Branding skins (see page 152)
   - Custom code (see page 152)

Localization Prerequisites

Before configuring Identity Manager for localization, note the following:

- Identity Manager must be installed and configured, and at least one Environment must exist.

  Note: If you are using SiteMinder and your Environment manages users with multi-byte user IDs, those users must authenticate with a SiteMinder authentication scheme that supports multi-byte characters. For example, HTML Forms Based authentication. The basic authentication scheme does not support multi-byte authentication. For information on configuring an authentication scheme for Identity Manager, see the chapter on configuring SiteMinder Features for Identity Manager in the *Configuration Guide*.

- If you are using SiteMinder, ensure that a supported version of the SiteMinder Policy Server is installed and configured.
- Ensure the user directory is configured to support localization. For more information, see the documentation for the user store you are using.

- If you are using an LDAP directory as a user store, verify that the LDAP directory is not enforcing a 7-bit check for the user ID, password, and email attribute. For more information, see the documentation for the LDAP directory you are using.
- If you are using an Oracle database object store, verify that the character set component in the NLS_LANG parameter, and the database character encoding are set to AL32UTF8, a multi-byte character set.

  Identity Manager uses multi-byte character encoding (UTF-8). If you specify a single-byte character set, characters may appear corrupted because of insufficient character space.

  Note: For instructions on setting the NLS_LANG parameter and the database character encoding, see the Oracle documentation.
- Identity Manager includes tools that you use to localize a new Identity Manager environment.To use these tools, you need the following software:

  Java SDK native2ascii tool

  Translates to and from Unicode escape sequences. Because localization mainly involves the translation of property files, the tools process all files in ASCII with Unicode escape sequences.

  To configure the localization tools, install the native2ascii tool by installing a JDK on the system where the Identity Manager Administrative Tools are installed.

  For information about the native2ascii tool, go to the following location at the Java Web site:

  http://java.sun.com/j2se/1.5.0/docs/guide/intl/index.html

  ant 1.7 or greater

Configure the SiteMinder Web Agent

Configure the encoding for HTTP header values that the SiteMinder Web Agent passes to Identity Manager by setting the HTTPheaderEncodingSpec parameter as follows:

```
HTTPHeaderEncodingSpec=encoding_spec, wrapping_spec
```

where *encoding_spec* is a text string that represents one of the following encoding types: UTF-8 or Shift-JIS, and *wrapping_spec* is the wrapping specification, which must be RFC-2047.

For example:

```
HTTPHeaderEncodingSpec="Shift-JIS,RFC-2047"
```

Note: If no value is specified in the HTTPHeaderEncodingSpec parameter, the encoding is UTF-8 with no wrapping.

You can configure the HTTPHeaderEncodingSpec parameter centrally in the Agent Configuration Object or locally for each Web Agent, in the WebAgent.conf file.

Note: For more information, see the *CA SiteMinder Web Access Manager Web Agent Configuration Guide.*

Change the JBoss server.xml

If Identity Manager is hosted on a JBoss application server, localizing Identity Manager to a multibyte character set requires a configuration change. The JBoss server.xml file must specify UTF-8 encoding for URI. Change the encoding in the Connector element for the HTTP/1.1 Connector in the following file:

jboss_home/server/default/deploy/jboss-web.deployer/server.xml

Specify UTF-8 encoding for URI as follows:

```
<Server ...>
  <Service ...>
    <Connector port="8080" ... URIEncoding="UTF-8"/>
      ...
    </Connector>
  </Service>
</Server>
```

Configure the PATH Variable

Identity Manager includes translated versions (see page 144) of the default User Console and online help that you can use in localized environments. However, if you have a custom environment that you want to translate, Identity Manager provides localization tools that you can use to perform the translation. To use these tools, you need to have ant 1.7 or later and Java JDK 1.5 or later installed. Both ant and Java JDK must be on your path. You can add them with the following commands:

Windows:

```
PATH %ANT_HOME%\bin;%PATH%
PATH %JAVA_HOME%\bin;%PATH%
```

UNIX:

```
PATH=%ANT_HOME%\bin:$PATH
export PATH
```

How to Use the Sample Localized Environments

Identity Manager deploys the files that support the sample localized Environments during installation.

To view localized screens and tasks in the User Console, complete the following steps:

1. Import the RoleDefinitions.XML file (see page 149).
2. Install the Online Help. (see page 150)

Import a Sample RoleDefinitions.xml File

The RoleDefinitions.xml file creates the tasks that appear initially in the User Console.

You must import the required sample RoleDefinitions.xml file to the required Identity Manager environment.

To import a RoleDefinitions.xml file

1. In the Management Console, click Identity Manager environments.

   A list of Identity Manager environments appears.

2. Click the name of a Identity Manager environment.

   The Properties screen for that environment appears.

3. Click Role and Task Settings.

4. Click Import.

   The Management Console lists the RoleDefinition.xml files that are available to add the Environment.

5. Search for and select the appropriate RoleDefinitions.xml file for the language that you need.

   Translated RoleDefinitions.xml files are located in the Administrative Tools:

   admin_tools\samples\Localization*language*

   admin_tools

   Specifies the installed location of the Identity Manager Administrative Tools.

   The Administrative Tools are placed in the following default locations:

   Windows: C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools

   UNIX: /opt/CA/IdentityManager/IAM_Suite/Identity_Manager/tools

   language

   Specifies the language that you want to use.

   Note: For a description of the available RoleDefinitions file, see readme.txt file in *admin_tools*\samples\Localization.

6. Click Finish.

   The status is displayed in the Role Configuration Output window.

7. Click Continue to exit.

8. Repeat Steps 1 through 6 for each RoleDefinitions.xml file for each Environment.

Install the Online Help

Identity Manager includes translated online help for each translated version of the User Console.

Follow these steps:

1. Download the translated online help for the CA Support site as follows:

   a. Open the CA Support site in a browser window.

   b. Click Documentation.

   c. Enter Identity Manager in the Select a Product field in the Find Other Product Documentation section.

   d. Select the applicable release and language, then click Go.

      The Support site displays the translated documents and online help for each service pack in the release that you selected.

   e. Click the online help ZIP file to download it.

2. Copy the online help ZIP file as follows:

   copy *filename*.zip *iam_im.ear*\user_console.war\

   filename

   The name of the ZIP file that you downloaded in step 1.

   iam_im.ear

   The deployed location of the Identity Manager application on the application server.

   Note: Consider creating a backup copy of the default online help before replacing it with a translated version. The default online help is overwritten by the translated version.

3. Unzip the im_help.zip in the user_console.war directory.

   The translated version of the online help is available for use.

How to Create Custom Localization Files

Localizing resource bundles involves the following operations:

1. Export your roles into a RoleDefinitions.xml file, and do the following:
   - If you localize Identity Manager for a single language, go to the next step.
   - If you localize Identity Manager for multiple languages, tokenize the RoleDefinitions.xml file.

     When you run the command to create a tokenized RoleDefinitions.xml file, Identity Manager creates *RoleDefinitions*_Tokenized.xml and *RoleDefinitions*_Tokenized.properties.
2. Translate the RoleDefinitions.xml file.
3. Translate the resource bundles.
4. Install the localized files.
5. Import a tokenized RoleDefinitions.xml file into a new or existing Environment.

Note: For complete instructions, see the readme file located in *admin_tools*\samples\Localization. The Administrative Tools are placed in the following default locations:

- **Windows:** C:\Program Files\CA\Identity Manager\IAM Suite\Identity Manager\tools
- **UNIX:** /opt/CA/IdentityManager/IAM_Suite/Identity_Manager/tools

Translate Email Templates

Identity Manager can send email notifications to interested users when certain events occur, such as when an administrator approves a request or when a task completes.

You create and manage email notifications using email notification policies in the User Console.

Note: If the email contains internationalized characters, users must set the encoding for their email client to Unicode (UTF-8) to view the message correctly. For information about changing email encodings, see the documentation included with the email client.

To translate email templates

1. Configure support for email notifications as described in the *Administration Guide*.
2. Complete the procedures for using email notification in the *Administration Guide*.

   When you customize the email notifications, translate the content to the appropriate language.

How to Localize Branding Skins

Branding skins determine the graphics, colors, and fonts for a Identity Manager environment. Identity Manager includes a default skin and several sample skins that you can customize for a particular language.

Modify the Change My Account Task

For localized environments, the default consultation process has English role names in the configuration which causes the Change My Account task to fail. To fix this fail, do the following tasks:

1. In the User Console, modify the Change my Account task.
2. Click the Events tab.
3. Select the language-specific User Manager and User Approver roles from the role list.
4. Save the changes.

Localizing Custom Code

When you create custom code using the Identity Manager APIs, you can do the following tasks:

- Localize the exception messages that are displayed in Logical Attribute Handlers, Business Logic Task Handlers, and validation rules.
- Localize the display names for logical attributes.
- Specify the language that Identity Manager uses for all its communications with a third-party client when using the Remote Task Execution (TEWS) feature.
- Use the Localizer helper class to retrieve the localized version of a string for a given locale. For example, you can retrieve the localized version of a label on a task screen or a task name.

For more information about localizing custom code, see the following documents:

- The Localizer and IMSException area of the Support Objects information in *Programming Guide for Java*.
- The class descriptions for the Localizer and IMSException objects in the Javadoc.

How to Display Identity Manager in a Different Language

When an Environment supports multiple languages, users can select the language they want to view from a Choose Language field in the login screen and in the User Console. When a user chooses a different language, Identity Manager displays the login screen and User Console screens in the new language. The user does not need to log out of the Environment for changes to take effect.

The list of languages in the Choose Language field depends on the availability of localized files for the Environment. The language preference defined in the user's browser determines the default language.

Note: To display the Choose Language field, you complete the following configuration steps in Identity Manager:

- Configure the Enable Language Switching field in the Management Console
- Add support for the language in the Environment by importing translated RoleDefinitions files, and installing online help files

  Note: If you are creating custom localized files instead of using the sample files, additional configuration steps are required.

Enable Language Switching

You enable support for switching languages in the Identity Manager login screen and the User Console in the Management Console.

To enable language switching

1. Open the Management Console using the following URL:

   http://*hostname:port*/iam/immanage

   Note: If you are using a Web Agent to provide advanced authentication for Identity Manager, note the following:

   - You do not need to specify the port number.
   - The Choose Language field will not appear on the login screen.

2. Go to Environments, *YourEnvironment*, Advanced Settings, User Console.
3. In the General Behavior section, select Enable Language Switching.
4. Click Save.
5. Restart the Environment for the changes to take effect.

Set the imlanguage HTTP Header in a SiteMinder Response

When Identity Manager integrates with SiteMinder, you can define a user's locale preference using an imlanguage HTTP header. In SiteMinder, you set this header within a SiteMinder response and specify a user attribute as the header's value. This imlanguage header acts as the highest priority locale preference for a user.

To set the imlanguage HTTP header in a SiteMinder response

1. Log in to one of the following interfaces as an administrator with Domain privileges:

   - For CA SiteMinder Web Access Manager r12 or higher, log in to the Administrative UI.
   - For CA eTrust SiteMinder 6.0 SP5, log in to the Policy Server User Interface.

   Note: For information on using these interfaces, see the documentation for the version of SiteMinder that you are using.

2. Locate the SiteMinder response for the protected Identity Manager environment:

   *Identity Manager-environment*response_ims

3. Create a SiteMinder Response Attribute Editor with the following values:
   - Attribute: WebAgent-HTTP-Header-Variable

     This adds a new HTTP header to the response.
   - Attribute Kind: User Attribute
   - Variable Name: imlanguage
   - Attribute Name: Enter the name of the physical attribute from the user directory that contains the preferred locale, for example, employeetype.

     The contents of the user attribute in the user directory should contain a language preference consisting of a two-letter language abbreviation followed by an optional two-letter country code, separated by a hyphen or underscore. The language preference setting in the user attribute is not case-sensitive, as in the following examples:
     - ja
     - en-US
     - en_uk
4. Save the response.

   The imlanguage HTTP header is now set in a SiteMinder response.

Chapter 10: Validation Rules

This section contains the following topics:

Validation Rules Introduction

Values are assigned to data store attributes through task screen fields or programmatically. Attribute validation rules help ensure that the values users type in task screen fields or that are supplied programmatically meet certain requirements, as in the following examples:

- User directory requirements, such as enforcing a data type, or verifying that an entry such as a date is formatted in a particular way.
- Data integrity. Does an entry make sense in the context of other information about the task screen or according to site-specific business rules?

A validation rule can be directly associated with a task screen field, or be indirectly associated with the field by being associated with a managed object attribute that is configured for the field.

All validation rules directly or indirectly associated with a task screen's fields must be satisfied before Identity Manager can begin processing the task. When a supplied value is invalid, a message associated with the violated rule is displayed, and the user can then correct the entry and resubmit the task.

About Validation Rules

Validation rules enforce requirements, such as in the following examples:

- A Quantity field must contain only numeric characters.
- A Telephone Number field must be formatted as nnn-nnn-nnnn.
- An Employee ID field must contain a number no higher than 9999.

- The value typed in a ZIP Code field must be appropriate for the values typed in the City and State field.
- Does the value typed in a Title field qualify the user for the security clearance typed in Security Level?

In addition to verifying a user entry, a validation rule can *change* an entry so that the entry conforms to the rule's requirements without further user intervention, as in the following examples:

- A validation rule for a Telephone Number field requires that telephone numbers be formatted as nnn-nnn-nnnn. If a user types the value 9785551234, the validation rule automatically changes the entry to the correct format, 978-555-1234.
- A validation rule for a Department Number field requires that the number must be prefixed with a three-character code representing the name typed in the Region field. When the prefix is missing or incorrect, the validation rule supplies the correct prefix.

Changing an entry through a validation rule is named *transformation*.

Types of Validation Rules

The two types of validation rules are as follows:

- **Task-level validation**—validates an attribute value against other attributes in the task. For example, you can verify that the area code in a user-supplied telephone number is appropriate for the user's city and state.

  During task-screen configuration, task-level validation rules are directly associated with task screen fields.

  You can use this type of validation to enforce data integrity.

- **Directory-level validation**—validates the attribute value itself, and not in the context of other attributes in the task. For example, you can verify that a user-supplied telephone number matches the nnn-nnn-nnnn format used in the directory.

  In directory.xml, directory-level validation rules are mapped to a managed object attribute through a rule set. The rules in the rule set are applied to any task screen field configured with the attribute.

  You can use this type of validation to enforce user directory requirements.

Identity Manager executes task-level validation rules before directory-level validation rules.

Example: Comparing Directory-Level Validation and Task-Level Validation

In this example, a telephone attribute is mapped in directory.xml to a directory-level validation rule requiring telephone numbers to be formatted as nnn-nnn-nnnn. All fields configured with the telephone attribute are validated against the nnn-nnn-nnnn format whether the field appears in a Create User task screen, a Create Supplier task screen, or any other task screen.

If a Telephone Number field appears on a Create Customer task screen, like telephone number fields in other task screens, this field is configured with the telephone attribute that requires the nnn-nnn-nnnn telephone number format. However, because some of the company's customers are located in other states, the Telephone Number field on the Create Customer task screen is also associated with the following task-level validation logic:

- Check the value in the State field.
- When the customer is located out of state, be sure that the area code of the customer's telephone number is appropriate for the customer's state.

Validation Rule Sets

With directory-level validation, one or more validation rules are assigned to a rule set, and the rule set is associated with a managed object attribute.

Rule sets let you define and apply rules in a granular way, such as in the following examples:

- A rule can be used in different rule sets
- Rules can be executed in different combinations

When a rule in a rule set fails (for example, a Java or JavaScript rule returns False), any exception messages associated with the rule are presented to the user. All validation rules associated with the attribute must be satisfied before the attribute is considered validated.

Order of Execution

Rules are executed in the order in which they are listed in the rule set. Identity Manager executes each rule in a rule set separately, and transparently continues to each subsequent rule in the rule set unless a rule fails.

Because validation rules are executed in a predictable order, you can implement rules whose actions are dependent upon the outcome of previous rules, as in the following examples:

- One rule's output can become input to the next rule.
- When a field value is changed during validation, the new value can be evaluated in subsequent rules.

Basics of Validation Rule Definition

Perform the following basic operations when defining custom validation rules:

- **Implement a validation rule**. Implement a validation rule in any of the following ways:
  - Regular expression
  - JavaScript
  - Java class
- **Integrate a validation rule with Identity Manager through a task screen or directory.xml**. Do so either inline (directly in the task screen or directory.xml file) or by reference (referencing a JavaScript source file or compiled Java class file), as shown in the following table:

	Inline	By Reference
Regular Expression	directory.xml or task screen	—
JavaScript	directory.xml or task screen	Source file referenced in directory.xml
Java	—	Class file referenced in directory.xml or task screen

- **Associate one or more validation rules with a task screen field**. Do so in either or both of the following ways:
  - With task-level validation, you assign a validation rule directly to a field on a particular task screen.

    Task-level validation has task-specific scope—that is, it can be used only in the context of the particular task screen where it is assigned.
  - With directory-level validation, you map a rule set to a managed object attribute in directory.xml. Any task screen field that is configured with the attribute is validated against the rules in the rule set.

    Directory-level validation has global scope. This means that directory-level validation can be used on any field configured with the managed object attribute, regardless of the task screen that contains the field, and regardless of the Identity Manager environment that includes the task screen.

Using Default Validation Rules

Identity Manager is shipped with the following types of default validation rules:

- Data validation of task screen fields
- Predefined validation rules defined in the directory.xml file

Default Data Validations

By default, Identity Manager checks certain data when an administrator submits a task for processing. When the data is invalid, Identity Manager stops processing the task and displays an error message. The data validations that Identity Manager performs are based on the type of task, as shown in the following table:

Tasks	Validation
All tasks	Required fields must have a value.
Create User Create Group Create Organization Create Access Role Create Access Task Create Admin Role Create Admin Task	An administrator cannot create an object with the same name as an existing object of the same type. For example, an administrator cannot create two admin roles with the same name. **Note:** For users and groups, Identity Manager checks only the current organization.

Tasks	Validation
Create User Create Group Create Organization	An administrator cannot create a user, group, or organization with a name that contains any of the following characters: ■ comma (,) ■ single quote (’) ■ double quote (") ■ asterisk (*) ■ ampersand (&) ■ slash (/) ■ back slash (\) ■ less than sign (<) ■ greater than sign (>) ■ equal to sign (=) ■ plus sign (+) ■ semicolon (;) ■ pound sign (#) ■ leading or trailing spaces **Note:** Organization names can contain a comma (,) or an ampersand (&).
All Create and Modify tasks	Attributes with read/write permission (excluding passwords) cannot contain the following characters: ■ comma (,) ■ percent sign (%) ■ less than sign (<) ■ greater than sign (>) ■ semicolon (;) These characters are vulnerable to cross-site scripting attacks.

Tasks	Validation
Create User Self-register Change My Password Reset User Password Any custom task that collects and stores user passwords	If you are using SiteMinder's Password Services feature to enforce password rules (such as minimum length), user passwords are validated against these rules. If the password does not satisfy the password policy, the password is not accepted. **Note:** For more information, see the *CA SiteMinder Web Access Manager Policy Server Configuration Guide*.
Modify User	Administrators cannot give themselves a role or the ability to assign a role.
Forgotten Password	If a user profile does not have a password hint and answer, that user cannot use the forgotten password feature.
Delete User Enable/Disable User	Administrators cannot delete their own profile or change the status of their account.
Delete Organization	Administrators cannot delete the organization where they are assigned the role that contains the Delete Organization task. Consider an administrator who is assigned the Organization Manager role in the Dealers organization. The Organization Manager role enables this user to delete organizations. This administrator can delete suborganizations of Dealers, but cannot delete Dealers.
Modify Organization	Administrators cannot modify the organization where they are assigned the role that contains the Modify Organization task.

Predefined Validation Rules

Identity Manager includes the following validation rules predefined in the directory.xml file. Predefined validation rules are used for directory-level validation only, as shown in the following table:

Predefined Rule Name	Description
Phone pattern	Enforces the following format for telephone numbers: +nn nnn-nnn-nnnn
Set international	Adds the prefix +1 to an international telephone number.

Predefined Rule Name	Description
Valid User	Verifies that the specified User object exists in the directory.
Valid Group	Verifies that the specified Group object exists in the directory.
Valid Organization	Verifies that the specified Organization object exists in the directory.

Predefined validation rules and custom validation rules can appear in the same rule set.

How to Implement Custom Validation Rules

You can implement a validation rule for one of the following:

- Regular expression
- JavaScript
- Java class

Regular Expression Implementation

A validation rule can be based on regular expression pattern matching. For example, you can do the following:

- Specify a list of invalid characters or values for an attribute
- Restrict the user from typing invalid constructs, such as an improperly formed DN or telephone number

The following JavaScript example enforces telephone number format as +nn nnn-nnn-nnnn:

```
phone=/\+\d{1,3} \d{3}-\d{3}-\d{4}/;
```

Wrap regular expressions defined in XML in CDATA, as in the following example:

```
<ValidationRule name="Phone pattern" description="+nn nnn-nnn-nnnn"
 messageid="4001">
   <RegularExpression>
      <![CDATA[ ((\+|\d)*+(\s*|\x2D))?\d\d\d-\d\d\d-\d\d\d\d]]>
   </RegularExpression>
</ValidationRule>
```

Validation rules based on regular expressions must comply with the requirements defined in the java.util.regex package.

JavaScript Implementation

A JavaScript-based validation rule must implement the relevant interface, depending on whether the rule is used for task-level validation or directory-level validation.

At validation time, Identity Manager calls validate() and passes the value to be validated.

JavaScript Interface for Task-Level Validation

The definition of the JavaScript interface for task-level validation is as follows:

Syntax

```
public boolean validate(
   BLTHContext context,
   String attributeValue,
   StringRef changedValue,
   StringRef errorMessage
   );
```

Parameters

context

Input parameter

Specifies an object that contains methods for retrieving information in the current task session.

attributeValue

Input parameter

Specifies the value of the attribute being validated.

changedValue

Output parameter

Provides an optional transformation value that replaces the user-supplied value being validated. If no transformation is necessary, pass back null.

errorMessage

Output parameter

If validation fails, it displays a message to the user.

The message is displayed through AttributeValidationException. If the method returns false, Identity Manager generates this exception.

Comments

The output parameters *changedValue* and *errorMessage* are of data type StringRef. StringRef is a predefined data type that contains the field *reference* to which you assign a value, as shown in the following examples:

- Add a 1 prefix for a properly formatted telephone number:
  ```
  changedValue.reference="+1 " + phoneNumber;
  ```
- Provide an error message for an improperly formatted number:
  ```
  errorMessage.reference="Phone number " + phoneNumber +
              " does not match the format nnn-nnn-nnnn.";
  ```

Returns

- True. The implementation considers the value in *attributeValue* to be valid, or it passes back a transformed value in *changedValue*.
- False. The implementation considers *attributeValue* to be invalid. Identity Manager generates an AttributeValidationException that includes *errorMessage*.

JavaScript Interface for Directory-Level Validation

The definition of the JavaScript interface for directory-level validation is as follows:

Syntax

```
public boolean validate(
   String attributeValue,
   StringRef changedValue,
   StringRef errorMessage
   );
```

Parameters

attributeValue

Input parameter

Specifies the value of the attribute being validated.

changedValue

Output parameter

Provides an optional transformation value that replaces the user-supplied value being validated. If no transformation is necessary, pass back null.

errorMessage

Output parameter

If validation fails, it displays a message to the user.

The message is displayed through AttributeValidationException. If the method returns false, Identity Manager generates this exception.

Comments

The output parameters *changedValue* and *errorMessage* are of data type StringRef. StringRef is a predefined data type that contains the field *reference,* to which you assign a value, as shown in the following examples:

- Add a 1 prefix for a properly formatted telephone number:
  ```
  changedValue.reference="+1 " + phoneNumber;
  ```
- Provide an error message for an improperly formatted number:
  ```
  errorMessage.reference="Phone number " + phoneNumber +
              " does not match the format nnn-nnn-nnnn.";
  ```

Returns

- True—the implementation considers the value in *attributeValue* to be valid, or it passes back a transformed value in *changedValue*.
- False—the implementation considers *attributeValue* to be invalid. Identity Manager generates an AttributeValidationException that includes *errorMessage*.

Java Implementation

A Java-based validation rule must implement the relevant interface, depending on whether the rule is used for task-level validation or directory-level validation.

At validation time, Identity Manager calls validate() and passes the value to be validated.

Java Interface for Task-Level Validation

The definition of the JavaScript interface for task-level validation is as follows:

Syntax

```
public interface TaskValidator {
public class StringRef {
   public String reference = new String();
   public String toString(){return reference;}
public boolean validate(
   BLTHContext ctx,
   String attrValue,
   StringRef updatedValue,
   StringRef errorMessage
   ) throws AttributeValidationException;
}
```

Parameters

ctx

Input parameter

Specifies an object that contains methods for retrieving information in the current task session.

attrValue

Input parameter

Specifies the value of the attribute being validated.

updatedValue

Output parameter

Provides an optional transformation value that replaces the user-supplied value being validated. When no transformation is necessary, pass back null.

errorMessage

Output Parameter

If validation fails, it displays a message to the user.

Comments

For more information about Java validation rules and on managed objects, see the Identity Manager Javadoc.

Returns

- True—the implementation considers the value in *attributeValue* to be valid, or it passes back a transformed value in *changedValue*.
- False—the implementation considers *attributeValue* to be invalid.

Throws

AttributeValidationException

Java Interface for Directory-Level Validation

The definition of the Java interface for directory-level validation is as follows:

Syntax

```
public interface IAttributeValidator {
public class StringRef {
   public String reference = new String();
   public String toString(){return reference;}
public boolean validate(
   Object attributeValue,
   StringRef changedValue,
   StringRef errorMessage
   ) throws AttributeValidationException;
}
```

Parameters

attributeValue

Input parameter

Specifies the value of the attribute being validated.

changedValue

Output parameter

Provides an optional transformation value that replaces the user-supplied value being validated. When no transformation is necessary, pass back null.

errorMessage

Output parameter

If validation fails, it displays a message to the user.

Comments

If the validation operation requires managed objects from the directory, use AttributeValidator. This abstract class implements the IAttributeValidator interface, and includes a method for retrieving the managed object providers.

Returns

- True—the implementation considers the value in *attributeValue* to be valid, or it passes back a transformed value in *changedValue*.
- False—the implementation considers *attributeValue* to be invalid.

Throws

AttributeValidationException.

Exceptions

AttributeValidationException is thrown when a validation rule cannot validate an attribute value supplied in a task screen field or programmatically. The exception contains one or more messages that are presented to the user, enabling the user to correct the entry and resubmit the task.

How this exception is thrown and how the error messages are presented for the exception depends on whether the rule is implemented as JavaScript, a Java class, or a regular expression.

Exceptions with Task-Level Validation

With task-level validation errors, AttributeValidationException is thrown as shown in the following table:

Rule Type	How Thrown	Error Message Source
Regular expression	By Identity Manager if the regular expression validation fails.	Identity Manager uses a generalized exception message.
JavaScript	By Identity Manager if the validate() method returns False.	The *errorMessage* parameter of the validate() method.
Java	By the custom validation rule or by Identity Manager. Identity Manager throws the exception when the custom rule does not and the custom rule's validate() method returns False.	One of the following sources: ■ If the custom validation rule throws the exception, the exception's constructor. The constructor lets you specify the ID of a message in a resource bundle and the text of an additional message. ■ If Identity Manager throws the exception, the *errorMessage* parameter of the validate() method.

If the validation rule implementation does not provide an error message, Identity Manager uses a generalized error message.

Exceptions with Directory-Level Validation

Exception messages for directory-level validation errors come from two sources:

- A resource bundle. In directory.xml, definitions of all types of validation rules (Java, JavaScript, and regular expression) include the attribute messageid. This ID maps to a custom exception message in the resource bundle IMSExceptions.properties. When AttributeValidationException is thrown, Identity Manager includes the mapped message with other error information that may be defined for the validation rule.
- Custom validation rule code. Java and JavaScript implementations can define additional exception messages for the rule. If a validation error occurs in the Java or JavaScript rule, the message is presented to the user with the message that is mapped to the rule in the resource bundle.

  The sources of these Java and JavaScript exception messages are defined in the previous table.

  This feature does not apply to directory-level validation rules implemented as regular expressions.

Note: For more information about exception messages in resource bundles, see AttributeValidationException in the Identity Manager Javadoc.

AttributeValidationException Constructor

When you create an AttributeValidationException object for a Java validate() method, use the following constructor:

Syntax

```
public AttributeValidationException(String attrName,
   String attrValue,
   String messageid,
   String message);
```

Parameters

attrName

Specifies the name of the managed object attribute being validated.

attrValue

Specifies the value to validate.

messageid

If the value cannot be validated, it provides the ID associated with the message to display. The ID corresponds to a message in the resource bundle IMSExceptions.properties.

message

Provides an additional message that can be displayed to the user. This parameter gives you an opportunity to display a more specific message than the one in the resource bundle, or a message from a custom resource bundle.

Note: For more information about AttributeValidationException, see the Identity Manager Javadoc.

How to Configure Validation Rules

Configure a validation rule by integrating it with Identity Manager, and by directly or indirectly associating it with a task screen field.

How you configure a validation rule determines whether you want the rule applied to a field in a particular task screen (task-level validation) or a field in any task screen (directory-level validation), as follows:

- With task-level validation, you make a direct association between the rule and a field in a particular task screen. Validation is performed on the field in the context of that task screen only.
- With directory-level validation, the association between the rule and the task screen field is indirect, as follows:
  - In directory.xml, you specify the validation rule, add the rule to a rule set, and associate the rule set with a managed object attribute.
  - In the User Console, a field that is configured with the managed object attribute is validated against the rule set mapped to the attribute.

Validation is performed on any field configured with the attribute, regardless of the task screen that contains the field, and regardless of the Identity Manager environment that contains the task screen.

How to Configure Task-Level Validation

Configure task-level validation in the User Console, when defining field properties on a profile task screen. The basic steps are as follows:

1. Navigate to the Field properties section of the profile configuration screen containing the field to be validated.

   Note: For more information about field properties, see the *Administration Guide* and the User Console online help.

2. Specify a value in one of the following fields, depending on how the validation rule is to be implemented:

   - Validation Expression. Contains a regular expression that performs the validation.
   - Validation Java Class. Contains the fully qualified name of a Java class that performs the validation, for example:

     ```
     com.mycompany.MyJavaValidator
     ```

     Identity Manager expects the class file to be located in the root directory designated for custom Java class files.
   - Validation JavaScript. Contains the complete JavaScript code that performs the validation.

     You must provide JavaScript code in this field. With task-level validation, you cannot reference a file containing JavaScript code.

Note: For information about defining other field properties on a profile configuration screen, click the Help button on the screen.

How to Configure Directory-Level Validation

You configure directory-level validation in the directory.xml file and in a task screen. The basic steps are as follows:

- In the directory.xml file, do the following:
  - Specify a validation rule in the ValidationRule element.
  - Specify a rule set in the ValidationRuleSet element. A rule set contains one or more predefined rules, custom validation rules, or rules of both types.
  - Associate a rule set with a managed object attribute in the ImsManagedObjectAttr element.
- In a task screen, the field to be validated must be configured with the attribute mapped to the rule set.

Integration of Directory-Level Validation with Identity Manager

Define validation rules and rule sets to Identity Manager through the ImsManagedObjectAttrValidation element of the directory.xml file.

The schema for the ImsManagedObjectAttrValidation element is as follows:

```
<xs:element name="ImsManagedObjectAttrValidation" minOccurs="0">
   <xs:complexType>
      <xs:sequence>
         <xs:element name="ValidationRule" minOccurs="0"
                                          maxOccurs="unbounded">
            <xs:complexType>
               <xs:choice>
                  <xs:element name="Java">
                     <xs:complexType>
                        <xs:attribute name="class" type="xs:string"
                                                use="required"/>
                     </xs:complexType>
                           </xs:element>
                  <xs:element name="JavaScript">
                     <xs:complexType>
                        <xs:simpleContent>
                           <xs:extension base="xs:string"/>
                        </xs:simpleContent>
                     </xs:complexType>
                  </xs:element>
                  <xs:element name="JavaScriptFile">
                     <xs:complexType>
                      <xs:attribute name="file" type="xs:string"
                                             use="required"/>
                     </xs:complexType>
                  </xs:element>
                  <xs:element name="RegularExpression">
                     <xs:complexType>
                        <xs:simpleContent>
                           <xs:extension base="xs:string"/>
                        </xs:simpleContent>
                     </xs:complexType>
                  </xs:element>
               </xs:choice>
               <xs:attribute name="name" type="xs:string"
                                          use="required"/>
               <xs:attribute name="description" type="xs:string"
                                          use="optional"/>
               <xs:attribute name="messageid" type="xs:string"
                                          use="required"/>
            </xs:complexType>
         </xs:element>
```

```
        <xs:element name="ValidationRuleSet" minOccurs="0"
                                             maxOccurs="unbounded">
           <xs:complexType>
              <xs:sequence>
                 <xs:element name="ValidationRule"
                                             maxOccurs="unbounded">
                    <xs:complexType>
                       <xs:attribute name="name" type="xs:string"
                                             use="required"/>
                    </xs:complexType>
                 </xs:element>
              </xs:sequence>
              <xs:attribute name="name" type="xs:string"
                                             use="required"/>
              <xs:attribute name="description" type="xs:string"
                                             use="optional"/>
           </xs:complexType>
        </xs:element>
     </xs:sequence>
  </xs:complexType>
</xs:element>
```

The following elements are defined:

ValidationRuleSet

Consists of one or more predefined or custom validation rules. A validation rule is specified in the ValidationRule element.

Both predefined rules and custom rules can appear in the same rule set. Also, a rule set can contain any combination of Java, JavaScript, and regular expression implementations.

Validation rules are performed in the order in which they appear in ValidationRuleSet. This allows for cascading validation, where output from one rule is used as input to the next.

ValidationRuleSet is associated with a managed object attribute in the ImsManagedObjectAttr element of the directory.xml file.

ValidationRule

Specifies a validation rule for use in a ValidationRuleSet.

ValidationRule must contain only *one* of the following subelements:

- Java. References the Java class file that implements the rule.
- JavaScript. Contains the inline JavaScript code that implements the rule.
- JavaScriptFile. References the JavaScript source file that implements the rule.
- RegularExpression. Contains the inline regular expression that implements the rule. The regular expression must be wrapped in CDATA, as shown on Unsupported "page" cross-reference.

Key Attributes

Most of the attributes of the previously described elements are self-explanatory. However, the following attributes require explanation:

- Attribute class of element <Java>

  With Java validation rules, the Java class must be deployed in the following root location within your application server:

  `iam_im.ear\custom`

  Class files in this root location must be fully qualified, but need no other path information, for example, com.mycompany.MyJavaImpl.

- Attribute file of element <JavaScriptFile>

  With a validation rule implemented in a JavaScript source file, the file must be deployed in the following root location within your application server:

  `iam_im.ear\custom\validationscripts`

  JavaScript source files in this root location are referenced by name only, for example, MyJavaScriptImpl.js.

- Attribute messageid of element <ValidationRule>

  The message id specified in this attribute maps to an error message in the resource bundle IMSExceptions.properties.

  All types of validation rules (Java, JavaScript, JavaScriptFile, and RegularExpression) contain a messageid attribute.

Example: Inline Regular Expression

The following example shows the predefined Phone pattern validation rule, which is included in the rule set Phone format. The rule is implemented inline as a regular expression:

```
<ValidationRule name="Phone pattern" description="+nn nnn-nnn-nnnn"
                                                    messageid="4001">
   <RegularExpression>
      <![CDATA[ ((\+|\d)*+(\s*|\x2D))?\d\d\d-\d\d\d-\d\d\d\d]]>
   </RegularExpression>
</ValidationRule>
<ValidationRuleSet name="Phone format" description=
                                "Verify format +nn nnn-nnn-nnnn">
   <ValidationRule name="Phone pattern" />
</ValidationRuleSet>
```

In the preceding example, messageid="4001" maps to the following line in IMSExceptions.properties:

```
4001=Attribute Validation: {0} value must match regular expression
                                                        nnn-nnn-nnnn.
```

Example: Reference to JavaScript File

The following example specifies the rule EndWithZ_js. This rule is implemented in JavaScript, and the script is located in the file EndWithZ.js. The rule set that includes the rule is not shown in the example:

```
<ValidationRule name="EndWithZ_js" messageid="custom-5001">
   <JavaScriptFile file="EndWithZ.js" />
</ValidationRule>
```

In the preceding example, the JavaScript file is assumed to be in the following default location:

iam_im.ear\custom\validationscripts

Association of a Validation Rule Set with a Managed Object Attribute

Associate a validation rule set with a managed object attribute through the ImsManagedObjectAttr element of the directory.xml file.

In the following example, the validation rule set Phone format is associated with the managed object attribute telephonenumber:

```
<ImsManagedObjectAttr physicalname="telephonenumber" displayname="Business Phone"
description="Business Phone" valuetype="String" required="false"
multivalued="false" maxlength="0" validationruleset="Phone format" />
```

Note: When a managed object attribute is associated with a validation rule set, the rule set name is displayed in the Attribute Properties screen of the Management Console.

Association of a Validation Rule Set with a Task Screen Field

With directory-level validation, you can associate a rule set with a task screen field indirectly, as follows:

1. Associate the rule set with a managed object attribute, as described in the previous section.
2. Be sure that the task screen field to be validated is configured with the managed object attribute associated with the rule set. At runtime, a field value supplied by an end user is validated against the rules in the rule set.

Typically, task screen fields are already configured with attributes. However, you can add a field to a task screen, or you can change the attribute assigned to a field. In those cases, if you want the value supplied to the field to be subject to directory-level validation, configure the field with an attribute that is mapped in directory.xml to the appropriate rule set.

How to Initiate Validation

At run time, validation is initiated in any of the following ways:

User submits a task

Validates the fields on the submitted task screen that are associated with validation rules.

User navigates to a different task screen tab

Validates the fields in the vacated tab that are associated with validation rules.

User clicks a Validate button on a tab

Validates the fields in the current tab that are associated with validation rules.

The Validate button also executes Logical Attribute Handlers that include the validate method.

User changes a value in a field who's Validate on Change property is yes

Validates the fields in the current tab that are associated with validation rules.

For example, if Validate on change is enabled for an Employee Type field, and the field value is changed from Non-exempt to Exempt, all fields on the tab that are associated with validation rules are validated. One rule could require that a Salary field contain a value, and another rule could automatically change an Hourly Rate field to 0.

Custom code uses a setAttribute... method in AttributeCollection or a tab handler to set a managed object attribute value

The field is configured with the managed object attribute being set.

Sample Implementations

Sample JavaScript implementations of validation rules are located in the following samples directory of your Identity Manager installment:

```
Identity Manager\samples\validationscripts
```

Appendix A: List of Default Tabs

Identity Manager includes the following default tabs for admin tasks.

Access Role Administrators

Lets you add, view, or remove administrators of the current access role.

Access Role Membership

Lets you add, view, or remove members of the current access role.

Access Role Profile

Defines the profile for access roles.

Access Role Tasks

Lets you view a role's access tasks, or add or remove access tasks. You can select access tasks from different applications.

Access Roles

Lets you view, add, or remove the roles for the selected user and view that user's privileges.

Access Task Profile

Defines the profile for access tasks.

Accounts

Lists accounts in managed endpoints for users who have been assigned provisioning roles. Typically, this tab is added to tasks that allow you to view or modify a user.

Account Templates

Lets you add, remove, or view account templates associated with a provisioning role.

Admin Role Administrators

Lets you add, view, or remove administrators of the current admin role.

Admin Role Membership

Lets you add, view, or remove members of the current admin role.

Admin Role Profile

Defines the profile for admin tasks.

Admin Role Tasks

Lets you view a role's admin tasks, add or remove admin tasks, and select admin tasks from different categories.

Admin Roles

Lets you view, add, or remove admin roles for a selected user and view that user's member and administrator privileges.

Admin Task Profile

Defines the Profile tab for admin tasks.

Administrators

Lets you add, edit, or remove admin policies.

Approvers

Lists all participants who can approve or reject the work item. It also allows reassignment of the work item.

Approve Task

Displays information about individual approval tasks in a work list.

Approve Event

Displays information about individual approval tasks in a work list.

Certify User

Lets you certify or revoke a user's roles.

Currently Matched Policies/Policies Already Applied

Displays the synchronization status for users.

Events

Lets you select and configure a workflow process for each event that the task initiates.

Execute Explore and Correlate

Lets you select an explore and correlate definition to execute.

Execute Explore and Correlate Profile

Displays the containers in an endpoint that you can explore or correlate.

External Tab (ExternalTab)

Displays the contents of a URL within the tab in a task.

Fields

Lets you view the fields contained in the task. The fields are the attributes defined on the associated profile screen.

Group Administrators

Adds or removes administrators of the current group.

Group Membership

Adds or removes users as group members or adds or removes nested groups to this group.

Group Profile

Allows you to define or view the profile of the group.

Groups

Lets you view, add, or remove the groups for a selected user and view that user's privileges.

Identity Policy Set Owners

Lets you add owner rules, which are rules about who can modify the identity policy set.

Identity Policy Set Profile

Defines the profile of the identity policy set.

JSP

Displays custom information. See your system administrator for details.

Manage System or Orphan Accounts

Assigns a global user to a system or orphan account.

Members

Lets you add, edit, or remove member policies.

Organization Profile

Lets you create, modify, or view the profile of an organization.

Owners

Lets you add, edit, or remove owner policies.

Policies

Creates or modifies an identity policy.

Profile (AdminTaskProfile)

Lets you define the profile of the admin task.

Profile (Generic) (ObjectProfile)

Lets you define the profile for any managed object.

Provisioning Role Administrators

Lets you add, view, or remove administrators of the current provisioning role.

Provisioning Role Membership

Lets you add, view, or remove members of the current provisioning role.

Provisioning Role Profile

Defines the profile of the provisioning role.

Recurrence Tab

Controls the schedule for when the explore and correlate action should occur.

Scope (TaskScope)

Lets you limit the scope of the task. If the task has no primary object, or if the action is self-modify, self-view, or approve, the Search tab does not appear.

Schedule

Lets you automate the execution of a task at a later date.

Synchronization Summary

Displays the synchronization status for users.

Tabs (TaskTab)

Lets you select a tab controller, which determines how the tabs in a task are displayed, and view, add or remove the tabs included in the task.

User History

Displays a history of all the tasks that are initiated, approved, executed on, and performed by any user.

Work List

Displays a list of work items (or approval tasks) that appears in the Identity Manager User Console of the participant authorized to approve the task.

User Profile

Defines or displays the profile of a user.

This tab includes additional functionality, such as generating separate events for password changes, that is specific to user objects.

Appendix B: Compile the Identity Manager JSPs

After making changes to the Identity Manager JSPs used to generate the User Console and the Management Console on a JBoss application server, recompile the JSPs for changes to take effect.

The JSPs must be compiled using the compile_jsp.bat or .sh script.

The compile_jsp script creates a backup copy of the JSPs, and then recompiles them. The backup copies are located in the following directories:

- For the User Console, the compile_jsp script creates the iam/im_jsp_backup directory in the following location:
  iam_im.ear\user_console.war
- For the Management Console, the compile_jsp script creates the iam/im_jsp_backup directory in the following location:
  iam_im.ear\management_console.war

To recompile the JSPs in a JBoss environment

1. Stop the JBoss application server, if it is running.
2. From a command prompt, navigate to *jboss_home*\bin, where *jboss_home* is the installed location of the JBoss application server.
3. Execute one of the following scripts:
   - **Windows:** compile_jsp.bat
   - **UNIX:** compile_jsp.sh
4. Start the JBoss application server.

Index

F

H

I

J

L

M

O

P

R

S

T

U

V